

**PURDUE UNIVERSITY  
GRADUATE SCHOOL  
Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By Albert Parra Pozo

Entitled

Integrated Mobile Systems Using Image Analysis with Applications in Public Safety

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

EDWARD J. DELP

Chair

JAN P. ALLEBACH

MARY L. COMER

MIREILLE BOUTIN

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): EDWARD J. DELP

Approved by: M. R. Melloch

Head of the Graduate Program

07-02-2014

Date

INTEGRATED MOBILE SYSTEMS USING IMAGE ANALYSIS WITH  
APPLICATIONS IN PUBLIC SAFETY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Albert Parra Pozo

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2014

Purdue University

West Lafayette, Indiana

## ACKNOWLEDGMENTS

I would like to thank my first major advisor, Professor Edward J. Delp, for all the challenges he has given me so far, and for believing that I can overcome them. I really value his advice and criticism; it helps me make the most of my academic career.

I am also very thankful for the support and guidance of Professor Mireille Boutin. I appreciate her finding the time to help me with the research, and helping me organize my ideas and see things from different points of view.

I would like to thank the remaining members of my Graduate Committee, Professor Jan P. Allebach and Professor Mary L. Comer.

I want to give special thanks to Dr. Marc Bosch for his advice and support during the time we share at Purdue, and to Andrew W. Haddad for his patience and help in my both my academic and personal life. Special thanks to Dr. Ye He for believing in me and helping me become a better person.

It has been a pleasure being part of the Video and Image Processing Laboratory (VIPER), both for the quality of the research carried out in the lab and for the people involved. Thanks to my current and former colleagues Jeehyun Choe, Neeraj Gadgil, Joonsoo Kim, Deen King-Smith, Dr. Nitin Khanna, Soonam Lee, He Li, Dr. Kevin Lorenz, Dr. Aravind Mikkilineni, Dr. Ka Ki Ng, Thitiporn Pramoun, Dr. Satyam Srivastava, Khalid Tahboub, Kharittha Thongkor, Yu Wang, Dr. Chang Xu, Dr. Meilin Yang, Bin Zhao, and Dr. Fengqing Maggie Zhu.

I would like to thank my parents for supporting my career decisions and always believing in me. Thanks to them for giving me the opportunity to acquire and share knowledge with others.

The gang graffiti images shown in this thesis were obtained in cooperation with the Indianapolis Metropolitan Police Department.

The hazmat sign images shown in this thesis were obtained in cooperation with the Transportation Security Administration.

We gratefully acknowledge their cooperation in GARI and MERGE.

This work was supported by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI000.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	xii
ABSTRACT . . . . .	xxiv
1 INTRODUCTION . . . . .	1
1.1 Problem Formulation and Challenges . . . . .	1
1.2 Contributions of This Thesis . . . . .	2
1.3 Publications Resulting From This Work . . . . .	4
2 OVERVIEW OF GANG GRAFFITI AND HAZMAT SIGN DETECTION SYSTEMS . . . . .	5
2.1 Overview of Gang Graffiti Systems . . . . .	5
2.1.1 Graffiti Tracker . . . . .	5
2.1.2 TAGRS . . . . .	6
2.1.3 GRIP . . . . .	6
2.1.4 GTS . . . . .	7
2.1.5 GAT . . . . .	7
2.1.6 TAG-IMAGE . . . . .	8
2.1.7 Graffiti-ID . . . . .	9
2.1.8 Other Work on Graffiti and Tattoos . . . . .	12
2.1.9 Comparison to GARI . . . . .	13
2.2 Overview of Hazmat Sign Detection Systems . . . . .	17
2.2.1 Hazmat Sign Detection Based on SURF and HBP . . . . .	17
2.2.2 Hazmat Sign Detection Based on HOG . . . . .	18
2.2.3 Comparison to MERGE . . . . .	19
2.3 Proposed Systems . . . . .	20

	Page
2.3.1 GARI . . . . .	20
2.3.2 MERGE . . . . .	29
3 GANG GRAFFITI AUTOMATIC RECOGNITION AND INTERPRETA- TION (GARI) . . . . .	35
3.1 Review of Existing Methods . . . . .	35
3.1.1 Blur Detection . . . . .	35
3.1.2 Color Correction . . . . .	37
3.1.3 Color Recognition . . . . .	39
3.1.4 Color Image Segmentation . . . . .	40
3.1.5 Graffiti Content Analysis . . . . .	42
3.1.6 Image Features . . . . .	44
3.1.7 Image Retrieval . . . . .	48
3.2 Mobile-Based Motion Blur Prevention and Detection . . . . .	50
3.3 Color Correction Based on Mobile Light Sensor . . . . .	56
3.4 Color Recognition Based on Touchscreen Tracing . . . . .	64
3.5 Automatic Graffiti Component Segmentation . . . . .	68
3.5.1 Color Image Segmentation Based on Gaussian Thresholding	68
3.5.2 Block-Wise Gaussian Segmentation Enhancement . . . . .	72
3.5.3 Background Stripe Removal . . . . .	79
3.5.4 Graffiti Component Reconnection . . . . .	90
3.6 Gang Graffiti Features . . . . .	98
3.7 Content Based Gang Graffiti Image Retrieval . . . . .	106
3.8 System Implementation . . . . .	115
3.8.1 System Architecture . . . . .	115
3.8.2 GARI Databases . . . . .	115
3.8.3 Android/iOS Implementation . . . . .	121
3.8.4 Web Interface . . . . .	141
4 MOBILE EMERGENCY RESPONSE GUIDE (MERGE) . . . . .	161

	Page	
4.1	Review of Existing Methods . . . . .	161
4.1.1	Sign location detection . . . . .	161
4.1.2	Sign recognition . . . . .	164
4.2	Segment Detection Using Geometric Constraints . . . . .	165
4.3	Convex Quadrilateral Detection Based on Saliency Map . . . . .	168
4.4	Sign Location Detection Based on Fourier Descriptors . . . . .	178
4.5	System Implementation . . . . .	190
4.5.1	System Overview . . . . .	190
4.5.2	MERGE Databases . . . . .	192
4.5.3	Android/iOS Implementation . . . . .	195
4.5.4	Web Interface . . . . .	210
5	EXPERIMENTAL RESULTS . . . . .	215
5.1	GARI . . . . .	215
5.1.1	RGB to Y'CH Conversion . . . . .	215
5.1.2	Color Correction Based on Mobile Light Sensor . . . . .	218
5.1.3	Content Based Image Retrieval . . . . .	224
5.1.4	End-To-End System . . . . .	254
5.1.5	Database of Gang Graffiti . . . . .	277
5.1.6	Database Query Performance . . . . .	277
5.2	MERGE . . . . .	278
5.2.1	Segment Detection Using Geometric Constraints . . . . .	281
5.2.2	Convex Quadrilateral Detection Based on Saliency Map . . . . .	281
5.2.3	Sign Location Detection Based on Fourier Descriptors . . . . .	285
6	CONCLUSIONS AND FUTURE WORK . . . . .	287
6.1	Conclusions . . . . .	287
6.2	Project Status . . . . .	289
6.3	Future Work . . . . .	291
6.3.1	GARI . . . . .	291

	Page
6.3.2 MERGE . . . . .	294
6.4 Publications Resulting From This Work . . . . .	295
LIST OF REFERENCES . . . . .	296
A RGB TO Y'CH COLOR SPACE CONVERSION . . . . .	320
B EXAMPLES OF GRAFFITI COLOR IMAGE SEGMENTATION . . . . .	327
C IMAGE THRESHOLDING METHODS . . . . .	336
D GARI DATABASE TABLES . . . . .	359
E MERGE DATABASE TABLES . . . . .	363
F GARI IMAGE ACQUISITION PROTOCOL . . . . .	370
G MERGE IMAGE ACQUISITION PROTOCOL . . . . .	377
VITA . . . . .	385



## LIST OF TABLES

Table	Page
2.1 Accuracy and execution time for various numbers of candidate images from the manual annotation matching step. . . . .	11
2.2 Comparison of features between different gang graffiti systems and GARI.	14
3.1 Image feature types and sizes. . . . .	45
3.2 Parameters and thresholds used in Mobile-Based Motion Blur Prevention.	55
3.3 Thresholds for common lighting conditions and corresponding lighting steps. . . . .	57
3.4 Parameters and thresholds used in Color Recognition Based on Touch-screen Tracing. . . . .	66
3.5 Parameters and thresholds used in Color Image Segmentation Based on Gaussian Thresholding. $W_X$ and $H_X$ are the width and height of $X$ respectively. . . . .	71
3.6 Parameters and thresholds used in Block-Wise Gaussian Segmentation Enhancement. $W_X$ and $H_X$ are the width and height of $X$ respectively.	76
3.7 Parameters and thresholds used in Background Stripe Removal. $W_X$ and $H_X$ are the width and height of $X$ respectively. . . . .	88
3.8 Relationship Between Directions and Zones in the Chain Code. . . . .	91
3.9 Parameters and thresholds used in Graffiti Component Reconnection. .	97
3.10 Parameters and thresholds used for the Gang Graffiti Features. . . . .	104
3.11 Parameters and thresholds used in Content Based Gang Graffiti Image Retrieval. . . . .	113
3.12 Web Browsers Supporting HTML5 Geolocation Service. . . . .	143
4.1 Parameters and thresholds used in Segment Detection Using Geometric Constraints. $W_X$ and $H_X$ are the width and height of $X$ respectively. $e = \max(l_p, l_r)$ . . . . .	169
4.2 Parameters and thresholds used in Convex Quadrilateral Detection Based on Saliency Map. $W$ and $H$ are the width and height of the saliency map. $S(x, y)$ is the saliency value at $(x, y)$ . . . . .	178

Table	Page
4.3 Parameters and thresholds used in our proposed method. Automatically determined values are denoted by *. $W$ and $H$ are the width and height of the image. . . . .	189
5.1 Execution Time (seconds) of the Arithmetic and the Trigonometric Approaches For Color Conversion. . . . .	217
5.2 Mean Channel Errors ( $\Delta$ ) and Average Running Times (seconds) For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL). . . . .	222
5.3 Weighted Top-5 Accuracies of Scene Recognition for Different Values of $k$ and $n_w$ (percentage). . . . .	228
5.4 Top-1 Accuracies of Scene Recognition for Different Values of $k$ and $n_w$ (percentage). . . . .	229
5.5 Training Times of Scene Recognition for Different Values of $k$ and $n_w$ (minutes). . . . .	231
5.6 Query Times of Scene Recognition for Different Values of $k$ and $n_w$ (seconds). . . . .	232
5.7 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [4 \dots 17]$ (percentage). . . . .	238
5.8 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [18 \dots 30]$ (percentage). . . . .	239
5.9 Top-10 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [4 \dots 17]$ (percentage). . . . .	241
5.10 Top-10 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [18 \dots 30]$ (percentage). . . . .	242
5.11 Top-5 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [4 \dots 17]$ (percentage). . . . .	244
5.12 Top-5 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [18 \dots 30]$ (percentage). . . . .	245
5.13 Classification Accuracy, Precision, Recall and $F_1$ Score for Each Class.	247
5.14 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 10]$ and $n_\theta \in [4 \dots 17]$ using SIFT Descriptors (percentage). . . . .	250
5.15 Top-10 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 10]$ and $n_\theta \in [4 \dots 17]$ using SIFT Descriptors (percentage). . . . .	251

Table	Page
5.16 Top-5 Classification Accuracies of Gang Graffiti Component Classification for $n_r \in [1 \dots 10]$ and $n_\theta \in [4 \dots 17]$ using SIFT Descriptors (percentage).	252
5.17 Example of <i>MAP</i> score calculation for a set of two queries. The total <i>MAP</i> score is $\frac{0.22+0.41}{2} = 0.31$ .	253
5.18 MAP Scores of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [4 \dots 17]$ (percentage).	255
5.19 MAP Scores of Gang Graffiti Component Classification for $n_r \in [1 \dots 20]$ and $n_\theta \in [18 \dots 30]$ (percentage).	256
5.20 Running Times (seconds) of Each Step in The GARI End-To-End System. 1: Color Correction Based on Mobile Light Sensor, 2: Color Image Segmentation Based on Gaussian Thresholding, 3: Block-Wise Gaussian Segmentation Enhancement, 4: Background Stripe Removal, 5: Graffiti Component Reconnection, 6: Graffiti Component Classification.	260
5.21 Running Times (seconds) of The Three Main Blocks in The GARI End-To-End System. 1: Color Correction, 2: Automatic Graffiti Component Segmentation, 3: Graffiti Component Classification. CCs: Number of Connected Components.	261
5.22 Automatic Segmentation and Graffiti Component Classification Accuracies. N GC: Number of gang graffiti components. N GC Rec: Number of recognizable gang graffiti components.	275
5.23 Average Running Times (seconds) and Accuracies of The Three Main Blocks in The GARI System on Testing Dataset.	276
5.24 Number of Images and Users In the Different GARI Systems.	277
5.25 Elapsed Time On the Hand-Held Device and the Server When Uploading an Image.	278
5.26 Analysis Results: Segment Detection Using Geometric Constraints.	281
5.27 Average Execution Time (in Seconds), Distribution and Score of Each Saliency Map Method (Color Spaces).	283
5.28 Image Analysis Results: Convex Quadrilateral Detection Based on Saliency Map.	284
5.29 Analysis Results: Sign Location Detection Based on Fourier Descriptors.	285
5.30 Image Analysis Results for the Three Proposed Methods. 1: Segment Detection Using Geometric Constraints, 2: Convex Quadrilateral Detection Based on Saliency Map, 3: Sign Location Detection Based on Fourier Descriptors.	285

Table	Page
6.1 Android/iOS versions of the GARI and MERGE mobile applications. .	289
D.1 EXIF data fields in Table <i>images</i> . . . . .	359
D.2 Image location fields in Table <i>images</i> . . . . .	360
D.3 Graffiti analysis fields in Table <i>images</i> . . . . .	360
D.4 Image information fields in Table <i>images</i> . . . . .	361
D.5 User information fields in Table <i>users</i> . . . . .	361
D.6 Image blobs information fields in Table <i>imageBlobs</i> . . . . .	362
E.1 EXIF data fields in Table <i>images</i> . . . . .	363
E.2 Image location fields in Table <i>images</i> . . . . .	364
E.3 Image information fields in Table <i>images</i> . . . . .	364
E.4 User information fields in Table <i>users</i> . . . . .	365
E.5 Fields in Table <i>class</i> . . . . .	365
E.6 Fields in Table <i>colorids</i> . . . . .	365
E.7 Fields in Table <i>colorpages</i> . . . . .	365
E.8 Fields in Table <i>placard</i> . . . . .	366
E.9 Fields in Table <i>symbol</i> . . . . .	366
E.10 Fields in Table <i>textcolors</i> . . . . .	366
E.11 Fields in Table <i>textids</i> . . . . .	366
E.12 Fields in Table <i>textpages</i> . . . . .	367
E.13 Fields in Table <i>unids</i> . . . . .	367
E.14 Fields in Table <i>vw01_orange_page</i> . . . . .	367
E.15 Fields in Table <i>vw03_yellow_page</i> . . . . .	367
E.16 Fields in Table <i>vw05_water_reactive_materials</i> . . . . .	368
E.17 Fields in Table <i>vw06_tiiapad</i> . . . . .	368

## LIST OF FIGURES

Figure	Page
2.1 Block Diagram of the Graffiti-ID System. . . . .	10
2.2 Block Diagram of The System in [18]. . . . .	13
2.3 Block Diagram of the GARI System. . . . .	23
2.4 Modular Components of the GARI System. . . . .	23
2.5 Examples of Graffiti Elements. . . . .	26
2.6 Examples of Graffiti Color Recognition. . . . .	27
2.7 Block Diagram of the MERGE System. . . . .	31
2.8 Possible Shapes of Hazmat Signs. . . . .	32
2.9 Elements That Uniquely Identify a Hazmat Sign. From Left to Right: UNID, Symbol, and Class Number. . . . .	33
2.10 Possible Symbols On a Placard. . . . .	33
2.11 Possible Colors On a Placard. . . . .	34
3.1 Example of Blur Metric Results. . . . .	54
3.2 Lighting Step vs. Luminance (lux). . . . .	57
3.3 Lighting Step vs. Luminance (log(lux)). . . . .	58
3.4 Color Correction Based on Mobile Light Sensor. . . . .	59
3.5 Example of ground-truth image with a lux value of 5,116. . . . .	60
3.6 Example of color correction when $LX = 35,611$ . Left: before correction; right: after correction. . . . .	63
3.7 Example of color correction when $LX = 41,980$ . Left: before correction; right: after correction. . . . .	63
3.8 Color Recognition Based on Touch Screen Tracing. . . . .	64
3.9 Separation Between Hue Averages. . . . .	67
3.10 Color Image Segmentation Using Gaussian Thresholding. . . . .	69
3.11 Gaussian Thresholding on Blue. $(\tilde{H}, \sigma_{\tilde{H}}^2) = (4.19, 0.05)$ . . . . .	70

Figure	Page
3.12 Probability Map Created By The Gaussian Thresholding. . . . .	71
3.13 Gaussian Thresholding results with non-uniform scene illumination. . .	72
3.14 Gaussian Thresholding results with foreground-background hue similarity.	73
3.15 Block-Wise Gaussian Segmentation Enhancement. . . . .	74
3.16 Example of Block-Wise Gaussian Segmentation Enhancement. . . . .	77
3.17 Example of Block-Wise Gaussian Segmentation Enhancement (continued).	78
3.18 Background stripes affecting gang graffiti component segmentation. . .	79
3.19 Background Stripe Removal. . . . .	79
3.20 Example of Background Stripes Removal During the Gaussian Thresholding Step. . . . .	80
3.21 Connectivity of $p$ . Pixels are connected to $p$ if they have the same value as $p$ . Only pixel locations in red are considered in each connectivity. . .	80
3.22 Skeletonization via Parallel Thinning [225]. . . . .	82
3.23 Parametric Representation of a Line. . . . .	83
3.24 Standard Hough Transform accumulator array. Peaks corresponding to potential lines are marked with green squares. . . . .	84
3.25 Bresenham's Technique: mathematical line (red) and elements of $S_{(x,y)}$ (gray). . . . .	85
3.26 Step of Bresenham's Technique. . . . .	86
3.27 Final window sizes at different locations using our modified Bresenham's Technique. . . . .	87
3.28 Modified Bresenham Technique. Green areas correspond to removed line segments; blue areas correspond to ignored line segments. . . . .	88
3.29 Example of Background Stripe Removal. . . . .	89
3.30 Graffiti Component Reconnection. . . . .	90
3.31 $3 \times 3$ templates to detect an endpoint. The endpoint is at the center of the template. . . . .	90
3.32 Endpoint Detection. . . . .	92
3.33 Chain Code For Endpoint Direction Detection. . . . .	93
3.34 Example of Graffiti Component Reconnection. . . . .	95

Figure	Page
3.35 Example of connected components after Gaussian Thresholding and after Graffiti Component Reconnection. . . . .	96
3.36 DoG Pyramid. . . . .	99
3.37 Neighboring Pixels (green) For Keypoint Extraction (red). . . . .	100
3.38 Keypoint Descriptor Generation. The red dot represents the location of the keypoint. . . . .	101
3.39 25 SIFT descriptors selected at random. Each keypoint is represented by a set of gradient magnitude histograms (green) rotated to its dominant local orientation (yellow). The size of the green grid represents the scale of the descriptor. . . . .	102
3.40 Local Shape Descriptor histogram for a specific keypoint and its matrix representation. The matrix holds the count distribution of SIFT keypoint locations relative the specific keypoint. . . . .	105
3.41 Gang Graffiti Scene Recognition. . . . .	106
3.42 Gang Graffiti Component Classification. . . . .	107
3.43 Four Main Steps in $k$ -Means. . . . .	109
3.44 Vocabulary Tree Built From Hierarchical $k$ -Means. Each black dot corresponds to a descriptor from a database image. . . . .	110
3.45 Scalability Results of Vocabulary Tree tested on a 6,376 ground-truth image dataset [196]. From left to right: Performance vs number of leaf nodes with branch factor $k = 8, 10$ and $16$ . Performance vs $k$ for one million leaves. Performance vs training data volume in $720 \times 480$ frames, run with 20 training cycles and $k = 10$ . Performance vs number of training cycles run on 7K frames of training data and $k = 10$ . The image belongs to [196]. . . . .	113
3.46 Majority Voting Matching. . . . .	114
3.47 Overview of The GARI System - Client-Side Components (green) and Server-Side Components (blue). . . . .	116
3.48 Database Schema Showing The Associations Between the Tables in the Database. . . . .	118
3.49 Example of Graffiti (Manually Labeled). . . . .	120
3.50 Database Fields With Information From The Graffiti in Figure 3.49. . .	121
3.51 Overview of the GARI System. . . . .	123

Figure	Page
3.52 Automatic updates. . . . .	124
3.53 User options screens for Android (4.26a, 4.26b) and iPhone (3.53c, 3.53d).	125
3.54 Examples of location of the menu button (red square) on Android devices.	125
3.55 Example of image browsing. . . . .	126
3.56 Browse by radius screen for Android (left) and iPhone (right). . . . .	127
3.57 Progress dialog notifying the user of a location retrieval, for Android (left) and iPhone (right). . . . .	128
3.58 3.58a Dialog notifying the user that no Network or GPS systems are enabled, and 3.58b location settings of the device, for Android. . . . .	128
3.59 Screen notifications during database browsing for Android (3.59a, 3.59b) and iPhone (3.59c, 3.59d). . . . .	129
3.60 Results after querying the image database for Android (left)) and iPhone (right). . . . .	129
3.61 Extended results after querying the image database for Android (left) and iPhone (right). . . . .	130
3.62 Graffiti locations displayed on a map for Android (left) and iPhone (right)	131
3.63 Graffiti locations displayed on an Augmented Reality feed for Android	132
3.64 Camera Activity. . . . .	133
3.65 Result of uploading an image to the server for Android (3.65a and 3.65b) and iPhone (3.65c and 3.65d). . . . .	134
3.66 Image uploading on the background on Android (top) and iPhone (bottom). From left to right (Android): Uploading image (icon), waiting for Internet connection, uploading 3 images, image successfully uploaded. From left to right (iPhone): Messages on the notification bar, Uploading image (message), image successfully uploaded (message). . . . .	135
3.67 Image upload successfully (3.67a) and image already uploaded to database (3.67b). . . . .	136
3.68 Screen notifications when finding similar images (Android). . . . .	136
3.69 Steps to follow when selecting the region to analyze the color for Android (top) and iPhone (bottom). . . . .	137
3.70 Image Analysis Results. . . . .	138



Figure	Page
3.71 Gangs related to the traced color and images in the database that match the traced color for Android (3.71a, 3.71b) and iPhone (3.71c, 3.71d). . . . .	139
3.72 User ID Prompt. . . . .	140
3.73 “Settings” Dialog, Showing the Various Options. . . . .	141
3.74 Overview of the Web Interface of the GARI System. . . . .	142
3.75 Main Page of the Web Interface of GARI. . . . .	148
3.76 “Archive” Section of Desktop GARI. . . . .	149
3.77 “Browse database” section of the web-based interface for GARI. . . . .	149
3.78 The current location of the user is only acquired upon request. . . . .	149
3.79 Results of browsing the database. . . . .	150
3.80 Example of the interactive map when a single image is displayed. . . . .	150
3.81 Example of the interactive map when multiple images are displayed. . . . .	151
3.82 If “Open in a new window” is clicked, the interactive map expands to a full screen to make navigation easier. . . . .	151
3.83 Example of a popped out balloon on the interactive map when a marker is clicked. . . . .	152
3.84 Example of “More information” result for a specific search in the database. . . . .	153
3.85 “Upload Image” Section of Desktop GARI. . . . .	154
3.86 Preview of an Image Before Uploading It to the Graffiti Database. . . . .	154
3.87 After uploading the image to the database, the user can select where the image was taken using an interactive map. . . . .	155
3.88 After uploading the image to the database, information can still be added. . . . .	155
3.89 Upload multiple images: Main screen. . . . .	156
3.90 Upload multiple images: Select multiple files. Note that the appearance of this screen may vary depending on the operating system used. . . . .	156
3.91 Upload multiple images: List of images to upload. . . . .	157
3.92 Upload multiple images: Upload progress. . . . .	157
3.93 Upload multiple images: Review screen. . . . .	157
3.94 Create database report. . . . .	159
3.95 Create database report: download screen. . . . .	159

Figure	Page
3.96 Login Page for Accessing the Gang Graffiti Archive. . . . .	160
4.1 Segment Detection Using Geometric Constraints. . . . .	166
4.2 Structuring Elements Used for Erosion. . . . .	166
4.3 First method (left to right): original image, segments at $\pm 45^\circ$ , grouped segments, optimal bounding box. . . . .	168
4.4 Issue With First Method: Grayscale. Sign Is Lost On Line Detection Process. . . . .	171
4.5 Issue With First Method: Low Resolution. Sign Is Lost On Erosion Process. . . . .	171
4.6 Issue With First Method: Sign Distortion. Sign Is Lost On Erosion Process. . . . .	171
4.7 Issue With First Method: Segment Merging. Sign Is Lost On Segment Grouping Process. . . . .	172
4.8 Issue With First Method: Shade. Sign Color Is Not Recognized Properly. . . . .	172
4.9 Proposed Hazmat Sign Detection and Recognition Method. . . . .	173
4.10 Saliency Map Method Obtained On Lab (Middle) and RGB (Right) Color Spaces. . . . .	173
4.11 Saliency Map Method Obtained On Lab (Middle) and RGB (Right) Color Spaces. . . . .	174
4.12 Structuring Element Used for Dilation. . . . .	176
4.13 Second Method: True Positives. . . . .	177
4.14 Second Method: True Positive/False Positive. . . . .	177
4.15 Sign Location Detection Based on Fourier Descriptors. . . . .	179
4.16 Example of image binarization using our proposed color channel thresholding method comparing with Ostu's method. . . . .	181
4.17 Examples of input images (left) and their contours (right). . . . .	182
4.18 A diamond shaped binary image is used as a shape template. . . . .	185
4.19 Reconstruction of the shape template using the first 2, 5, 8, 16, 30, 50, 80 and 100 Fourier coefficients. . . . .	188
4.20 Comparison of our shape template contour against different shape templates and their matching costs $e$ . . . . .	188

Figure	Page
4.21 Mobile-Based Hazmat Sign Detection and Recognition. . . . .	191
4.22 Overview of the MERGE Client-Side Components. . . . .	192
4.23 Overview of the MERGE Server-Side Components. . . . .	193
4.24 Database Schema Showing The Associations Between the Tables in the Database. . . . .	195
4.25 Automatic updates. . . . .	197
4.26 Main Screen. . . . .	197
4.27 Screens for browsing images. . . . .	198
4.28 Methods for browsing. Android (top) and iPhone (bottom). . . . .	199
4.29 Guide page in the ERG 2012 and corresponding guide page in Mobile MERGE for Android (middle) and iPhone (right). . . . .	201
4.30 Evacuation region for Android (top) and iPhone (bottom). From left to right, questions asked to refine evacuation region, and general evacuation circle and weather-based plume model. . . . .	203
4.31 Camera Interface with “SIGN” and “SCENE” options. . . . .	204
4.32 Results of the Image Analysis Process. Android (top) and iPhone (bottom)	205
4.33 User ID Screen. . . . .	207
4.34 Settings Menu Options. Android (top) and iPhone (bottom). . . . .	209
4.35 “Internal” Section of Desktop MERGE. . . . .	211
4.36 Search Guidebook Pages by Color, Symbol, Class, or UNID . . . . .	211
4.37 Browse Guidebook Page Results . . . . .	212
4.38 View Guidebook Page . . . . .	213
4.39 Browse Images . . . . .	214
5.1 Execution Time with Respect to the Number of Data Points for the Arithmetic and the Trigonometric Approaches For Color Conversion. . . . .	217
5.2 Distribution of Lux Values for Each Lightning Step. . . . .	218
5.3 Fiducial Marker (left) and GrehtagMacbeth Colorchecker (right). . . . .	220
5.4 Color Correction Example Under Each Scenario and Each Mapping. M1: using a fiducial marker in every image, M2: using a fiducial marker every week, M3: using the mobile light sensor value. . . . .	221

Figure	Page
5.5 Mean Channel Errors ( $\Delta$ ) For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL). . . . .	222
5.6 Average Running Times For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL). . . . .	223
5.7 Samples from Training Dataset. . . . .	225
5.8 Samples Image Matches. Left: Training Images (Samsung Galaxy Nexus). Right: Matching Testing Images (Casio PowerShot S95). . . . .	226
5.9 Color Map of Weighted Top-5 Accuracies of Scene Recognition Using Different Values of $k$ and $n_w$ . . . . .	227
5.10 Color Map of Top-1 Accuracies of Scene Recognition Using Different Values of $k$ and $n_w$ . . . . .	230
5.11 Color Map of Query Times of Scene Recognition Using Different Values of $k$ and $n_w$ . . . . .	233
5.12 Number of Vocabulary Tree Nodes As a Function of $k$ and $n_w$ . . . . .	234
5.13 Number of Vocabulary Tree Levels As a Function of $k$ and $n_w$ . . . . .	234
5.14 Query Images (Left) And Similar Retrieved Scenes (Right). . . . .	235
5.15 Sample Images for Each Class. From left to right, top to bottom, in groups of 4 images: <i>0</i> , <i>1</i> , <i>8</i> , <i>X</i> , <i>G</i> , <i>5-point star</i> , <i>3</i> , <i>6-point star</i> , <i>E</i> , <i>4</i> , <i>S</i> , <i>pitchfork</i> , <i>2</i> , and <i>arrow</i> . Note the inter-class variance as well as the intra-class similarity. . . . .	236
5.16 Color Map of Classification Accuracies of Gang Graffiti Component Classification Using Different Values of $n_r$ and $n_\theta$ . . . . .	240
5.17 Color Map of Top-10 Classification Accuracies of Gang Graffiti Component Classification Using Different Values of $n_r$ and $n_\theta$ . . . . .	243
5.18 Color Map of Top-5 Classification Accuracies of Gang Graffiti Component Classification Using Different Values of $n_r$ and $n_\theta$ . . . . .	246
5.19 Confusion Matrix for the 14 Graffiti Component Classes. . . . .	248
5.20 Color Map of MAP Scores of Gang Graffiti Component Classification Using Different Values of $n_r$ and $n_\theta$ . . . . .	254
5.21 GARI End-To-End System. . . . .	257
5.22 Test Images for Automatic Gang Graffiti Segmentation. . . . .	258
5.23 Images Segmented Separately From Two Different TouchScreen Tracings. . . . .	259

Figure	Page
5.24 Examples of our proposed Color Image Segmentation Based on Gaussian Thresholding followed by Block-Wise Gaussian Segmentation Enhancement. . . . .	263
5.25 Comparison of our proposed color image segmentation method against Niblack and Otsu thresholding. From top to bottom: 1001, 1002, 1004, 1017. . . . .	264
5.26 Examples of Background Strip Removal. . . . .	265
5.27 Examples of Background Strips Automatically Removed in Previous Steps. . . . .	266
5.28 End-Points in Skeleton of Image 1011. . . . .	266
5.29 Examples of Graffiti Component Reconnection. . . . .	267
5.30 Number of Connected Components (CCs) Before and After Automatic Gang Graffiti Segmentation. . . . .	268
5.31 Merged Connected Components Forming Words. . . . .	269
5.32 Automatically Segmented Candidate Graffiti Components. . . . .	271
5.33 Classification Results and Top-10 Matches for Candidates 1 to 8. . . . .	272
5.34 Classification Results and Top-10 Matches for Candidates 9 to 15. . . . .	273
5.35 Automatic Segmentation and Classification from Multiple Colors. . . . .	274
5.36 Example Images From The Test Dataset. . . . .	280
5.37 Saliency map categories (top to bottom, left to right): original image, good, fair; original image, bad, lost. . . . .	284
5.38 Examples of sign location detection. Column from left to right: results from [257], results from [314], results from proposed method. . . . .	286
6.1 Evolution of the Elements in $M$ With the Lightning Step (Lux Value). . . . .	292
A.1 Steps For Transforming from RGB to Y'CH Using The Arithmetic Approach. . . . .	321
A.2 Warping of the Hexagon Projection Into A Circumference in Our Y'CH Color Space. . . . .	322
A.3 3D view of Our Y'CH Color Space (Using the Arithmetic Approach). . . . .	323
A.4 Cross-Section of Constant Hue $H = 0$ rad in Our Y'CH Color Space. . . . .	324
A.5 Cross-Section of Constant Hue $H = \frac{\pi}{3}$ rad in Our Y'CH Color Space. . . . .	324
A.6 Cross-Section of Constant Hue $H = \frac{2\pi}{3}$ rad in Our Y'CH Color Space. . . . .	325

Figure	Page
A.7 Bottom View of Our Y'CH Color Space (Using the Arithmetic Approach).	325
A.8 Bottom View of Our Y'CH Color Space (Using the Trigonometric Approach). . . . .	326
B.1 Red text: $\tilde{H} = 0.49$ and $\sigma_H^2 = 0.05$ . . . . .	327
B.2 $T_C = 0.04$ . . . . .	328
B.3 White text: $\tilde{Y} = 0.83$ and $\sigma_Y^2 = 0.003$ . . . . .	329
B.4 $T_{Yb} = 0, T_{Yw} = 1$ . . . . .	329
B.5 Black text: $\tilde{Y} = 0.13$ and $\sigma_Y^2 = 0.001$ . . . . .	330
B.6 $T_{Yb} = 0, T_{Yw} = 0.2$ . . . . .	330
B.7 Blue text: $\tilde{H} = 2.56$ and $\sigma_H^2 = 0.034$ . . . . .	331
B.8 $T_C = 0.04$ . . . . .	331
B.9 Blue text: $\tilde{H} = 2.60$ and $\sigma_H^2 = 0.020$ . . . . .	332
B.10 $T_C = 0.05$ . . . . .	332
B.11 Blue text: $\tilde{H} = 2.73$ and $\sigma_H^2 = 0.049$ . . . . .	333
B.12 $T_C = 0.02$ . . . . .	333
B.13 Black text: $\tilde{Y} = 0.17$ and $\sigma_Y^2 = 0.008$ . . . . .	334
B.14 $T_{Yb} = 0, T_{Yw} = 1$ . . . . .	334
B.15 Black text: $\tilde{Y} = 0.19$ and $\sigma_Y^2 = 0.002$ . . . . .	335
B.16 $T_{Yb} = 0, T_{Yw} = 1$ . . . . .	335
C.1 For Proposed Method: [boolHL, medH, medY, varH, varY] = [1 3.6046, 0.3486, 0.0012, 0.0013]. . . . .	337
C.2 For Proposed Method: [boolHL, medH, medY, varH, varY] = [0, 6.0868, 0.7381, 0.0075, 0.0033]. . . . .	338
C.3 For Proposed Method: [boolHL, medH, medY, varH, varY] = [1, 6.0868, 0.3298, 0.0018, 0.0010]. . . . .	339
C.4 For Proposed Method: [boolHL, medH, medY, varH, varY] = [1, 0.2448, 0.3145, 0.0107, 0.0023]. . . . .	340
C.5 For Proposed Method: [boolHL, medH, medY, varH, varY] = [1, 6.0974, 0.5332, 0.0244, 0.0011]. . . . .	341

Figure	Page
C.6 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 6.1730, 0.7483, 0.0093, 0.0037]$ . . . . .	342
C.7 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.1145, 0.2670, 0.0080, 0.0028]$ . . . . .	343
C.8 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.1848, 0.2120, 0.0656, 0.0017]$ . . . . .	344
C.9 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 4.8869, 0.1329, 1.2905, 0.0029]$ . . . . .	345
C.10 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 3.6070, 0.1894, 2.3252, 0.0013]$ . . . . .	346
C.11 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 2.7925, 0.3618, 0.1469, 0.0028]$ . . . . .	347
C.12 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 1.0472, 0.2784, 2.6779, 0.0161]$ . . . . .	348
C.13 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 3.5358, 0.4344, 0.0016, 0.0028]$ . . . . .	349
C.14 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.7854, 0.3680, 0.0250, 0.0019]$ . . . . .	350
C.15 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 4.8171, 0.8821, 0.3069, 0.0046]$ . . . . .	351
C.16 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.0423, 0.3018, 0.0012, 0.0018]$ . . . . .	352
C.17 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.1309, 0.2317, 0.3181, 0.0093]$ . . . . .	353
C.18 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 4.0075, 0.1993, 0.0021, 0.0015]$ . . . . .	354
C.19 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 3.9924, 0.1886, 0.1030, 0.0014]$ . . . . .	355
C.20 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.1496, 0.3147, 0.0049, 0.0022]$ . . . . .	356
C.21 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 1.0472, 0.1529, 1.7701, 0.0005]$ . . . . .	357
C.22 For Proposed Method: $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 2.6180, 0.1305, 2.3481, 0.0019]$ . . . . .	358

Figure	Page
F.1 Top view of the setup environment. . . . .	375
F.2 Side view of the setup environment. . . . .	375
F.3 Graffiti Information Form. . . . .	376
G.1 Top view of the setup environment. . . . .	381
G.2 Image Recording Form. . . . .	382
G.3 Hazmat sign identifiers. . . . .	383
G.4 Example of Completed Image Recording Form for Figure G.6 (left). . .	383
G.5 Example of Completed Image Recording Form for Figure G.6 (right). .	384
G.6 Screenshots for hazmat sign found (left) and not found (right). . . . .	384



## ABSTRACT

Parra Pozo, Albert Ph.D., Purdue University, August 2014. Integrated mobile systems using image analysis with applications in public safety. Major Professor: Edward J. Delp.

One of the roles of emergency first responders (e.g. police and fire departments) is to prevent and protect against events that can jeopardize the safety and well being of a community. Examples include criminal gang activity and the handling and transportation of dangerous materials. In each of these cases first responders need tools for finding, documenting, and taking the necessary actions to mitigate the problem or issue.

The goal of this thesis is to develop integrated mobile-based systems capable of using location-based-services, combined with image analysis, to provide accurate and useful information to the first responders in real time. Two systems have been developed.

The first is a system to track and analyze gang activity through the acquisition, indexing and recognition of gang graffiti images. This approach uses image analysis methods for color correction, color recognition, image segmentation, and image retrieval and classification. A database of gang graffiti images is described that includes not only the images but also metadata related to the images, such as date and time, geoposition, gang, gang member, colors, and symbols. The user can then query the data in a useful manner.

The second is a system that can recognize and interpret hazardous material (hazmat) signs typically displayed by vehicles transporting dangerous materials. This approach uses image analysis methods for hazmat sign interpretation, including shape location detection and color recognition. The detection results are used to query an

electronic version of the Emergency Response Guidebook (ERG) and return information and advice to help first responders. A database of hazmat sign and scene images for forensic analysis is described that includes images and metadata.

# 1. INTRODUCTION

## 1.1 Problem Formulation and Challenges

One of the roles of public safety is to prevent and protect against events that can jeopardize the safety and well being of the community. These include criminal gang activity and handling and transportation of dangerous materials. In each of these cases first responders have the potential for finding and documenting evidence in real time. However, the number of actions that can be taken while on the streets are limited. If there is an incident and law enforcement officers need to compare information, they have to communicate with the corresponding police department.

For example, if gang graffiti is spotted by a first responder in an area, the information that can be obtained in situ is very limited. In the best case scenario, the user has expertise with gang graffiti interpretation and carries a camera. The only actions the user can take are reduced to taking an image and writing down some basic context information.

In a different scenario, a truck hauling a hazardous substance must carry a placard that helps identify the material and determine what specialty equipment, procedures and precautions should be taken in the event of an emergency. This information is contained in the Emergency Response Guidebook (ERG), published by the US Department of Transportation (DOT) [1]. As one might expect, the guidebook is large and requires precious time to search an index to determine the best way to handle a particular hazardous material.

The goal of this thesis is to develop integrated mobile-based systems capable of using location-based-services, combined with image analysis, to provide accurate and useful information to the first responders in real time.

## 1.2 Contributions of This Thesis

In this thesis two integrated mobile systems are described. First, a system for gang graffiti image acquisition and recognition. We called this system Gang Graffiti Automatic Recognition and Interpretation or GARI. GARI includes motion blur prevention and detection, color correction based on light sensor, color recognition based on touchscreen tracing, color image segmentation based on Gaussian thresholding, and content-based gang graffiti image retrieval. We have also investigated the design and deployment of an integrated image-based database system. Second, a system for hazmat sign detection and recognition. We called this system Mobile Emergency Response Guidebook or MERGE. MERGE includes segment detection using geometric constraints, convex quadrilateral detection based on saliency map, and sign location detection based on Fourier descriptors.

The main contributions of GARI and MERGE in the area of image analysis are as follows:

- We presented a motion blur prevention and detection method based on mobile device sensors.
- We presented a color correction method based on mobile device light sensor.
- We described a color recognition method based on touchscreen tracing.
- We presented a color image segmentation method based on Gaussian thresholding, block-wise Gaussian segmentation enhancement, background stripe removal, and connected component reconnection.
- We presented a feature extraction method based on local shape context descriptors from SIFT keypoint locations.

- We presented a gang graffiti content based image retrieval method based on bag-of-words model.
- We presented a segment detection method based on geometric constraints.
- We presented a convex quadrilateral detection method based on saliency map.
- We presented a sign location detection based on Fourier descriptors.

The main contributions of GARI and MERGE in the design and deployment of the integrated image-based database system are as follows:

- We developed an integrated image-based database system where data from users and images is connected to gang graffiti information for analysis and tracking.
- We developed an integrated image-based database system where data from users and images is connected to hazmat sign information for image analysis and forensics.
- We created a web-based interface for first responders and researchers to upload images and browse gang related information by location, date and time, using interactive maps for better visualization. It is accessible from any device capable of connecting to the Internet, including iPhone and Blackberry.
- We created a web-based interface for first responders and researchers to upload images and browse hazardous material information by location, date and time for forensic analysis. It is accessible from any device capable of connecting to the Internet, including iPhone and Blackberry.
- We created Android and iOS applications for first responders on the field to upload images to the server, use image analysis and conduct forensic tasks, browse related information, and use location-based services to populate interactive maps.

### 1.3 Publications Resulting From This Work

#### Conference Papers

1. Bin Zhao, **Albert Parra** and Edward J. Delp, “Mobile-Based Hazmat Sign Detection System,” *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 735-738, December 2013, Austin, TX.
2. **Albert Parra**, Bin Zhao, Joonsoo Kim and Edward J. Delp, “Recognition, Segmentation and Retrieval of Gang Graffiti Images on a Mobile Device,” *Proceedings of the IEEE International Conference on Technologies for Homeland Security*, pp. 178-183, November 2013, Waltham, MA.
3. **Albert Parra**, Bin Zhao, Andrew Haddad, Mireille Boutin and Edward J. Delp, “Hazardous Material Sign Detection and Recognition,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 2640-2644, September 2013, Melbourne, Australia.
4. **Albert Parra**, Mireille Boutin and Edward J. Delp, “Location-Aware Gang Graffiti Acquisition and Browsing on a Mobile Device,” *Proceedings of the IS&T/SPIE Electronic Imaging on Multimedia on Mobile Devices*, pp. 830402-1-13, January 2012, San Francisco, CA.

## 2. OVERVIEW OF GANG GRAFFITI AND HAZMAT SIGN DETECTION SYSTEMS

### 2.1 Overview of Gang Graffiti Systems

There are several methods that have been described to identify gang graffiti using feature matching as well as tracking gang graffiti using large databases. This section overviews the current methods describing their advantages and disadvantages.<sup>1</sup> We also compare some of the methods with GARI.

#### 2.1.1 Graffiti Tracker

Graffiti Tracker is a web-based system that began in 2002 [3]. It was designed to help first responders identify, track, prosecute and seek restitution from graffiti vandals. It is primarily used by law enforcement and public works agencies. The database contains more than 2 million manually analyzed graffiti images from 75 cities in two countries and nine states, mainly from the state of California.

The web-based services include graffiti analysis, interactive map browsing, graffiti storing and organization, and graffiti report. Graffiti Tracker provides clients with GPS-enabled digital cameras to generate reports of graffiti activity. The images can then be uploaded through the web interface to the database, where they are manually analyzed by trained analysts within 24 hours of submission.

The GPS coordinates of each image are used to build an interactive map where the user can view activity from individual vandals or monikers to specific crews or gangs. Gang trends or migration can be identified if the volume of graffiti for the same gang or vandal is large. A part from the interactive map, the user can browse the

---

<sup>1</sup>This chapter is an updated version of a chapter that appeared in [2].

stored graffiti by moniker, gang, type of incident, graffiti surface, or removal method. The information can be used to generate reports based on gang or moniker activity, such as total square feet of damage, locations of the incidents, or frequency of graffiti vandalism over a specific period of time.

### **2.1.2 TAGRS**

Tracking and Automated Graffiti Reporting System (TAGRS) is a system developed by 594 Graffiti, LLC in Irvine, CA in 2010 [4] in cooperation with the Orange County Sheriff's Department (OCSD)/Transit Police Services and Orange County Transportation Authority.

Government employees can access TAGRS through an Internet portal using a smartphone or PDA to input graffiti information including address, amount of damage, images of the graffiti and the date and time it was discovered. Law enforcement officers input their information through a secure intranet. After the graffiti data is entered it is sent to the investigator or analyst designed to handle graffiti offenses. Email support enables investigators to share information. TAGRS also reports on cost analysis and graffiti trends. Training for TAGRS takes about two hours.

The TAGRS program has helped solve more than 300 graffiti cases in Orange County since 2008.

The TAGRS application is provided at no cost, but any implementing agency is responsible for purchasing the hardware and services responsible to utilize and maintain the system. Once a client's device is registered in the TAGRS database it is a cross-mobile platform compatible with iOS, Blackberry and Android.

### **2.1.3 GRIP**

Graffiti Reduction & Interception Program (GRIP) is a graffiti and crime database developed by GRIP Systems in 1999 [5]. Graffiti experts, law enforcement and city management and infrastructure groups designed it.



GRIP allows a contractor to take an image and fill out a form detailing the image, and then send it to GRIP database for instant reading and analysis. An application for GRIP has been created using both iOS and Android. GRIP allows residents to send in images of graffiti from mobile devices or use their computers to email images and graffiti locations. GRIP offers free unlimited use of its database for six months.

Users can do their own data entry with GRIP's guidance, or can choose to use GRIP systems for entry work and analysis. There are multiple access levels including citizen, clerk, law enforcement agency, reader only or contractor.

#### **2.1.4 GTS**

The Graffiti Tracking System (GTS) is a system developed by Blue Archer in Pittsburgh, PA in 2005 [6]. It is a centralized, web-based application that enables multiple users to document instances of graffiti crime, manage investigations, track graffiti removal requests and compile actionable intelligence through the Internet.

GTS is designed for use by any organization that is fighting graffiti crime including law enforcement, prosecutors, public works departments, railways, and local and state officials.

Features of the GTS include tracking an unlimited number of graffiti incidents, uploading an unlimited number of photos per incident; intelligent searching of all GTS records; automatic linking of similar incidents to develop actionable intelligence; tracking of unlimited number of suspects, witnesses and contacts per incident; automated notification of new incidents based on user-defined filters; fully customizable drop-down menus to record incident criteria.

#### **2.1.5 GAT**

Graffiti Abatement Tool (GAT) is a system developed by the Public Works, Police, and Information Technology departments in Riverside, CA in 2007. This system is not currently commercially available. GAT was developed to coordinate inter-

departmental efforts and address the problem of connecting instances of graffiti to an individual vandal or tagger. It stores and manages images of graffiti with other tabular data. It is claimed that GAT is useful in tracking, prosecuting and suing taggers.

Public Works crews that remove graffiti take a picture of the tag using a GPS camera and complete a customized digital form on the camera including basic information about the incident. The images and data are uploaded onto a server that automatically adds the data to an online database. Graffiti images can be matched with other instances of graffiti by the same tagger.

GAT allows the total cost of graffiti to be estimated. When the Public Works abatement crew removes the graffiti, the cleanup method and materials used as well as how much time was required are entered. The cost associated with prosecuting and suing a tagger in a civil lawsuit is entered by the city attorney. GAT allows for the construction of a chain of evidence for the prosecution. There are more than 200,000 images and associated information in Riverside's central police database, with the number increasing by up to 500 per week. Nearly 83,000 instances of graffiti have been removed since January 2009.

#### **2.1.6 TAG-IMAGE**

Tattoo and Graffiti Image-Matching and Graphic Evaluation (TAG-IMAGE) is a system developed by the Federal Bureau of Investigation (FBI) Biometric Center of Excellence (BCOE) in Clarksburg, WV in 2012. The system, which is not currently commercially available, is a collaboration with the Cryptanalysis and Racketeering Records Unit (CRRU) of the FBI's Laboratory Division.

TAG-IMAGE is an image-comparison system designed to help the CRRU match images within its database to determine the significance of tattoos, graffiti or other cryptic symbols for FBI investigative programs dealing with foreign or domestic terrorism, violent crime or gangs.

TAG-IMAGE uses image-to-image technology to match symbols based on appearances. A user emails an image to the CRRU where an analyst enters it into the system. The system then compares the image against images stored in the CRRU database. When the search is completed a CRRU analyst emails a response to the user, including associated details and contact information. The submitted image becomes available for future comparisons by other agencies.

TAG-IMAGE is currently in pilot phase and will become available to local, state, tribal and federal law enforcement and correctional agencies when the pilot phase ends. The BCOE also plans to conduct a small operational pilot program with the National Gang Intelligence Center to determine the feasibility of image-based matching and to gain user feedback.

### **2.1.7 Graffiti-ID**

Graffiti-ID is an ongoing project (since 2009) at Michigan State University [7, 8]. The project is focused on matching and retrieval of graffiti images. There is similar work from the same team on gang tattoo identification, called Tattoo-ID [9–14].

The goal of Graffiti-ID is to identify gang/moniker names related to a graffiti image, based on visual and content similarities of graffiti images in a database. Figure 2.1 shows a block diagram of the system. There are two modules, one for populating the database (offline) and another for querying and obtaining results from the database (online). The offline module includes two processes. First, automatic feature extraction using the Scale Invariant Feature Transform (SIFT) [15]. Second, manual annotation of graffiti images by letters and numbers. This is done on images taken from an external gallery of images with the information stored in a database. The online modules includes manual annotation of input images to filter the database and SIFT feature extraction to obtain keypoint matching.

The image database used is based on the Tracking Automated and Graffiti Reporting System (TAGRS) from the Orange County Sheriff Department in California.

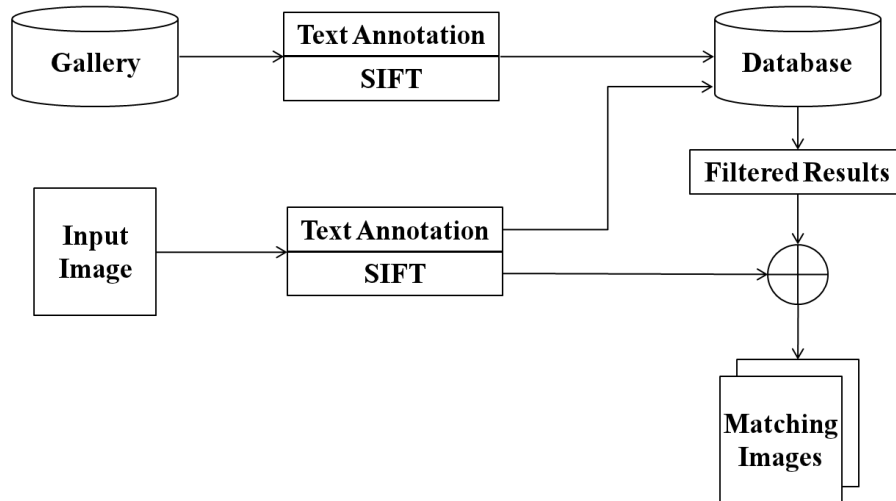


Fig. 2.1.: Block Diagram of the Graffiti-ID System.

The database consists of 64,000 graffiti images the main sources of the images are the Orange County Transportation Authority and crime reports. A subset of 9,367 images were used for evaluation. Each of these images contains up to four information parameters: moniker, gang, date and time, and address.

The Graffiti-ID system was tested using graffiti images from the original database subset. The retrieval accuracy was evaluated using Cumulative Match Scores (CMS) [16]. The graffiti images were used as query for the manual annotation matching step, which returns candidate images from the database that match the text description (presence of letters and numbers). SIFT features from the input image and compared against SIFT features from each of the candidate images. The candidates that best match the SIFT features of the query, given the Cumulative Match Scores, are returned to the user. Table 2.1 shows performance results of the output of the second step. The rank- $k$  accuracy refers to the percentage of queries for which the correctly matched images are found within the  $k$  candidate images.

Table 2.1: Accuracy and execution time for various numbers of candidate images from the manual annotation matching step.

<b>Candidate Images</b>	300	500	1,000	9,367
<b>Rank-30 accuracy</b>	63.8 %	65.4 %	66.5 %	64.3 %
<b>Retrieval Time (seconds/query)</b>	12.4 s	20.1 s	39.8 s	415.7 s

### 2.1.8 Other Work on Graffiti and Tattoos

There exist other methods in the literature that use image analysis techniques on graffiti and tattoo images. In [17] methods for segmenting and retrieving graffiti images are described using global thresholding and template matching. The system consists of two main components: character detection and string recognition and retrieval. The character detection process includes image preprocessing and binarization, text detection and image refinement. The string recognition and retrieval process is further subdivided into two modules: image-wise retrieval and semantic-wise retrieval. The image-wise retrieval includes bounding-box extraction and interest point matching. The semantic-wise retrieval includes bounding-box extraction, character recognition and string matching. The results of the image-wise retrieval and semantic-wise retrieval modules are combined to produce the final output. The experimental results on a database of 194 graffiti images show a retrieval accuracy of 88% when using the proposed bounding box framework.

In [9–14] the authors describe image retrieval approaches for tattoo images, Tattoo-ID. The goal of Tattoo-ID is to create a content based image retrieval system to find images from a database that are related to the query image. The image analysis methods used are very similar to those in Graffiti-ID, including SIFT keypoints and the use of a matching technique to measure visual similarities. The system was tested in a database of 100,000 tattoo images. The retrieval accuracy was 85.6%, with an average retrieval time of 191 seconds on an Intel Core 2, 2.66GHz and 3GB RAM processor.

In [18] the authors propose a tattoo retrieval system using a combination of existing image retrieval techniques. Figure 2.2 illustrates the system. The experimental results on a dataset of more than 300,000 tattoo images show a retrieval accuracy of 85% in the best case. The running times depend on the database used, and range from 145ms to 5 seconds on an Intel i7-930 using 4 cores with 2.8GHz and 8GB of main memory.

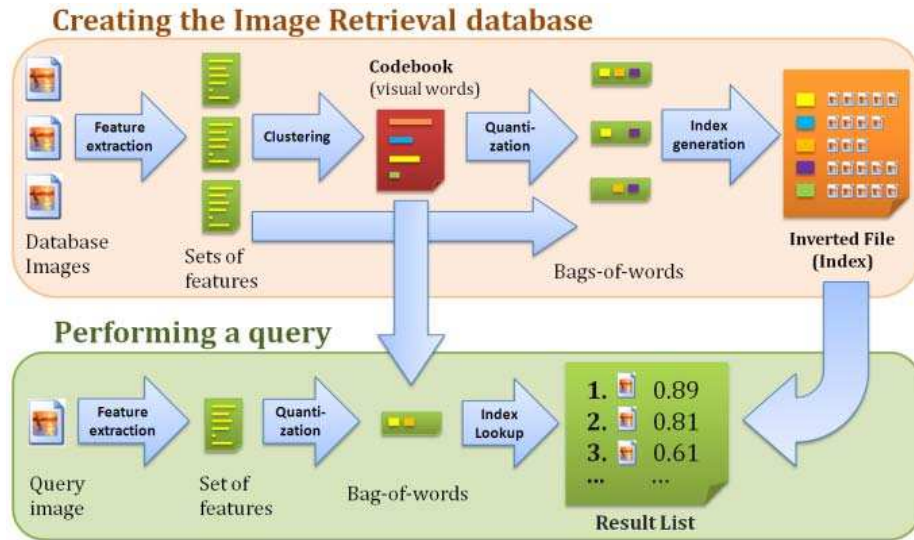


Fig. 2.2.: Block Diagram of The System in [18].

### 2.1.9 Comparison to GARI

Although our proposed system (GARI) shares some goals with the above systems, our methodology is different. Table 2.2 summarizes a comparison between the features of the various Gang Graffiti Systems described above.

We present a detailed comparison between the most similar systems to GARI: Graffiti-ID and Graffiti Tracker. We then compare the image analysis methods used in other work on graffiti and tattoos to the ones used in GARI.

#### GARI vs. Graffiti-ID vs. Graffiti Tracker

Both Graffiti-ID and GARI have goals of identifying gangs and gang members based on the graffiti content. Graffiti-ID uses SIFT features between an input image and images from the database. GARI currently uses color recognition techniques, along with metadata information from an image to query the database. GARI uses SIFT features to detect if an image of a same graffiti was already acquired at a specific location. GARI also uses shape techniques to detect graffiti components. By graffiti

Table 2.2: Comparison of features between different gang graffiti systems and GARI.

<b>Feature</b>	<b>GARI</b>	<b>Graffiti-ID</b>	<b>Graffiti Tracker</b>	<b>TAGRS</b>	<b>GRIP</b>	<b>GTS</b>	<b>GAT</b>	<b>TAG-IMAGE</b>
Used in field	YES	NO	YES	YES	YES	YES	YES	YES
Graffiti location	IN, IL	CA	CA	CA	CA, CO	CA, PA	CA	-
Images in database	1,000	6,000	+4 million	-	-	-	200,000	-
Analysis (time)	Seconds	-	24h	-	-	-	-	-
Analysis (method)	Semiautomatic	Automatic	Manual	Manual	Manual	Manual	Manual	Semiautomatic
Web version	YES	NO	YES	YES	YES	YES	YES	NO
Mobile version	YES	NO	NO	YES	YES	NO	NO	NO
Device	Smartphone	-	GPS Camera	Smartphone	Smartphone	-	GPS Camera	-
Interactive Map	YES	NO	YES	NO	NO	NO	YES	NO
Price	\$10,000 server	-	\$6,000/year	\$15,000 server	\$987/year	-	-	-



components we mean the objects and shapes contained in a graffiti image, such as stars, pitchforks, crowns, and arrows.

Both Graffiti Tracker and GARI keep track of gang activity based on GPS tags from the images and the graffiti content. However, all the image analysis in Graffiti Tracker is done manually, while the only user input on GARI is the touchscreen tracing for color recognition.

Graffiti-ID does not exploit the first responders action in the field, such as capture and upload images to a server or browse the database from a mobile device; the analyzed images are on the server. Graffiti Tracker allows users to acquire images only with GPS-enabled cameras they provide and the images have to be transferred to a computer and sent to the server. GARI allows the users to take images with any camera, and the GPS coordinates are automatically extracted from the EXIF data of the image or inserted manually when uploaded to the server (i.e., by GPS coordinates or by address through reverse geocoding [19]). Moreover, GARI has a mobile application that allows the user to take an image with a smartphone and send it to the server in situ. GARI also allows the first responder to browse the database of graffiti. GARI allows the user to upload images to the server through a web-based interface from any device capable of connecting to the Internet.

In Graffiti Tracker image analysis is done manually by trained analysts with the results obtained within 24 hours of submission. GARI currently does the analysis in the field, automatically and in real-time, either on the device or on the server. Graffiti-ID uses, as GARI, SIFT features to match images on the server automatically, but the analysis of the content of the graffiti is done manually, by labeling the image. It just allows labels to be numbers (0-9) or letters (a-z), not symbols or other features such as color.

Graffiti-ID does not provide any type of gang activity tracking, while both Graffiti Tracker and GARI provide interactive maps that allow first responders to browse the database and keep track of specific gangs or individuals. The advantage of GARI is that it also provides additional methods for tracking gang activity, including browsing

the database by radius from specific locations, or by graffiti color. One advantage of Graffiti Tracker is that its database is currently dramatically larger than the GARI database. Therefore, the results retrieved from the Graffiti Tracker database can indicate more accurate gang activity.

In summary, our system combines features from both Graffiti-ID and Graffiti Tracker, and adds more services and functionality. The advantages of our system over Graffiti-ID and Graffiti Tracker are the following. We provide a mobile application to be used by first responders in the field, where they can capture, upload and browse graffiti images from the database. The image acquisition in our system is device independent; virtually any image type from any camera make and model can be uploaded using one of our supported platforms: Android, iOS, and web-based interfaces.

### **GARI vs. Other Work on Graffiti and Tattoos**

The work in [17] is the only method from our review that propose automatic segmentation of the graffiti components from the background. However, while GARI does color image segmentation based on touchscreen tracing, [17] uses local thresholding techniques such as Niblack [20] as a preprocessing step to binarize the image. Also, GARI uses SIFT features for graffiti component image retrieval (see Section 3.7), while [17] uses a template matching method that just considers letters and numbers.

The GARI system uses a vocabulary tree based on a bag-of-words model for content based image retrieval (see Section 3.7). The approaches described in [10, 11] do not use the bag-of-words models and report slower matching and retrieval times than we demonstrate in our experiments (see Section 5).

Finally, although [18] does use a bag-of-words model for image retrieval of gang and gang-like tattoos, the system is not intended for real-time retrieval in mobile-based environments.

## 2.2 Overview of Hazmat Sign Detection Systems

Although there exist several mobile-based applications that provide easy access to the Emergency Response Guidebook (ERG) guidebook [1, 21], they only provide manually browsing functionality. Several methods in the literature deal with sign location detection and recognition (see Section 4.1), but we are only aware of two other published papers with application to hazmat signs [22, 23].

### 2.2.1 Hazmat Sign Detection Based on SURF and HBP

In [22] the hazmat sign detection is done using color histogram back-projection (HBP) and Speeded Up Robust Feature (SURF) [24] matching. The method was implemented and tested on an autonomous mobile robot for the 2008 RoboCup World Championship. Histogram back-projection is used to detect regions of interest in the image and remove the background of the scene. A background image without a sign,  $h(x, y)$ , is used as a ground-truth to isolate the hazmat sign when it appears on the scene and an image of it is captured,  $f(x, y)$ . This is done by determining the euclidean distance of the color coordinates of each pixel within  $h(x, y)$  and the corresponding pixel within  $f(x, y)$ . A threshold  $K$  is used to create a binary mask of the hazmat sign by the use of an indicator function  $\delta(x, y) = \{(x, y) \text{ s.t. } |f(x, y) - h(x, y)| > K\}$ . Several color histograms are then estimated for the U and V channels on the YUV color space, and summed up to create a single histogram  $H_o(U, V)$  for every sign on the image. A threshold  $\theta(H_o, \epsilon)$  is used for  $H_o(U, V)$ , resulting in a binary indicator function  $\pi_o(U, V)$ , which specifies which pixels form part of a sign. The value of  $\epsilon$  is manually set to 0.05. Finally, morphological filters are used to segment the masked regions from the background and create one or more regions of interest to be used as inputs to the matching process using SURF features.

SURF matching is used to find interest points and retrieve images from a database. After the region of interest is determined from the image containing a hazmat signs, multiple interest points are found using SURF. Interest points surrounding regions

that overlap the region of interest are discarded, since they do not provide enough information about the sign. For the remaining interest points, their corresponding feature vectors are matched against all features of all images in a database corresponding to the colors found on the first step.

The experiments were done using a stereo camera system consisting of two cameras with a resolution of  $1024 \times 768$  pixels. The tests consisted of detecting five different hazmat signs in 240 images. The images were taken at 1, 1.5 and 2 meters, with a maximum distortion of  $30^\circ$ . The results show a detection accuracy of 92% from 1 meter, 52% from 1.5 meters, and less than 20% from 2 meters. The running time ranges from 1 to 1.6 second on a 2.7GHz Intel CPU.

### 2.2.2 Hazmat Sign Detection Based on HOG

In [23] hazmat sign detection using sliding windows and Histogram of Oriented Gradients (HOG) [25] is described. The method was implemented and tested on a wheeled USAR robot for the 2010 RoboCup World Championship.

The authors use the sliding window approach to exhaustively scan every pixel over a range of positions and scales, with steps of 8 pixels and relative scale factors of 1.05. For each position and scale a discriminative Support Vector Machine (SVM) classifier is used to make binary decisions about the presence or absence of an object. In order to describe the contents of the image at each particular location a HOG descriptor is used along with color histograms in the Lab color space to distinguish between multiple hazmat signs. For each hazmat sign hypothesis of the HOG based detector, the color histogram is used to do the final classification by applying a k-nearest neighbor approach in combination with  $\chi^2$ -distance.

The experimental results show a recognition rate of 37.5% using histograms based on entire sliding windows and a recognition rate of 58.3% using sub-region based histograms. Region-based histograms provide better representation of the image since

they are capable of capturing the spatial distribution of colors within the detection window.

### 2.2.3 Comparison to MERGE

Although all methods above are deployed on mobile environments, MERGE is intended for real-time use by first responders, while [22] and [23] were intended for use in a very specific context. The sign detection method proposed in [22] uses a ground-truth image of the background to aid in detection when the hazmat sign appears. This is not a feasible assumption in MERGE, since the first responders are expected to take images of hazmat signs in a large variety of scenarios. In [23] a dataset of 1,480 daylight images is used for both people and hazmat sign recognition. However, the authors do not specify how many images contain hazmat signs, or at what distances the signs are located. They do not provide information about the resolution of the images or the cameras used for acquisition. In MERGE no assumptions on the background are made in order to detect the sign. Instead, color information is used to detect candidate regions using a saliency map model.

Once the hazmat sign is detected [22] uses image matching based on SURF features, and [23] uses HOG and color histogram descriptors, both being very time consuming task. This step is not done in MERGE. Currently, the color of the hazmat sign is considered to be uniform, and the detection is made at different color channels. The recognition of non-uniformly-colored placards is presented as part of the future work (see Section 6).

The goal of MERGE is to be able to detect hazmat signs at long distances (up to 500 feet). Our experimental results show successful detections in some cases at more than 100 feet. However, the experiments in [22] can only be considered successful at 1.5 meters, and the accuracy reported by [23] is very low. Finally, the execution time of the sign detection method in MERGE is 0.84 seconds on average, faster than the sign detection method in [22]. No execution time is reported in [23].

## 2.3 Proposed Systems

Two systems have been developed. First, a system to track and analyze gang activity through the acquisition and recognition of gang graffiti images. This approach uses image analysis methods for color recognition, image segmentation, and image retrieval and classification. A database of gang graffiti images is maintained on a server, and includes not only the images, but metadata related to them, such as date and time, geoposition, gang, gang member, colors, or symbols. The user can then query the data in a useful manner. We call this system Gang Graffiti Automatic Recognition and Interpretation or GARI [26]<sup>2</sup>.

Second, a system to recognize and interpret hazardous material (hazmat) signs typically displayed by vehicles transporting dangerous materials. This approach uses image analysis methods for hazmat sign interpretation, including shape detection based on saliency maps, color recognition and sign interpretation. The detection results are used to query an electronic version of the ERG and return information and advice to help first responders. We call this system Mobile Emergency Response Guidebook or MERGE [27].

### 2.3.1 GARI

Gangs are a serious threat to public safety throughout the United States. Gang members are continuously migrating from urban cities to suburban areas. They are responsible for an increasing percentage of crime and violence in many communities. According to the National Gang Threat Assessment, approximately one million gang members belonging to more than 33,000 gangs were criminally active within the United States as of April 2011 [28], an increase of 13,000 since April 2008 [29]. Criminal gangs commit as much as 80% of the crime in many communities according to law enforcement officials. Gang graffiti is their most common way to communicate

---

<sup>2</sup>Parts of the work on GARI was done with my Purdue colleagues Andrew Haddad and Professor Mireille Boutin.

messages, including challenges, warnings and intimidation to rival gangs. It is an excellent way to track gang affiliation and growth or to obtain membership information.

Our goal is to develop a system, based on a mobile device such as a mobile telephone, capable of using location-based-services, combined with image analysis, to automatically populate a database of graffiti images with information that can be used by law enforcement to identify, track, and mitigate gang activity. The first step towards this goal was to create a system that includes the ability to acquire images in the field using the camera in a mobile telephone and a networked back-end database system that uses the metadata available at the time the image is acquired (geoposition, date and time) along with some basic image analysis functions (e.g. color features) [2].

The next step is to extend the image analysis to include segmentation, matching, retrieval and classification of gang graffiti images and gang graffiti components. By gang graffiti components we mean the objects and shapes contained in a gang graffiti image, such as stars, pitchforks, crowns, and arrows.

Apart from being able to send and retrieve multimedia data to the database, the first responder can take advantage of location-based-services. The information in the database of gang graffiti can be queried to extract information based on parameters such as date and time of capture, upload or modification of the graffiti image, or radius from a given location. The data includes not only the images, but information related to it, such as date and time, geoposition, gang, gang member, colors, or symbols.

We have implemented these features both as applications for Android and iOS<sup>3</sup> hand-held devices and as a web-based interface for any device capable of connecting to the Internet (e.g., desktop/laptop computer, Blackberry).

GARI also includes features for gang tattoo analysis [30]. By providing first responders with this capability, the process of identifying and tracking gang activity can be more efficient.

---

<sup>3</sup>The iOS application was developed with my Purdue colleague Joonsoo Kim.

## System Overview

Figure 2.3 illustrates a block diagram of the GARI system. It shows the various services available, both on the device (no network connection required) and on the server (network connection required). These services include capturing images of gang graffiti, automatic analysis and labeling (such as geolocation, date/time, and other EXIF (Exchangeable Image File Format) [31] data obtained from the image), uploading images to the database of gang graffiti, and querying the database to filter and browse its contents.

Figure 2.4 illustrates the modules of our image analysis system. Note that the modules in bold are currently implemented on the server.

When a first responder uses the mobile device to capture an image we use a customized camera with blur motion prevention (Section 3.2). The image is color corrected on the device using data from the light sensor (Section 3.3) and the user is given several options. The image can be uploaded to the server and added to the database of gang graffiti. If so, we extract EXIF data from the image, such as geolocation and date and time, in order to identify the image and its location. The color recognition module allows the user to detect the color of a graffiti component by tracing a path using their finger on the device's touchscreen (Section 3.4). The color recognition is done entirely on the device and extra data is obtained for color image segmentation from the server (Section 3.5). The content-based image retrieval module finds matches for each segmented graffiti component (Sections 3.7 and 3.6). The captured image can be used to find similar images in the database using the scene recognition module (Section 3.7). The results from the scene recognition and the graffiti component retrieval are sent back to the user. All the data from the different modules can be sent to the server along with the graffiti image, and added to the database to be browsed or analyzed in the future.



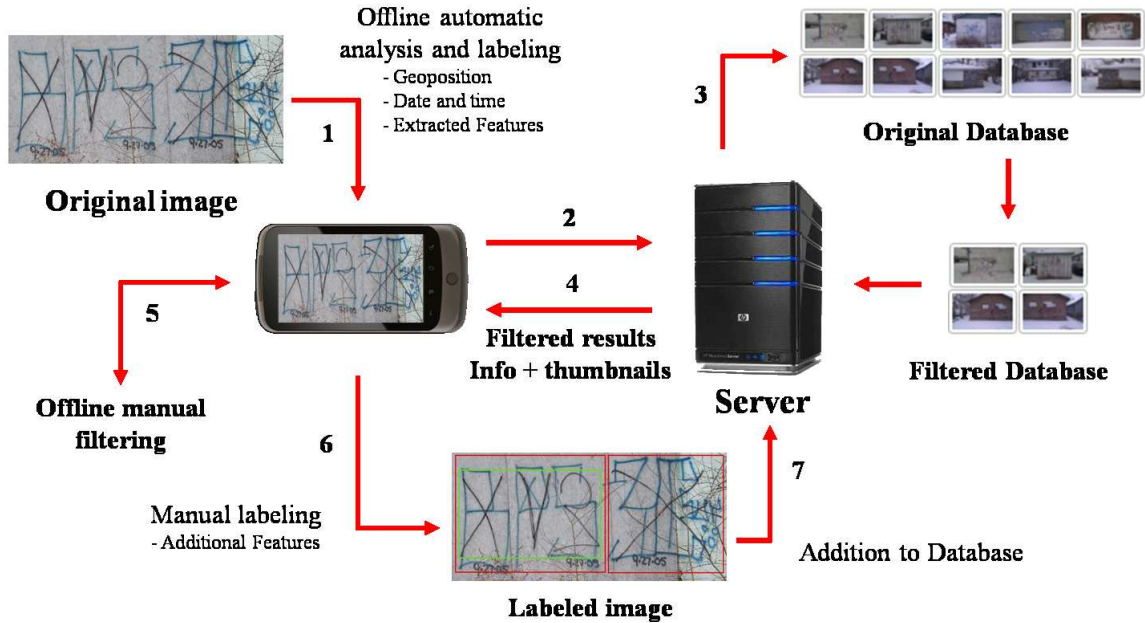


Fig. 2.3.: Block Diagram of the GARI System.

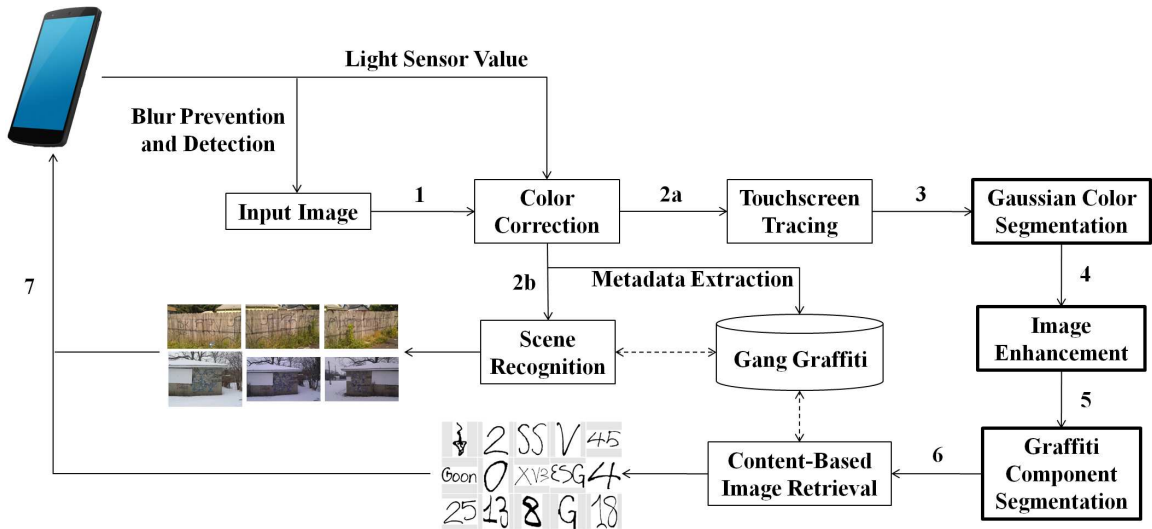


Fig. 2.4.: Modular Components of the GARI System.

## Interpretation of Gang Graffiti

Gangs have used street graffiti to communicate with each other for a long time [32, 33]. It is their most common way to communicate messages, including challenges,

warnings or intimidation to rival gangs. If graffiti are correctly interpreted, they are a great source of information that can be used to track gang affiliation and growth, or to obtain membership information.

It is worth noting the differences between “graffiti” terms that we use throughout this thesis.

- **Gang:** We use the word gang to refer to a street gang, defined by [34] as a “self-formed association of peers, united by mutual interests, with identifiable leadership and internal organization, who act collectively or as individuals to achieve specific purposes, including the conduct of illegal activity and control of a particular territory, facility, or enterprise”.
- **Gang member:** To be distinguished from a tagger. Gang members paint graffiti to mark territory, threaten other gangs or honor other gang members. In contrast, **taggers** paint graffiti to defy authority, or to obtain recognition or notoriety.
- **Gang graffiti:** To be distinguished from tagging. Gang graffiti are simple and usually monochromatic. In contrast, **tags** are artistic and colorful.
- **Component:** Any of the separable elements in a graffiti, such as symbols, acronyms, or numbers.
- **Blob:** Area of the graffiti containing only one component. Useful to identify relative positions of components to each other in the same graffiti.
- **Clique:** Subset of a larger gang with their own name, which may have connection to the gang’s neighborhood (e.g., street name, geographic location). Cliques are local, while gangs extend nationally or internationally. Also known as **factions** or **crews**.
- **Turf:** Slang for territory, or area of influence, specific in this thesis to gangs. Term used when talking about a fight between gangs for territory or power, also

known as a turf war, usually with the objective to gain control over the drug market in a specific area.

In the following subsections we describe how to interpret gang graffiti from its contents, including colors, shapes and structure. We also describe how gangs and gang members can be tracked from the graffiti contents and their location. Finally, we illustrate some examples on how a first responder can do the interpretation and tracking easier and faster. Note we are not claiming in this thesis to be an expert in the interpretation of gang graffiti. Our knowledge is limited. We are relying on law enforcement experts for the GARI project. <sup>4</sup>

### **Some Examples of Interpretation**

Gang graffiti can be considered a low-level language used by gangs to communicate with each other. The alphabet of this “language” consists not only of letters (Aa-Zz) and numbers (0-9) but also of symbols (e.g., stars, crowns, arrows) and colors. The contents of gang graffiti are simple and straightforward. Gangs usually paint handwritten graffiti using a single color (perhaps two at most). Gang graffiti do not contain complete sentences, but words, short phrases, abbreviations and acronyms (e.g., gang and gang member names, street names and numbers). As is the syntax in a regular language, the relative position and alignment of each component is important in the general structure of the graffiti. The syntax in gang graffiti is two-dimensional. For example, the meaning of a symbol is different if it is painted at the top right of a graffiti or if the symbol appears upright or upside down. Figures 2.5 and 2.6 illustrate some examples of gang graffiti alphabet, syntax, and color.

---

<sup>4</sup>The images shown in this thesis were obtained in cooperation with the Indianapolis Metropolitan Police Department (IMPD). We gratefully acknowledge their cooperation in GARI.



Fig. 2.5.: Examples of Graffiti Elements.

We will use Figure 2.5 as examples for interpreting gang graffiti. Figure 2.5a is a black gang graffiti. This particular color does not eliminate any gang from being the author of the painting. The 6-point star refers to the Folk Nation, one of the two “nations” to which most gangs belong. Each point means: love, life, loyalty, wisdom, knowledge, understanding. The numbers on both the left and the right of the star, 7



Fig. 2.6.: Examples of Graffiti Color Recognition.

and 4, refer the 7th and 4th letters of the alphabet, G and D, respectively. That is, the Gangster Disciples gang. The three-pointed pitchfork is another sign of the Folk Nation. In this particular case, two upright three-pointed pitchforks make a total of six points, making reference to the 6-pointed star. Moreover, the inscription below the star makes reference to the clique with the street name, *2-8th st* or 28th street, and the nickname of the gang member who painted the graffiti, *Ruthless*.

Figure 2.5b is a black gang graffiti containing the name of a clique, as usual taking its name is taken from the street where they operate. In this case, it refers to the 42nd Street Gang from Indianapolis. The color itself does not indicate anything concerning which gang this clique may belong to.

Figure 2.5c is a blue gang graffiti with a 6-point star similar to the one in Figure 2.5a. The blue color is used by the Gangster Disciples (and others). The numbers on the sides of the star, along with the additional letters at its bottom make it clear that

this graffiti makes reference to the Gangster Disciples. The number 6 in the center of the star is also an extra remainder of the Folk Nation.

Figure 2.5d is a red and black gang graffiti containing the name of a gang/cliue in red, Goon Squad (also spelled *Goon Sqaud* or *Goun Sqoud*). This gang/cliue name is very common, since it originally refers to a group of thugs or mercenaries associated with violent acts. With the little information from this graffiti it is not possible to determine which gang they belong to or if they are a gang themselves. However, the use of the red color seems to be related to the People Nation, although there are gangs from the Folk Nation that also use the same color. Below the gang name we find the name of the neighborhood where the gang operates (i.e., *Brightwood 2-5st* or Brightwood 25th Street, Indianapolis) in black. The two down arrows at each side of the gang name express turf dominance. The inscription at the very bottom, also in black, appears to be the nickname of the gang member who painted the graffiti, *7MOB*, also known as “Brightwood 7 M.O.B. Bitch.” There is an additional down arrow, again expressing turf dominance of this particular gang member.

Figure 2.5d is a simple black gang graffiti containing the acronym *ESG*, referring to the East Street Gang in Indianapolis.

Figure 2.5f is a multicolor gang graffiti. It seems the blue graffiti was painted over the black graffiti. The black graffiti is very similar to the one in Figure 2.5c, belonging to the Gangster Disciples. The 28th Street cliue name, along with the nickname *Ruthless*, are also painted next to the 6-point star. The blue graffiti contains the name of a different cliue, the *25th Hillside*, from Hillside Avenue in Indianapolis. The inscription at the very bottom, in blue, could make reference to an insult to the gang or gang member who painted the black graffiti originally, however the upside-down 5-point star indicates disrespect for the People Nation. Therefore, both the black and the blue graffiti have been painted by gang members of Folk Nation’s gangs, and the blue inscription to the left of the upside-down 5-point star is the nickname of a gang member of the 25th Hillside cliue, from the Folk Nation.

### 2.3.2 MERGE

Hazardous materials can react differently to environmental stimuli and cause problems in accidents and emergency situations and therefore makes these materials particularly dangerous to civilians and first responders. A federal law in the US requires vehicles transporting hazardous materials be marked with a standard sign (i.e., a “hazmat sign”) identifying the type of material the vehicles is carrying [35]. These signs have identifying information described by the sign shape, color, symbols, and numbers.

Our goal is to develop a system, based on a mobile device such as mobile telephone, capable of using location-based-services, combined with image analysis, to automatically detect and interpret hazmat signs from an image taken by a first responder<sup>5</sup>.

This system includes the ability to acquire images in the field using the camera in a mobile telephone and a networked system that uses the metadata available at the time the image is acquired (geoposition, date and time) along with image analysis functions to interpret one or multiple hazmat sign on the same image.

The interpretation of the signs includes the association of the sign contents to a guide page on the ERG [1,21]. The information in the book determines what specialty equipment, procedures and precautions should be taken in the event of an emergency related with such chemical component.

Apart from being able to send and retrieve multimedia data to the server, the first responder can take advantage of location-based-services. The location information acquired through the mobile phone can be used along with the interpretation of the hazmat sign to provide the first responder with the best way to handle a particular hazardous material. This is done by projecting an action radius on a multimedia map on the hand-held device, so that the first responder can take the necessary actions to evacuate the affected area. The action radius takes into account real-time weather

---

<sup>5</sup>Parts of this work was done with my Purdue colleagues Bin Zhao, Andrew Haddad, He Li, Khariththa Thongkor and Professor Mireille Boutin.

information (i.e. wind speed and direction) to provide more accurate evacuation information.

We have implemented these features both as an application for Android hand-held devices and as a web-based interface for any device capable of connecting to the Internet (e.g., desktop/laptop computer, iPhone, Blackberry).

By providing first responders with this capability, the process of identifying and protecting citizens against hazardous materials can be faster and more efficient.

## **System Overview**

Figure 2.7 illustrates a block diagram of the MERGE system. It illustrates the various services available, both on the device (no network connection required) and on the server (network connection required). These services include capturing images of hazmat signs, uploading images to the server for automatic analysis, and querying an internal database containing a digitized version of the ERG [1, 21].

There are two basic operation modes: analysis of a new image and internal database browsing. The first mode includes capturing or browsing for an existing image on the hand-held device, uploading the image to the server and using sign detection and interpretation methods (Section 4). The results sent back to the user include the detected hazmat signs and a link to a guide page from the internal database containing the necessary information in case of an emergency. The second mode includes browsing an internal database to obtain information about the hazmat sign. The internal database can be browsed by UN number, class, symbol, or color (Section 4.5).



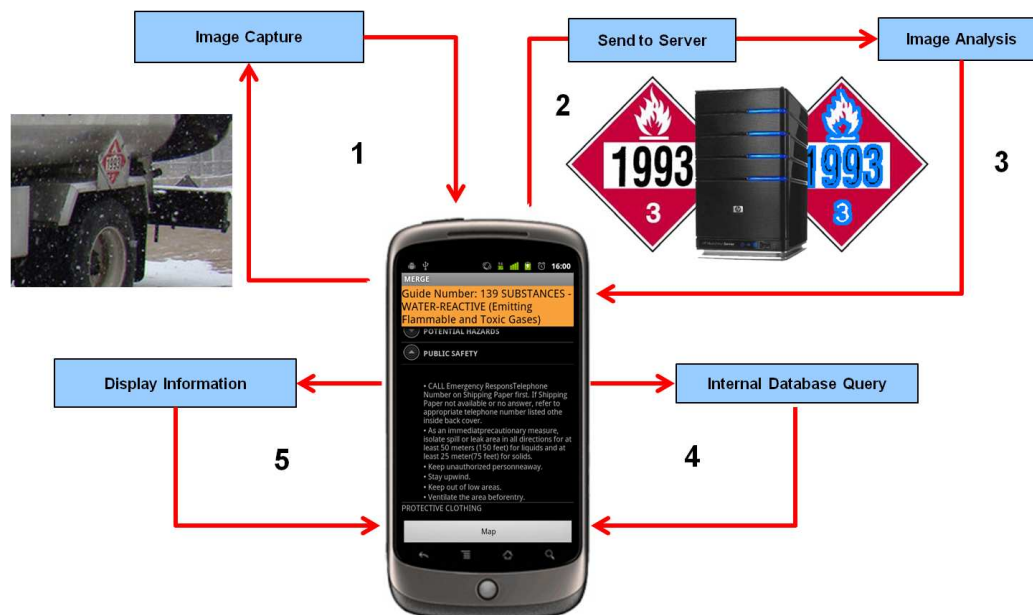


Fig. 2.7.: Block Diagram of the MERGE System.

## Interpretation of Hazmat Signs

Hazmat signs are characterized both by their shape and contents. Figure 2.8 illustrates possible shapes for hazmat signs, from which we only consider the diamond-shaped signs, or placards. Inside the placard there are four elements that uniquely identify the chemicals inside the container. Figure 2.9 illustrates three of the elements.

- UNID:** The United Nations Identification number (UNID) consists of a four-digit number used world-wide in international commerce and transportation to identify hazardous chemicals or classes of hazardous materials. UNID numbers range from 0001 to about 3500 and are assigned by the United Nations Committee of Experts on the Transport of Dangerous Goods. the UNID provides the user a direct link to the ERG guide page containing information on the placard of interest.
- Symbol:** The graphics and text in the placards representing the dangerous goods safety marks are derived from the UN-based system of identifying dan-

gerous goods. A comparison of symbols in the database will inform the user which guide page is associated with the symbol in the image. The possible symbols (shown in Figure 2.10) are: Corrosive, Explosive, Flammable, Gases, Infectious, Oxidizing, Pollutant, Radioactive, Toxic.

- **Class number:** Following the UN Model, the Department of Transportation divides regulated hazardous materials into nine classes, some of which are further subdivided. The class number on the placard provides the user a number of possible ERG guide pages. The possible classes are: Explosives, Gases, Flammable Liquids, Flammable Solids, Oxidizing Substances, Toxic Substances, Corrosive Substances, Miscellaneous Hazardous Materials.
- **Color:** The color of the hazmat also gives information about the chemical being hauled. The hazmat colors are red, blue, yellow and white. Red is for flammability, blue indicates health hazards, yellow is for reactivity and white is for personal protection. Figure 2.11 shows some possible combinations of colors on hazmat signs.

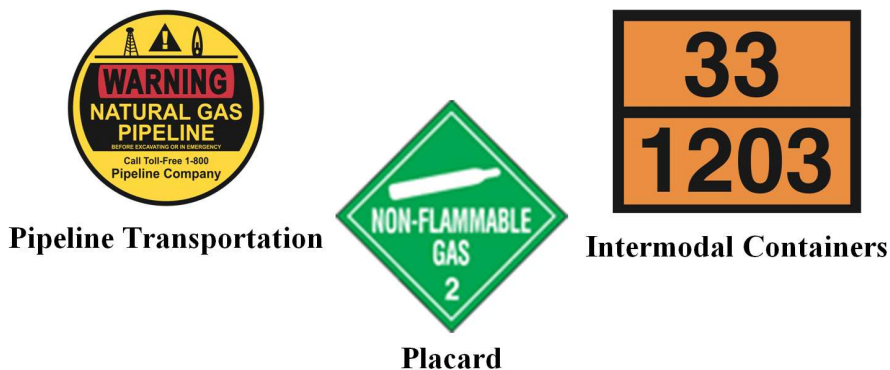


Fig. 2.8.: Possible Shapes of Hazmat Signs.



Fig. 2.9.: Elements That Uniquely Identify a Hazmat Sign. From Left to Right: UNID, Symbol, and Class Number.



Fig. 2.10.: Possible Symbols On a Placard.

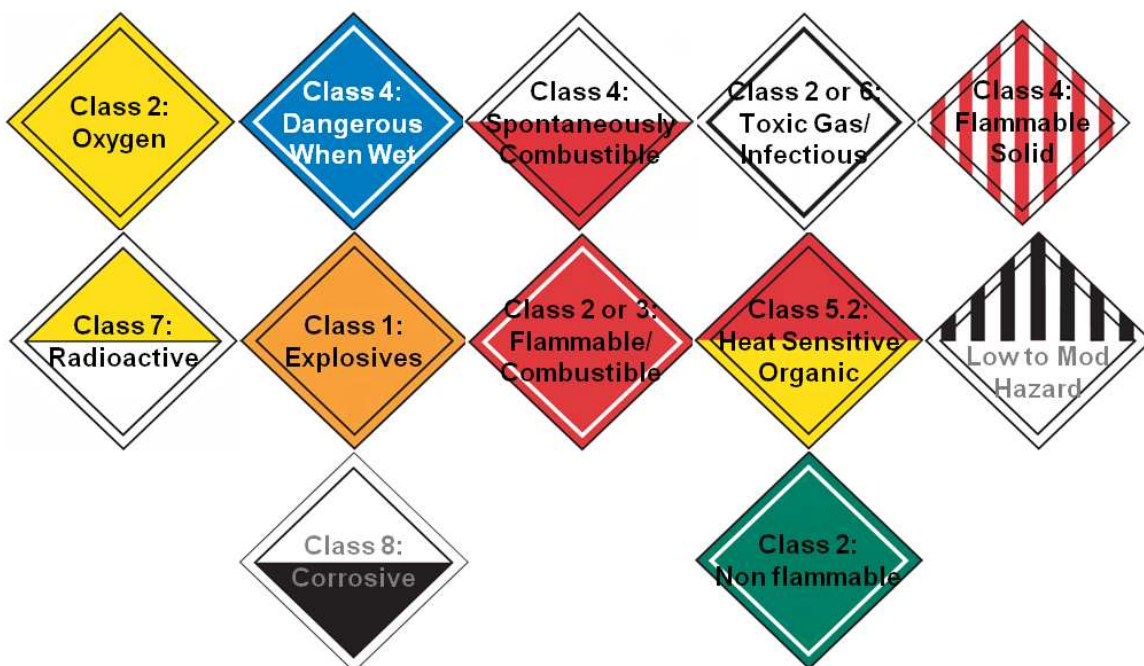


Fig. 2.11.: Possible Colors On a Placard.

## 3. GANG GRAFFITI AUTOMATIC RECOGNITION AND INTERPRETATION (GARI)

### 3.1 Review of Existing Methods

In this section we review some relevant literature in the areas of blur detection, color correction, color recognition, color image segmentation, graffiti content analysis, image feature extraction, and image retrieval and classification.

#### 3.1.1 Blur Detection

Image blur is one the most quality degrading distortions in images [36]. It may be caused by out-of-focus, relative motion between the camera and the objects, or inaccurate acquisition settings [37]. In particular, motion blur is one of the main source of blurriness in digital images [38]. Motion blur is caused by camera shake and other movements, and it can cause serious image degradation that can compromise the recognition of objects in the image. Since cheap camera modules in mobile device cameras are not robust to handshaking or low light conditions both hardware and software approaches have been proposed to overcome this problem [39].

Hardware approaches include stabilized lens [40] or Shift-CCD/CMOS used to compensate camera motion [41, 42]. However, this approaches require the use of special equipment, which makes them less suitable for general use. Software approaches can be divided into two categories: frequency domain methods (e.g., computing a transform) and spatial domain methods (e.g., analyzing edges) [43, 44].

In [45] the authors propose a method to measure the global blur using the Discrete Cosine Transform (DCT) [46] information in the image. In order to be as independent

as possible from the image content, their method looks at the distribution of null coefficients instead of the values themselves. This is based on the fact that blurred images tend to have a large number of their high frequency coefficients set to zero. The quality measure is obtained by using a weighting grid that gives more importance to the coefficients on the central diagonal of the DCT matrix, since they better characterize global (circular, non-directional) blur. This method is very sensitive to uniform background and over-illuminated images. Also, its design was aimed at detecting and quantifying only out-of-focus blur.

In [47] the authors propose a blur detection and quantification method based on edge type and sharpness analysis using the Haar-Wavelet Transform (HWT) [48]. The method takes advantage of the ability of the HWT in discriminating edge types, and can detect both out-of-focus and linear-motion blur. Edges are classified into four types: Dirac-Structure, Roof-Structure, Astep-Structure, and Gstep-Structure, the last two being derived from the Step-Structure type. A HWT with three levels of decomposition is first determined, an edge map is then constructed in each scale. After that, this edge map is partitioned, and local maxima in each window are found. If the number of Dirac and Astep structures occurrences are above a threshold, the image is considered blurred.

In [49] a no-reference blur metric based on edge length is proposed. First, a Sobel operator [50] is used to detect edge locations on the luminance component of the image. Then, the edge lengths corresponding to the distance between the starting and ending positions of the edge are computed. The global blur measure is obtained by averaging the lengths over all edges found. This method only considers Gaussian blur. In [44, 51] a low complexity blur metric based on Cumulative Probability of Blur Detection (CPBD) which utilizes probability distribution of edge widths is described.

Generally, spatial domain methods are more efficient than frequency domain methods for blur detection, as they do not require an additional transformation to another

domain.

A third category may be added to consider methods that use software approaches based on data obtained from hardware. In [52] inertial sensors (i.e., accelerometers and gyroscopes) built into the mobile device is used to detect motion trajectory of the camera during exposure and then estimate and remove blur from the resulting image. In [53] a “shake metric” technique for detecting camera shake using the mobile device built-in accelerometer to alert blind users in real-time to hold the camera more steadily is described. They do not propose any blur detection method to deal with out-of-focus or low light conditions.

### 3.1.2 Color Correction

One of the main properties of gang graffiti is its color. If the graffiti contents do not provide any useful information we can use color to filter gang cliques in the area and narrow the possibilities. When color correcting an image we alter its color intensities to match a reference color under a reference illumination [54, 55].

A common approach is to first estimate the scene illumination and then create a mapping between the estimate and the reference [56]. The concept is that both the intrinsic properties of a surface and the color of the illuminant have to be estimated, while only the product of the two (i.e. the actual image) is known. Current approaches can be divided into three categories: static methods, gamut-based methods, and learning-based methods [56, 57].

Static methods use a fixed parameter setting. In [58] using the gray-world assumption [59]: “the average reflectance in a scene under a neutral light source is achromatic” is described. Therefore, any deviation from achromaticity in the average scene color is caused by the effects of the illuminant. The color of the light source is estimated by segmenting the image and computing the average color of all segments.

In [60] a framework known as “gray edge” that uses higher order image statistics such as first and second image derivatives is presented. This method archives the same results as [58] by realizing that the gray-world methods are special instantiations of the  $L^\infty$  Minkowski norm. In [61] a fiducial marker with 12 color patches that they place in the image to estimate the illumination parameters is described. In [62] the use of a mobile device touchscreen to obtain the user input by displaying a captured image alongside a color grid of commonly occurring colors is investigated. The user specifies color pairs (i.e. patches in the scene and veridical colors on the grid), which are used to estimate the white point. The estimated white point is then used to construct a diagonal transform to determine the camera output under a desired illuminant.

Gamut-based methods are based on the assumption that in real-world images, for a given illuminant, one observes only a limited number of colors [63]. This limited set of colors that can occur under a specific illuminant is known as the canonical gamut and is determined in a training phase by observing as many surfaces, under one known light source (known as the canonical illuminant), as possible.

In [64,65] a gamut mapping method that takes as input an image taken under an unknown light source along with the precomputed canonical gamut and estimates the gamut of the unknown light source by assuming that the colors in the input image are representative for the gamut of the unknown light source is presented. In [66,67] the gamut mapping approach by adding dependence on the diagonal model is extended. Under the assumption of the diagonal model, a unique mapping exists that converts the gamut of the unknown light source to the canonical gamut. However, if the diagonal model does not fit the input data accurately, then it is possible that no feasible mapping can be found. This situation is avoided by incrementally augmenting the input gamut until a nonempty feasible set is found.



Learning-based methods estimate the illuminant using a model that is learned on training data.

In [68] a color-by-correlation method that replaces the canonical gamut with a correlation matrix is discussed. One correlation matrix is obtained for every considered illuminant and then used to obtain a probability for every considered light source. Using these probabilities a light source is selected using maximum likelihood [68] or Kullback-Leibler divergence [69]. Other methods use low-level statistics based on the Bayesian formulation [70, 71] and conditional random fields [72]. They model the variability of reflectance and light source as random variables. The illuminant is then estimated from the posterior distribution conditioned on the image intensity data.

Note that all the methods mentioned above use a single image from a regular digital camera to estimate the illuminant. There exist other methods that use additional images [73], specially designed devices [74] or video sequences [75].

### 3.1.3 Color Recognition

Gang graffiti are often sprayed in non-uniform surfaces, which makes them difficult to distinguish from the background. Since our system is deployed on a mobile telephone, we take advantage of the touchscreen capabilities of modern mobile devices to aid the recognition of color in gang graffiti images.

Since the first capacitive touchscreen was introduced in 1965 [76] multiple applications have been developed for the use of this device. Some examples include interactive surfaces such as sensitive walls [77], cooperative sharing and exchange of media [78], and freehand manipulation [79]. Most modern mobile devices use touchscreens with tactile feedback to interact with the user. This is used to control the device behavior with gestures [80]. The most common application is the virtual keyboard, which is known to be able to improve the performance of text entry with

respect to physical keyboards [81]. The touchscreen can be used to detect a path drawn with the finger on the screen for image analysis such as color recognition. This technique has been previously used to aid the acquisition of morphometric data from pulmonary tissues [82].

Color recognition techniques using tactile feedback use thresholds based on perceptual attributes of specific color spaces [83]. The perceptual thresholds (also known as discrimination thresholds) have been widely studied for human observers [84]. However, some methods do use thresholds based on human perceptibility, but use application-based thresholds. For example, some skin detection methods use an adaptive skin color filter to detect color regions, by setting thresholds in both RGB and HSV color spaces [85, 86].

#### **3.1.4 Color Image Segmentation**

In order to interpret the contents of a gang graffiti, we first need to segment the gang graffiti components from the background. By graffiti components we mean the objects and shapes contained in a graffiti image, such as stars, pitchforks, crowns, and arrows. Gang graffiti components are sprayed in different colors to catch the attention of rival gangs. Therefore, we can use color image segmentation techniques to identify the graffiti components for future analysis.

Since the advent of color imaging most of the image segmentation techniques were proposed for gray-level images [87–90] due to the fact that working with the color channels substantially increases the computational complexity of the method [91]. There has been a remarkable growth on color image segmentation approaches [92–96], which can be divided into three categories [97]: physics based, feature-space based, and image-domain based.

Methods based on physics include dichromatic reflection models [98] and unichromatic reflection models [99] for single illumination sources, and a more general model of image formation [100] for multiple illuminations.

In [98] a method that does not require explicit color segmentation. They separate diffuse and specular reflection components by comparing the intensity logarithmic differentiation of specular-free images and input images iteratively is described. The specular-free image is a set of diffuse components that can be generated by shifting a pixel's intensity and chromaticity nonlinearly while retaining its hue.

Methods based on feature spaces can be sub-categorized into three groups: clustering of regions given patterns with specific properties, including methods such as  $k$ -means clustering [101] or Iterative Self-Organizing Data Analysis Technique (ISODATA) [102]; adaptive  $k$ -means clustering, including methods based on maximum a posteriori (MAP) estimation [103] or split-and-merge strategies [104]; and histogram thresholding, including methods based on RGB thresholding and hue information [105], specific skin color domains [106], or entropy thresholding [107].

Methods based on the image-domain can be subcategorized into four groups: split-and-merge, including methods such as region smoothing by Markov Random Fields (MRF) [108] or splitting by either watershed transform [109] or quad-tree image representation for segmentation of skin cancers [110], among others; region growing, including methods such as RGB color distribution growing, HSV morphological open-close growing, or color quantization growing [111]; classification based, including methods such as minimization of Hopfield networks [112], or background extraction using two three-layered neural network [113]; edge based techniques, including methods such as combination of HSI gradients [114], active contours, or the Mumford-Shah variation model [115].

In [116, 117] a color histogram for each color channel in the RGB color space is used to detect the most frequently occurring color and segment the background in

food images. Snakes, or active contours, are then used to locate object boundaries and segment images by iteratively minimizing the segmentation energy [118].

In a separate category we can include methods that use external help for segmentation, such as tactile feedback from touchscreens on mobile devices. For example, in [119] a method to extract and segment text from subway signboard images via touchscreen tracing is presented. The text location is guided by the user selecting the region of interest, and the color information is then used to segment the connected components and use Optical Character Recognition (OCR).

### 3.1.5 Graffiti Content Analysis

Once the graffiti is segmented from the background we need to analyze its contents. This is done in multiple steps, including image enhancement and reconstruction, straight line removal, and connected component reconnection.

Image enhancement and reconstruction methods can be divided in three categories: spatial filters, neural networks, and fuzzy filters [120].

Spatial filters methods operate directly on the image pixels. In [121] an overview of super-resolution (SR) image reconstruction methods used to increase spatial resolution to overcome the limitations of the sensors and optics is presented. This includes nonuniform interpolation [122, 123], projection onto convex sets (POCS) [124, 125], adaptive filtering [126–128], motionless SR reconstruction [129–131], and blind SR reconstruction [132–134].

Neural network methods try to overcome two main disadvantages of spatial filters: 1) they treat all the pixels in the same way and 2) they operate in single pixels, thus not accounting for characteristics of the neighborhood. In [135] a human visual system (HVS)-directed neural-network-based adaptive interpolation scheme for natural image that produces a higher visual quality for the interpolated image is described. The

pixels pixels of the input image are classified into human perception nonsensitive class and sensitive class, and a neural network interpolates the sensitive regions along edge directions. High-resolution digital images along with supervised learning techniques are used to automatically train the proposed neural network. A supervised method for blood vessel detection and enhancement in digital retinal images is presented in [136]. Vessel enhancement is useful for further extraction of moment invariants-based features. A neural network scheme is used for pixel classification, and a 7-D vector composed of gray-level and moment invariants-based features is used for pixel representation.

Fuzzy filters are less sensitive to local variations and are used when images are corrupted with additive noise [137]. In [138] a method to reduce impulse noise known as a “Fuzzy Impulse Noise Detection and Reduction Method (FIDRM)” is described. Based on the concept of fuzzy gradient values, the detection method constructs a fuzzy set impulse noise represented by a membership function that is used by the filtering method, which is a fuzzy averaging of neighboring pixels. The fuzzy set is then used to filter the input image in an iterative fashion. However, FIDRM does not outperform the Median based filters for random impulse noise. In [139] a fingerprint image enhancement method by using fuzzy-based filtering technique and adaptive thresholding is investigated. A process called de-fuzzification, used to produce a quantifiable result in fuzzy logic given fuzzy sets and corresponding membership degrees, is used to improve the contrast of the noisy image.

Straight line removal is the process of deleting lines or segments that do not belong to relevant contents of an image. Since lines have a very similar pattern to character strokes in graffiti images they cannot be eliminated during their initial character extraction stages [17]. Therefore a Hough Transform (HT) is used to detect straight lines in binary images after segmentation and then delete all pixels connected along the lines. After that one reconnects the components originally belonging to graffiti components that intersected with the lines. In [140–142] scratch line detection, re-

removal and restoration on aged films is described. The methods are based on Canny operators, but pixel patches are also used for inpainting [143]. The scratch line detection is based on two general strategies: subdivision of video bands and progressive detection/inpainting. In [144] a method based on energy density and a shear transformation to separate lines from background presented. The shear transform overcomes the disadvantage that linear information loss would happen if the separation method is used only in one direction. Then templates in the horizontal and vertical directions are built to separate lines from background given the fact that the energy concentration of the lines usually reaches a higher level than that of the background in the negative image.

Connected component reconnection is used to merge components that belong to the same object but have been detached during the segmentation or the line removal steps. Contour reconnection methods are widely used in topographic map reconstruction [145]. In [146, 147] the authors propose a method to fill the gaps in contour lines by introducing properties based on geometrical and topological information such as parabolic and opposite directions and differences of y-ordinate of end points. In [148, 149] a method for restoration of degraded digits is presented. The proposed method uses a circular path detection and character stroke analysis based on inertial and centripetal forces. The method then artificially re-creates the stroke segments in order to reconstruct the digit.

### **3.1.6 Image Features**

In order to retrieve similar graffiti images from our database and classify the automatically segmented graffiti components we need to find features that represent images as uniquely as possible. There are four major types of features we can use: color features, texture features, shape features, and local features [150–156]. Given the nature of gang graffiti if we are only interested in describing the segmented graf-

fiti components features like graffiti color or surface texture will not provide useful information. In that case only shape features will be necessary. However, if we want to use information not only from the graffiti components but also from the graffiti background (for image matching and retrieval) we can use color and texture features.

Table 3.1 summarizes some of the state-of-the-art feature types.

Table 3.1: Image feature types and sizes.

Feature	Type	Dimension	Notes
GCH	color	$N_C$	$N_c$ : Num. colors in quantized space
CCV	color	$2 \times N_C$	
CM	color	$2 \times N_{M_O}$	$N_{M_O}$ : Num. moments
CW-HSV	color	63 bits	
TBD	texture	12 bits	
HTD	texture	$2 \times N_S \times N_K$	$N_S$ : Num. scales, $N_K$ : Num. orientations
EHD	texture	$2 \times N_S \times N_{B_Q}$	$N_{B_Q}$ : Num. borders quantization
Gabor	texture	$2 \times N_S$ (or $2 \times N_K$ )	
FD	shape	$N_{FD}$	$N_{FD}$ : Num. Fourier Descriptors
CSSD	shape	$N_P$ bytes	$N_P$ : Num. peaks on contour map
GMD	shape	$N_{M_O}$	
ZMD	shape	$N_{M_O}$	
SIFT	local	128	
SURF	local	64	
PHOW	local	128	
SC	local	$N_\theta \times N_r$	$N_\theta$ : Num. angles, $N_r$ : Num. of radius

Color features are the most used visual feature in Content-Based Image Retrieval (CBIR) systems and the most explored features in the literature [157,158]. The main reason is because humans tend to differentiate images mostly by means of color features. The Global Color Histogram (GCH) [159] analyzes the entire color information of the image. Usually, a quantization step is required to reduce the number of distinct colors.

The Color Coherence Vector (CCV) descriptor [160] classifies each pixel in either coherent or incoherent, based on whether or not it is part of a large similarly-colored region. The CCV first blurs the image and the color space is discretized to eliminate

small variations between neighbor pixels. Next, the connected components of the image are found in order to classify the pixels in coherent or incoherent.

The Chromaticity Moment (CM) descriptor [161] characterizes images by chromaticity in the CIE XYZ color space. A chromaticity histogram and a chromaticity trace is generated. The trace indicates the presence of a value  $(x, y)$  in the image. The trace and histogram are used to define the chromaticity moments. The reasons for us to choose CM are its compact feature vector generation and its fast distance function, which estimates the modular difference between corresponding moments.

The Color Wavelet HSV (CW-HSV) descriptor [162] computes color features in the wavelet domain [163]. First the image global color histogram in HSV color space is found. Then the Haar transform coefficients of the histogram are determined hierarchically by using Haar wavelet functions. In the end, 63 binary values compose the feature vector. The distance between two feature vectors is calculated by the Hamming distance. The reasons for us to choose CW-HSV are its compact feature vector generation (only 63 bits) and its fast distance function.

Texture features, like color features, create powerful low-level descriptors for image search and retrieval applications [164].

The Texture Browsing Descriptor (TBD) [165] relates to the perceptual characterization of texture, in terms of regularity, directionality and coarseness. The coarseness is related to image scale or resolution. This descriptor is useful for browsing type applications and coarse classification of textures. The Homogeneous Texture Descriptor (HTD) [164] provides a quantitative characterization of homogeneous texture regions for similarity retrieval. It is determined by first filtering the image with a bank of orientation and scale sensitive filters, and computing the mean and standard deviation of the filtered outputs in the frequency domain.

The local Edge Histogram Descriptor (EHD) [164] is useful when the underlying region is not homogeneous in texture properties. It is computed by first sub-dividing the image and computing local edge histograms. Edges are broadly grouped into five



categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic. Thus, each local histogram has five bins, and with the image partitioned into 16 sub-images results in 80 bins. The Gabor-based descriptor [166] is computed by passing the image through a bank of Gabor filters [167]. Filters in a Gabor filter bank can be considered as edge detectors with tunable orientation and scale so that information on texture can be derived from statistics of the outputs of those filters. The descriptor is then formed as a vector of means and standard deviations of filter responses.

Shape features are one of the primary low level image features exploited in content-based image retrieval [168]. They can represent images by their contours or regions.

The Fourier Descriptor (FD) [169–171] is a spectral descriptor obtained from a Fourier transform on a shape signature. The shape signature is a one-dimensional function, which is derived from shape boundary coordinates. The set of normalized Fourier transformed coefficients is known as the Fourier descriptor of the shape. The Curvature Scale Space Descriptor (CSSD) [172, 173] treats shape boundary as a 1D signal, and analyzes this 1D signal in scale space. By examining zero crossings of curvature at different scales, the concavities/convexities of shape contour are found. These concavities/convexities are useful for shape description because they represent the perceptual features of shape contour.

The Geometric Moment Descriptor (GMD) [174, 175] is based on moment invariants for shape representation and similarity measure. Moment invariants are derived from moments of shapes, and are invariant to 2D geometric transformations of shapes. The Zernike Moment Descriptor (ZMD) [176, 177] uses orthogonal moments to recover the image from moments based on the theory of orthogonal polynomials (Zernike polynomials). It allows independent moment invariants to be constructed to an arbitrarily high order.

Local features rely on the concept that objects in images consist of parts that can be modeled with varying degrees of independence [178, 179]. They are used in many applications, such as object detection, symbol spotting, or image registration.

The Scale Invariant Feature Transform (SIFT) descriptor [15] combines a scale invariant region detector and a descriptor based on the gradient distribution in the detected regions. The descriptor is represented by a 3 dimensional histogram of gradient locations and orientations. The contribution to the location and orientation bins is weighted by the gradient magnitude. The quantization of gradient locations and orientations makes the descriptor robust to small geometric distortions and small errors in the region detection. The Speed Up Robust Feature (SURF) descriptor [24] is based on similar properties as SIFT, but relies on integral images for image convolutions. First, it fixes a reproducible orientation based on information from a circular region around the interest point. Then, it constructs a square region aligned to the selected orientation, and extract the SURF descriptor from it.

The Pyramid Histogram Of visual Words (PHOW) descriptor [180, 181] is computed using SIFT on a dense grid at a fixed scale, which can be directly clustered using  $k$ -means [182] to form a “bag of words” feature. The Shape Context (SC) descriptor [183–185] is similar to the SIFT descriptor, but is based on edges. It is a 3 dimensional histogram of edge point locations and orientations. The edge locations are quantized into a log-polar coordinate system and the orientations are quantized into an angular coordinate system.

### 3.1.7 Image Retrieval

Retrieval of gang graffiti images is very useful for the first responder in the field. It can provide information about related graffiti in the area based on the contents of the image. For example, a user can check if someone else has taken an image of the same gang graffiti in the past, and pull all the related information without having to

do any further image analysis.

Content-Based Image Retrieval (CBIR) can be used for finding images from large and unannotated image databases. There are four core techniques for CBIR: visual signature, similarity measures, classification and clustering, and search paradigms [186–188]. Visual signature usually involves three steps: 1) segmenting images using methods such as  $k$ -means clustering [182], normalized cuts [189], or salient region detection [190]; 2) using features such as color, texture, or shape [191]; 3) constructing the signatures (or feature vectors) using distributions [192] or adaptivity [193]. Similarity measure methods include manifold embedding [194], and vector quantization [195]. Classification and clustering methods include hierarchical  $k$ -means [196], support vector machine [197], or Bayesian classifiers [198]. Search paradigms methods include learning-based [199], probabilistic [200], region-based [201], feedback specification [202], or user-driven [203].

In [204] a method for image-based retrieval using a mobile device is presented. Features are measured after detecting salient regions and then quantified to form a vector using a clustering-based bag-of-words model and sparse matrix methods. Invert document methods are used to speed up real-time queries. In [11] a CBIR system tattoo image retrieval is proposed. The system automatically uses SIFT features and additional information (i.e., body location of tattoos and tattoo classes) to improve the retrieval time and retrieval accuracy. Geometrical constraints are also introduced in SIFT keypoint matching to reduce false retrievals.

Sketch-Based Image Retrieval (SBIR) uses a line-based hand-drawing (a “sketch”) as a query. In some scenarios outline sketches are typically easier and faster to generate than a complete color description of the scene [205, 206].

In [207] a method based on elastic matching of sketched templates over the shapes in the images to evaluate similarity ranks is described. The degree of matching achieved and the elastic deformation energy spent by the sketch to achieve such a

match are used to derive a measure of similarity between the sketch and the images in the database and to rank images to be displayed. The elastic matching is integrated with arrangements to provide scale invariance and take into account spatial relationships between objects in multi-object queries.

In [208] a technique that deals with images containing several complex objects in an inhomogeneous background is presented. Two abstract images are obtained using strong edges of the model image and the morphologically thinned outline of the sketched image. The angular-spatial distribution of pixels in the abstract images is then employed to extract new compact and effective features using the Fourier transform. The features are rotation and scale invariant and robust against translation.

The image retrieval method used in GARI fall into the feature-space category in CBIR. However, our approach differs from the methods mentioned above. Although there are some techniques in the literature that use only hue or luma information, either circular histogram thresholding [209] or one-dimensional histogram thresholding [210], we do not obtain the descriptors of the probability distribution from the color histogram of the image. Instead, the median and the variance obtained from the tracing-bases color recognition process are used for segmentation. Our segmentation approach does not produce binarized images, but grayscale images weighed by a Gaussian distribution, thus creating a probability map for a specific luma or hue. These types of probability maps are used for increased accuracy and robustness in some clustering techniques [211, 212]. Our content based image retrieval approach uses hierarchical  $k$ -means to build a vocabulary tree based on the method in [196].

### 3.2 Mobile-Based Motion Blur Prevention and Detection

In order to analyze gang graffiti we need to preserve the details in the image acquired with a mobile device. Instead of doing blur detection after taking the image we propose a mobile-based method to prevent the user from producing blurred im-

ages. To that end we use a customized camera function on the mobile that detects shake events (i.e. motion blur). When the camera function is launched through the GARI application we start a three second countdown and listen for changes from the accelerometer sensor in the mobile device. A sensor of this type measures the acceleration of the device ( $Ad$ ) in SI units ( $m/s^2$ ). Conceptually, this is done by measuring forces applied to the sensor itself ( $Fs$ ) using the relation:

$$Ad = -\frac{\sum Fs}{mass}. \quad (3.1)$$

In particular, the force of gravity is always influencing the measured acceleration:

$$Ad = -g - \frac{\sum Fs}{mass}. \quad (3.2)$$

For this reason when the device is sitting on a table the accelerometer reads a magnitude of  $g = 9.81m/s^2$ . Similarly, when the device is in free-fall its accelerometer reads a magnitude of  $0m/s^2$ . We compute the total movement  $M$  as

$$M = \Delta A_x + \Delta A_y + \Delta A_z - (A_x + A_x + A_z), \quad (3.3)$$

where  $(\Delta_x, \Delta_y, \Delta_z)$  are the acceleration force changes along the  $(x, y, z)$  axes respectively, and  $(A_x, A_y, A_z)$  are the most recent acceleration values along the  $(x, y, z)$  axes respectively. If  $\Delta A$  and  $A$  occur in a time difference of  $T_t = 400$  milliseconds and  $M$  is above a threshold  $T_M = 3m/s^2$  we report a shake event. In that case the countdown is reset to three seconds and no image is taken. If no significant change on  $M$  is perceived when the countdown reaches zero, we trigger the auto-focus and an image is acquired.

Even though we try to prevent motion blur, if there is a shake event during auto-focus or image acquisition we can obtain a blurred image. For this reason motion

blur detection is done on a reduced size version of the image of width  $W_t = 400$  pixels. A reduced size version is enough to detect excessive motion blur produced in this particular case. We use a modification of the method proposed in [44] because of its simplicity and speed. In [44] a modification to a well known method known as cumulative probability of blur detection (CPBD) is presented. This utilizes the probability distribution of edge widths [51]. The blur metric estimation starts by creating an edge binary map using a Sobel operator in the vertical direction of the grayscale image. Then, the image is divided into blocks of size  $64 \times 64$ . A block is considered an edge block if it contains a number of edge pixels greater than a fixed threshold. For each edge block the probability of blur detection  $P_{BLUR}$  at each edge pixel  $e_i$  is computed as

$$P_{BLUR}(e_i) = 1 - e^{-\left|\frac{w(e_i)}{w_{JNB}(e_i)}\right|^\beta}, \quad (3.4)$$

where  $w(e_i)$  is the edge width [49],  $w_{JNB}(e_i)$  is the “just noticeable blur” (JNB) width with value of either 5 or 3 [51], and  $\beta$  is a parameter whose value is obtained from least squares fitting. The CPBD is estimated as:

$$CPBD = P(P_{BLUR} \leq P_{JNB}) = \sum_{P_{BLUR}=0}^{P_{BLUR}=P_{JNB}} P(P_{BLUR}), \quad (3.5)$$

where  $P(P_{BLUR})$  denotes the value of the probability distribution function at a given  $P_{BLUR}$ . This metric is based on the fact that, at the JNB,  $w(e_i) = w_{JNB}(e_i)$ , which corresponds to the probability of blur detection  $P_{BLUR} = P_{JNB} = 63\%$ . Therefore, for a given edge  $e_i$ , when  $P_{BLUR} \leq P_{JNB}$  the edge is considered not to be blurred. Hence, a higher metric value represents a sharper image. The modification proposed by [44] relies on the fact that the CPBD can be expressed by the ratio

$$CPBD = P(P_{BLUR} \leq P_{JNB}) = \frac{|S_1|}{|S_e|}, \quad (3.6)$$

where  $|S_1|$  is the set of edge pixels with  $P_{BLUR} \leq P_{JNB}$  and  $|S_e|$  is the set of all edge pixels. Since

$$1 - e^{-\left|\frac{w(e_i)}{w_{JNB}(e_i)}\right|^\beta} \leq 0.63 \Rightarrow w(e_i) \leq w_{JNB}(e_i)(-\ln(0.37))^{1/\beta} \quad (3.7)$$

the CPBD becomes

$$CPBD = \frac{\sum_{w_{JNB}=\{3,5\}} \sum_{w=2}^{w_{JNB}-1} H(w_{JNB}, w)}{|S_e|}, \quad (3.8)$$

where  $H(w_{JNB}, w)$  is the number of edge pixels with JNB width  $w_{JNB}$  and edge width  $w$ . By using this approach we avoid using exponentials for gradient estimations, thus reducing the computational complexity.

We can further increase the complexity by approximating the CPBD as

$$BM = \frac{\sum_{x,y} |G_x(x,y)| + \sum_{x,y} |G_y(x,y)|}{w_I h_I}, \quad (3.9)$$

where  $(G_x, G_y)$  are the Sobel derivatives in the  $x$  and  $y$  directions respectively, and  $(w_I, h_I)$  are the dimensions of the image. That is,  $BM$  is the ratio of edge pixels over the size of the image. Note that by doing this we cannot call the metric CPBD, since it is not based on cumulative probability.

By using  $BM$  as our blur metric we can set a threshold  $T_{BM}$  so that if  $BM < T_{BM}$  we consider the image to be blurred, and we ask the user to retake the image by resetting the countdown back to three. Our experiments with more than 1,000 images

from our dataset and different mobile devices showed that  $T_{BM} = 0.1$  produces the best results.

Note that since the proposed blur detection metric is solely based on the number of edge pixels, the method will also reject images with large uniform patches or images taken under low light conditions. In fact this properties are not a drawback, but rather desired in the context of gang graffiti recognition and interpretation. Also note that all the processing is done on the mobile device.

Figure 3.1 shows an example of the blur metric results.

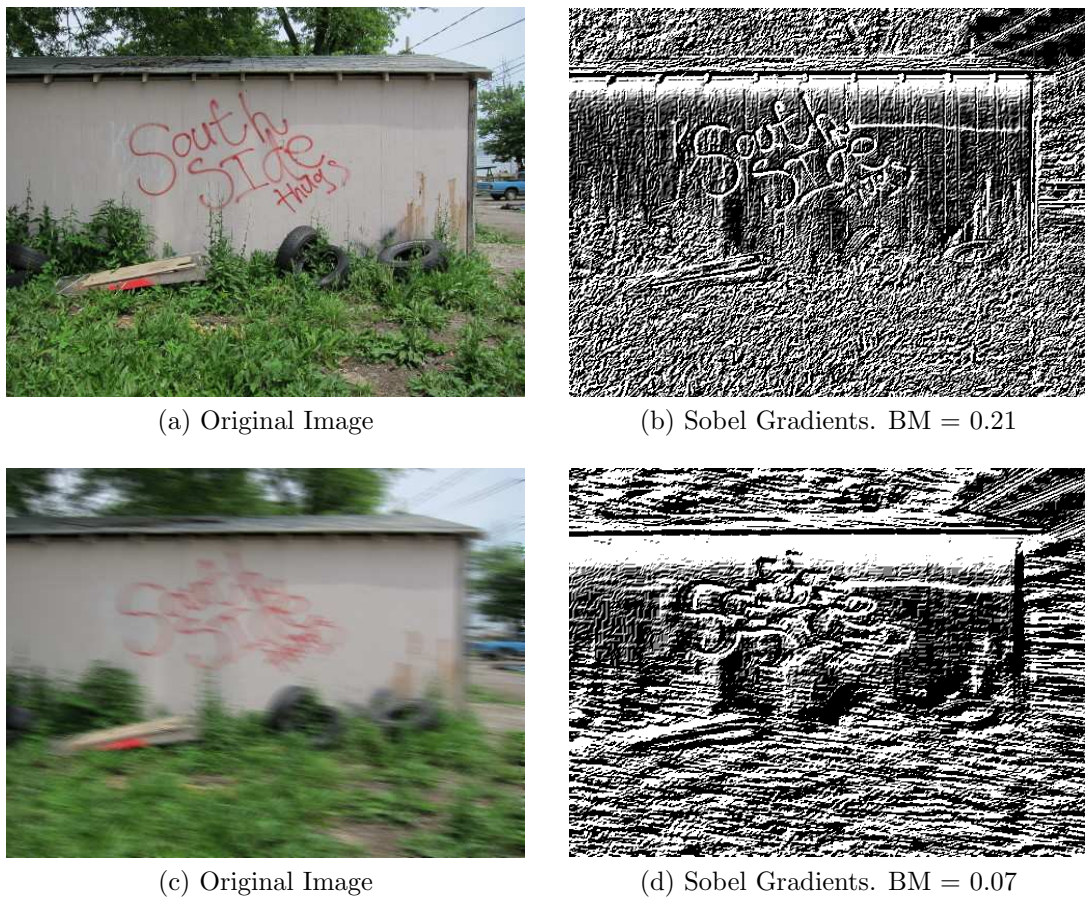


Fig. 3.1.: Example of Blur Metric Results.

Table 3.2 shows all the parameters/thresholds we used including empirically derived parameters.



Table 3.2: Parameters and thresholds used in Mobile-Based Motion Blur Prevention.

<b>Parameter</b>	<b>Description</b>	<b>Value</b>
$T_t$	Time between acceleration changes	400 ms
$T_M$	Threshold to consider shake event	$3m/s^2$
$W_t$	Width of resized image for blur detection	400 px
$T_{BM}$	Threshold for Blur Metric (BM) ratio	0.1

### 3.3 Color Correction Based on Mobile Light Sensor<sup>1</sup>

First responders are out in the field when using the mobile application to take images of gang graffiti. Since gang graffiti are usually found in dangerous neighborhoods we want to minimize the use of intrusive methods to do color correction. The use of fiducial markers may be suspicious to gang members in the surroundings. The use of face detection for white balancing [213] make first responders concerned about their privacy.

One way to do color correction is to first obtain information about the scene illumination. This can be done by using the light sensor on the mobile device. For example, the light sensor in an Android smartphone returns the ambient light level in SI lux units (lumens per square meter). Unlike human perception of light, lux readings are directly proportional to the energy per square meter that is absorbed per second [214,215]. However, human perception can be simplified by creating several ranges of interest with known upper and lower thresholds. Table 3.3 shows an example of several thresholds for common lighting conditions and the corresponding lighting steps obtained from the light sensor on a Samsung Galaxy Nexus smartphone. Each lighting step represents a change in lighting environment. Figure 3.2 illustrates the relationship between the lighting step and the lux values. Figure 3.3 illustrates the same relationship when using a logarithmic scale on the lux values to see how the relationship becomes linear.

Once we obtain a lux  $LX$  from the mobile device we want to associate a color correction matrix to it. A color correction matrix is a mapping between an image illuminated with reference lighting and an image acquired with unknown lighting condition.

The idea is to generate color correction matrices from ground-truth data to populate a database. The database acts as a look up table where a lux value maps to a color correction matrix. Later, when first responders use the application in the field

---

<sup>1</sup>The work presented in this section is partly based on the work by my Purdue colleague Dr. Chang Xu [44].

Table 3.3: Thresholds for common lighting conditions and corresponding lighting steps.

Condition	Lux (start)	Lux (end)	Lighting step
Pitch Black	0	10	1
Very Dark	11	50	2
Dark Indoors	51	200	3
Dim Indoors	201	400	4
Normal Indoors	401	1000	5
Bright Indoors	1001	5000	6
Dim Outdoors	5001	10,000	7
Cloudy Outdoors	10,001	30,000	8
Direct Sunlight	30,001	100,000	9

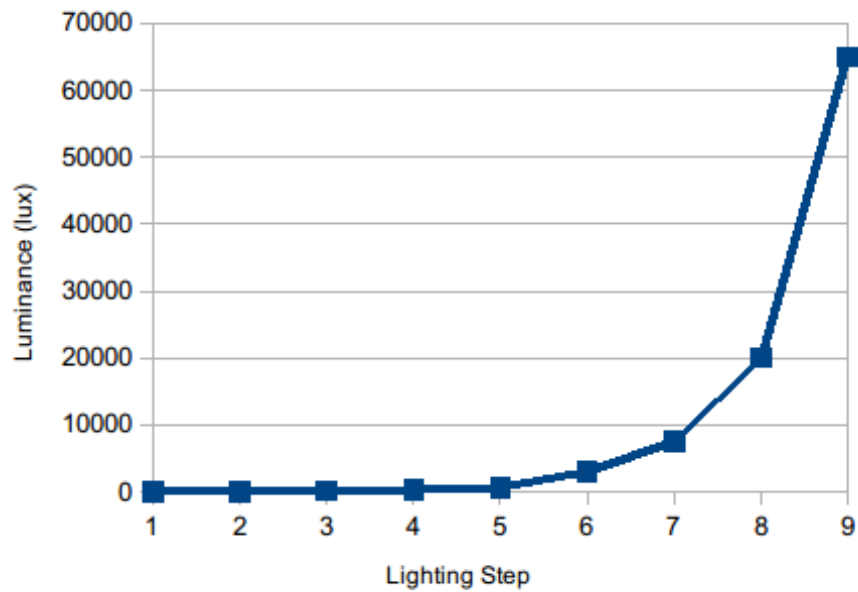


Fig. 3.2.: Lighting Step vs. Luminance (lux).

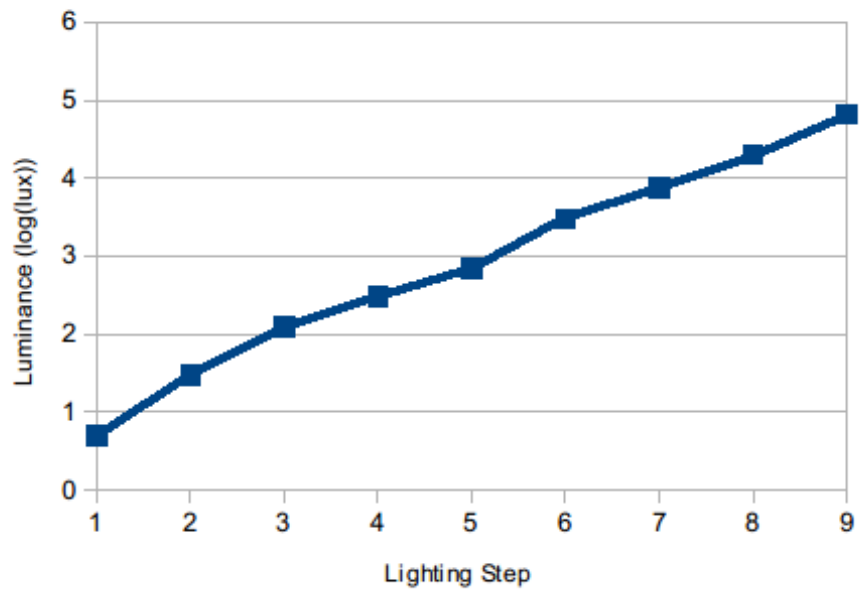


Fig. 3.3.: Lighting Step vs. Luminance (log(lux)).

we will only need a lux value to retrieve the corresponding color correction matrix and use it to correct the acquired image.

Figure 3.4 illustrates the process to populate the database with color correction matrices and lux values. Note that the computation of the color correction matrix  $M_{GT \rightarrow D65}$  is done on the mobile device. A ground-truth image is an image acquired with a mobile device under a specific scene illumination. Figure 3.5 shows an example of a ground-truth image with a lux value of 5,116. The image contains a checkerboard-like design known as a “fiducial marker” used as a reference of known dimensions and color patches [44, 61, 216].

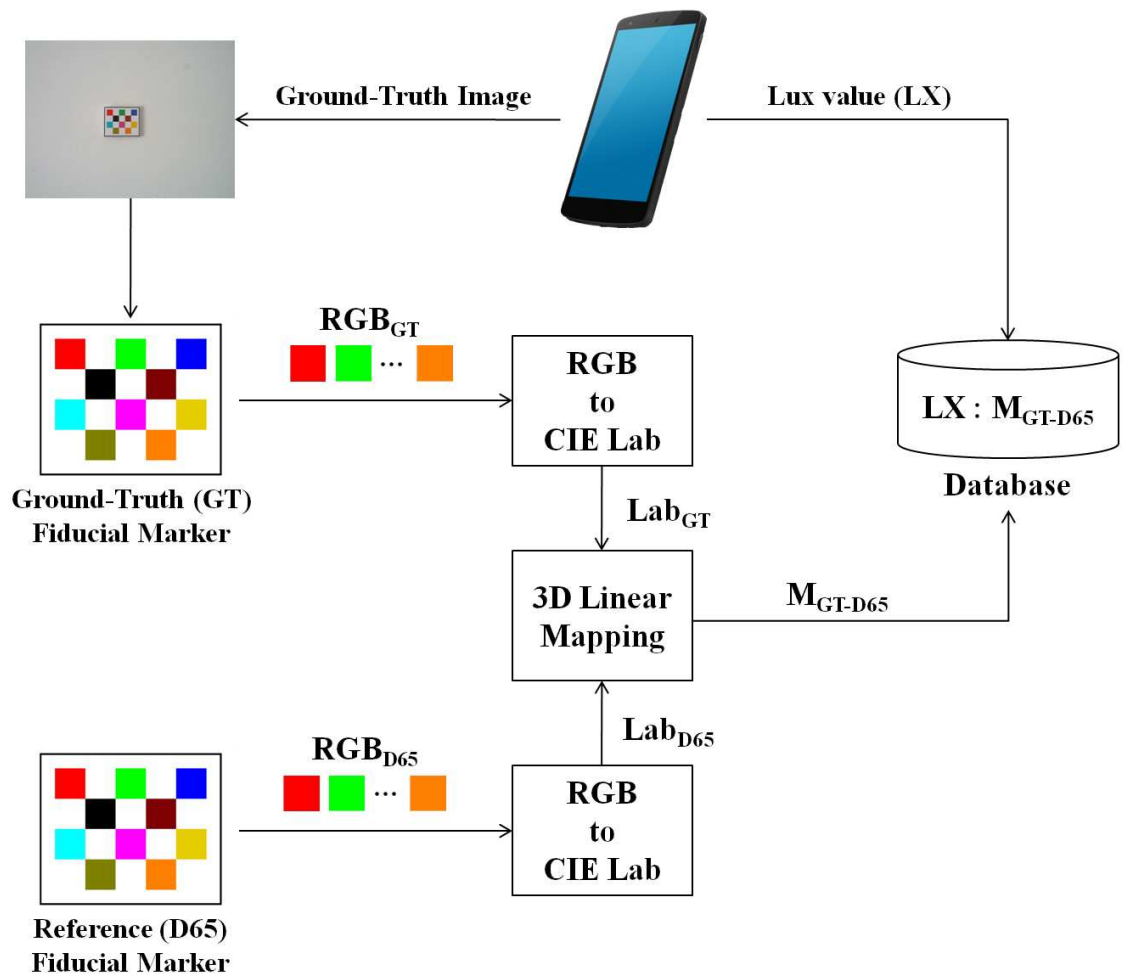


Fig. 3.4.: Color Correction Based on Mobile Light Sensor.

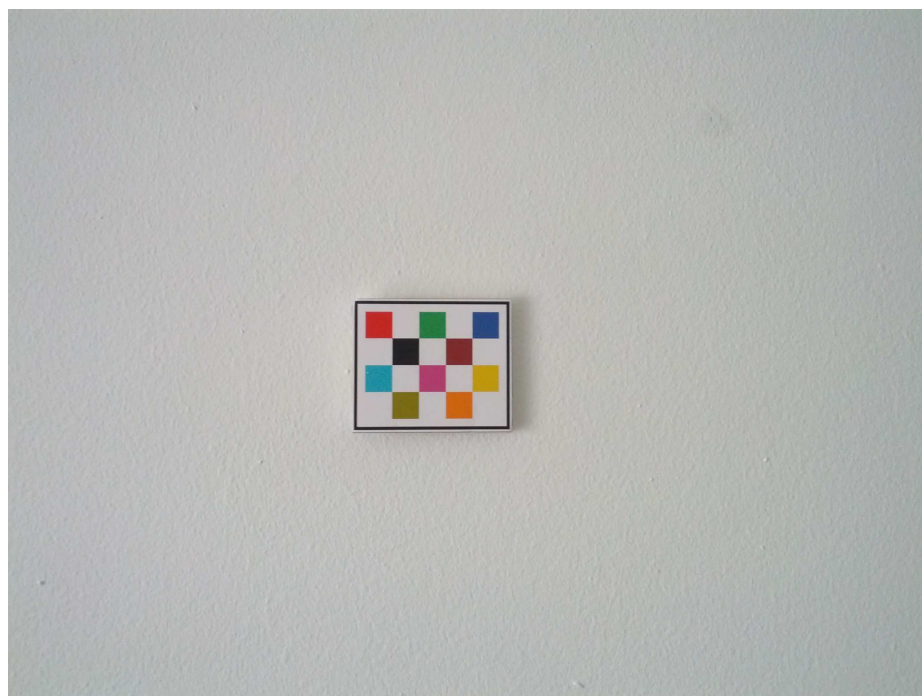


Fig. 3.5.: Example of ground-truth image with a lux value of 5,116.

We start by detecting the corners of the fiducial marker in the ground-truth image using the method described in [217]. The image is first converted to grayscale and binarized according to

$$I_{out}(x, y) = \begin{cases} 255 & \text{if } I(x, y) > T(x, y) \\ 0 & \text{else} \end{cases}, \quad (3.10)$$

where  $T(x, y)$  is a threshold calculated individually for each pixel using a Gaussian kernel. The kernel is a matrix of Gaussian filter coefficients:

$$G_i = \alpha \exp^{-\frac{i - \left(\frac{k-1}{2}\right)^2}{(2\sigma)^2}}, \quad (3.11)$$

where  $k$  is the aperture size (odd and positive),  $\sigma$  is the Gaussian standard deviation computed as  $\sigma = 0.3((k-1)^{1/2} - 1) + 0.8$ ,  $i = 0, \dots, k-1$  and  $\alpha$  is the scale factor chosen so that  $\sum_i G_i = 1$ . The binary image is eroded to separate the checkerboard at the corners and obtain a set of quadrangles. Finally, a quadrangle linking step checks the position of the fiducial marker patches to confirm the board pattern.

Once we have detected the location of the checkerboard corners we estimate the location of each of the 11 color patches and extract their mean RGB value. These color patches are used to generate a 3D linear mapping between the scene illumination (ground-truth image) and the reference fiducial marker colors [218]. We used the linear model in LAB color space from [44] for color correction, as it produced the best results in our experiments (Section 5.1.2). We convert each of the RGB color patches to CIE Lab using the standard RGB to CIE Lab transformation [219, 220] as follows:

RGB to XYZ:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.12)$$

XYZ to CIE Lab:

$$L = 116(Y/Y_n)^{1/3} - 16 \quad (3.13)$$

$$a = 500((X/X_n)^{1/3} - (Y/Y_n)^{1/3}) \quad (3.14)$$

$$b = 200((Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}) \quad (3.15)$$

$X_n$ ,  $Y_n$  and  $Z_n$  are the values of  $X$ ,  $Y$  and  $Z$  for the illuminant (reference white point). The  $L$  coordinate in CIE Lab is correlated to perceived lightness. The  $a$  and  $b$  coordinates are the red-green and yellow-blue of the color-opponent respectively. We followed the ITU-R Recommendation BT.709 and used illuminant  $D_{65}$ , where  $[X_n, Y_n, Z_n] = [0.950456, 1, 1.088754]$  [221]. To obtain the optimal 3-dimensional linear transformation  $M_{GT \rightarrow D65}$ , a  $3 \times 3$  matrix that converts the Lab color patches from the ground-truth to the Lab color patches from the  $D_{65}$  reference, we need to solve

$$M_{GT \rightarrow D65} = \operatorname{argmin}_{M_{3 \times 3}} \sum_{i=1}^{11} \left\| (Lab_i)_{D65}^T - M_{3 \times 3} (Lab_i)_{GT}^T \right\| \quad (3.16)$$

by linear regression by using ordinary least-squares estimates of the regression coefficients [222]. We follow this procedure for each ground-truth image to populate the database with mappings between lux values  $LX$  and color correction matrices  $M_{GT \rightarrow D65}$ .



Every time a user acquires an image  $I_q$  using the mobile device we sent it to the server along with the lux value  $LX_q$ . Then, we use the  $M_{GT \rightarrow D65}$  associated to the closest  $LX$  in the database to correct  $I_q$ .

Figures 3.6 and 3.7 show example outputs of our proposed color correction method. Details about the number of ground-truth images used and the efficiency of the method are described in Section 5.1.2.

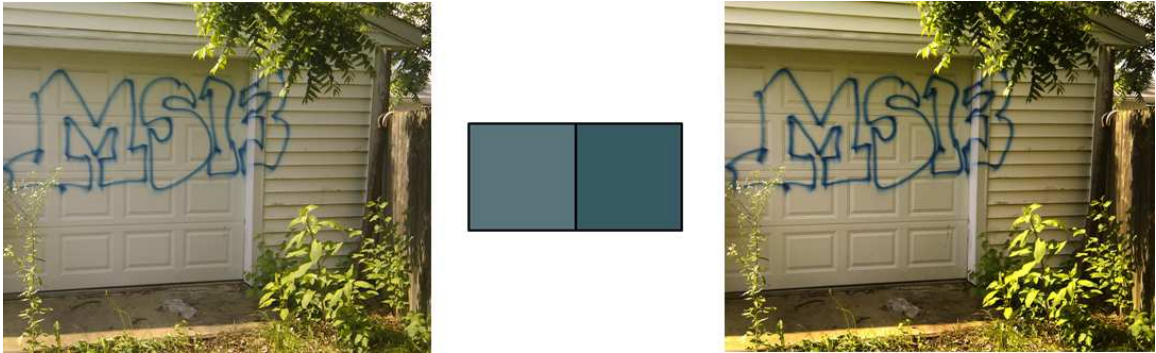


Fig. 3.6.: Example of color correction when  $LX = 35,611$ . Left: before correction; right: after correction.

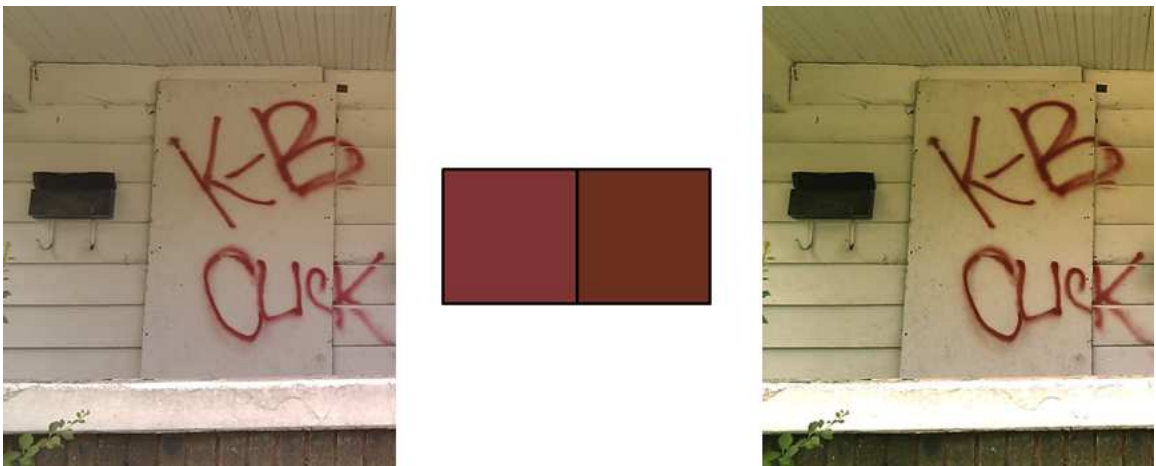


Fig. 3.7.: Example of color correction when  $LX = 41,980$ . Left: before correction; right: after correction.

### 3.4 Color Recognition Based on Touchscreen Tracing

In this method the user acquires an image of a gang graffiti and traces a path along a colored region using the touchscreen display. Then we recognize the color along the path and provide a list of gangs related to the color by querying an internal database on the mobile phone. For this method we use an RGB to Y'CH color space conversion. Figure 3.8 shows an overview of our color recognition method. Again note that this technique is done on the hand-held device.

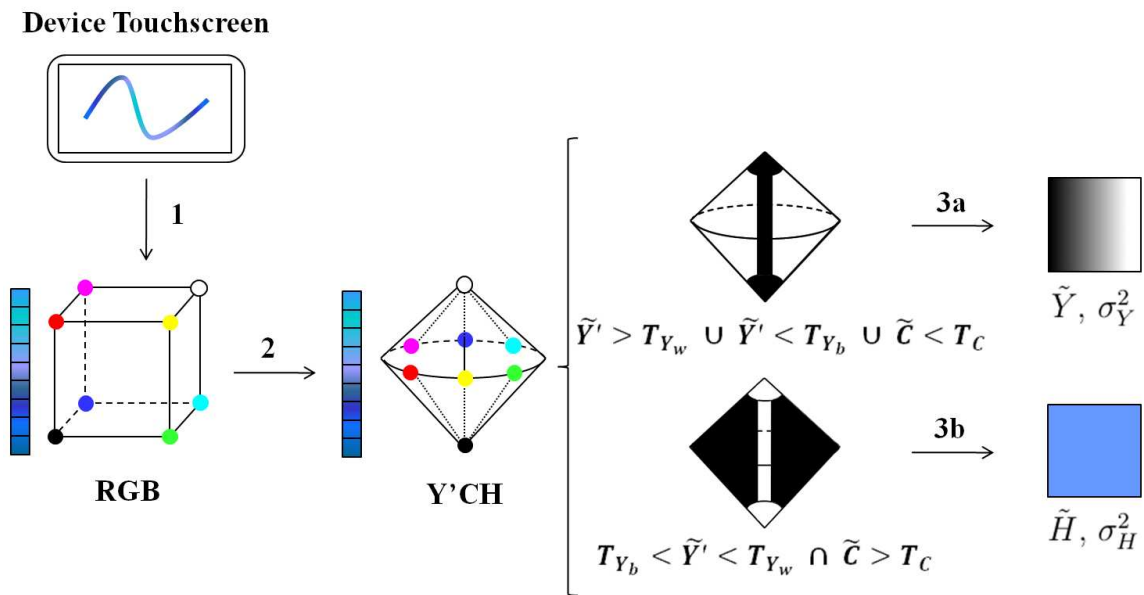


Fig. 3.8.: Color Recognition Based on Touch Screen Tracing.

First, the user captures an image or browses the internal gallery for an image on the device and draws a path with the finger on the touchscreen. The path is drawn along a graffiti component on the image assumed to be sprayed in uniform color. The RGB color channels of each pixel on the path are converted to a new luma/chroma/hue color space that we call the Y'CH color space. The Y'CH color space is used because color changes are more intuitive and perceptually relevant to represent in luma or hue than in RGB triplets, in order to obtain the median and the variance of the color along the traced path. Equation 3.17 shows the mapping between RGB and

Y'CH. Note that we use luma ( $Y'$ ) as opposed to luminance ( $Y$ ) [223]. Appendix A describes in detail the RGB to Y'CH color space conversion using both an arithmetic approach and a trigonometric approach. We compute three medians on the pixel array that forms the path, namely the luma median ( $\tilde{Y}$ ), the chroma median ( $\tilde{C}$ ) and the hue median ( $\tilde{H}$ ). We then define two disjoint regions in our Y'CH color space (luma region and hue region, labeled 3a and 3b in Figure 3.8 respectively), delimited by manually set thresholds based on luma ( $T_{Y_w} = 0.12$ ,  $T_{Y_b} = 0.85$ ) and chroma ( $T_C = 0.06$ ). These thresholds were empirically obtained from our database of gang graffiti, consisting of more than 700 gang graffiti images. Depending on the region where the medians are located, we do color recognition based on luma (3a) or hue (3b).

$$\begin{aligned}
 Y' &= 0.299R + 0.587G + 0.114B. \\
 C &= \max(R, G, B) - \min(R, G, B) = M - m \\
 H &= \begin{cases} \frac{G-B}{C} & \text{if } M = R \\ \frac{B-R}{C} + 2 & \text{if } M = G \\ \frac{R-G}{C} + 4 & \text{if } M = B \\ 0 & \text{if } C = 0 \end{cases} \quad (3.17)
 \end{aligned}$$

Once we have the median, either based on luma or hue, we need to decide which color is associated with it. From all the images in our database, the possible colors used on graffiti are black, white, red, blue, green, gold and purple. If the median is based on luma, the color detected is either black ( $\tilde{Y} \leq T_{\tilde{Y}}$ ) or white ( $\tilde{Y} > T_{\tilde{Y}}$ ), where  $T_{\tilde{Y}} = 0.5$ . If the median is based on hue, the color detected is  $H_d = \min_i(\theta(\tilde{H}, H_{A_i}))$ , where  $\theta(\tilde{H}, H_{A_i})$  is the angular distance between the computed hue ( $\tilde{H}$ ) and the  $i$ -th component of a set of average hues ( $H_A$ ), empirically obtained from analyzing 100 color calibrated images taken from our database. These colors are specified in Table 3.4. Figure 3.9 illustrates the separation between them in a hue slice of the Y'CH

color space. Once the color is detected, we provide a list of gangs related to that color by querying our database of gang graffiti from the mobile phone.

Finally, we also estimate the variance  $\sigma_{\tilde{X}}^2$  near the median  $\tilde{X} = \{\tilde{Y} \text{ or } \tilde{H}\}$ . This variance is used as an input to the color image segmentation method described next. Note that this method can be used with multi-colored graffiti by using it on each trace on the touchscreen.

Table 3.4 shows all the parameters/thresholds we used including empirically derived parameters.

Table 3.4: Parameters and thresholds used in Color Recognition Based on Touchscreen Tracing.

<b>Parameter</b>	<b>Description</b>	<b>Value</b>
$T_{Y_w}$	Low luma threshold	0.12
$T_{Y_b}$	High luma threshold	0.85
$T_C$	Low chroma threshold	0.05
$T_{\tilde{Y}}$	Luma threshold for black/white	0.5
$H_A^{red}$	Average hue (red)	6.10 rad
$H_A^{blue}$	Average hue (blue)	4.00 rad
$H_A^{green}$	Average hue (green)	2.20 rad
$H_A^{gold}$	Average hue (gold)	0.69 rad
$H_A^{purple}$	Average hue (purple)	5.15 rad

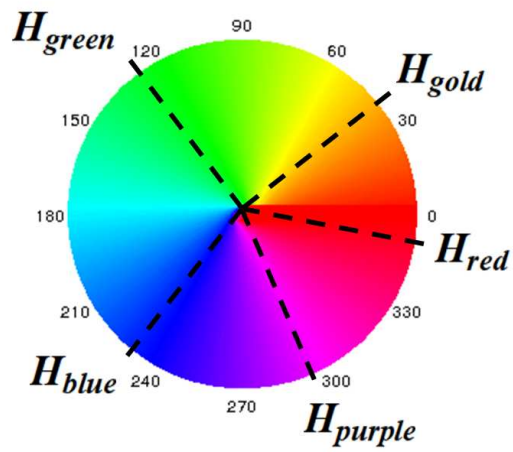


Fig. 3.9.: Separation Between Hue Averages.

### 3.5 Automatic Graffiti Component Segmentation

In this section we propose methods for automatic segmentation of graffiti components. We assume that the graffiti takes at least 50% of the image. With this assumption we resize all input images to  $W_X = 500$  pixels in width to reduce the computational complexity while maintaining the performance.

#### 3.5.1 Color Image Segmentation Based on Gaussian Thresholding

For the segmentation we use a Gaussian threshold near a specific luma or hue value in the Y'CH color space, in order to produce a segmented image where each pixel is given a weight depending on its distance from a median. Figure 3.10 shows an overview of our color segmentation method divided in 5 steps. Note that we currently use this method on the server in our system and do not use it on the hand-held device.

We assume that, given a graffiti image  $X$ , we have the median  $\tilde{X}$  and the variance,  $\sigma_{\tilde{X}}^2$ , of a traced path (step 1b). We then transform the entire RGB image to the our Y'CH color space (steps 1a and 2). Finally, we segment the image using Gaussian thresholding (steps 3 to 5). The segmentation works as follows. We first ignore all pixels in the image  $X$  that fall outside the region established during touchscreen tracing (luma or hue), using the same thresholds used for the color recognition process. This creates the thresholded grayscale image  $X_t$  (step 3). We weight the rest of the pixels using a normal distribution centered at  $\tilde{X}$  and a confidence interval of  $2\sigma_{\tilde{X}}$  (step 4), as shown in Equation 3.18, to obtain  $X_g$ . The output  $X_g$  is a grayscale image where each pixel is given a probability based on a normal distribution (step 5). This probability is higher as the pixel value gets closer to  $\tilde{X}$ . The image is then scaled to  $[0, 255]$ .

$$X_g(i, j) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma_{\tilde{X}}^2}} e^{-\frac{(X_t(i, j) - \tilde{X})^2}{2\sigma_{\tilde{X}}^2}} & |X_t(i, j)| < 2\sigma_{\tilde{X}} \\ 0 & \text{else} \end{cases} \quad (3.18)$$

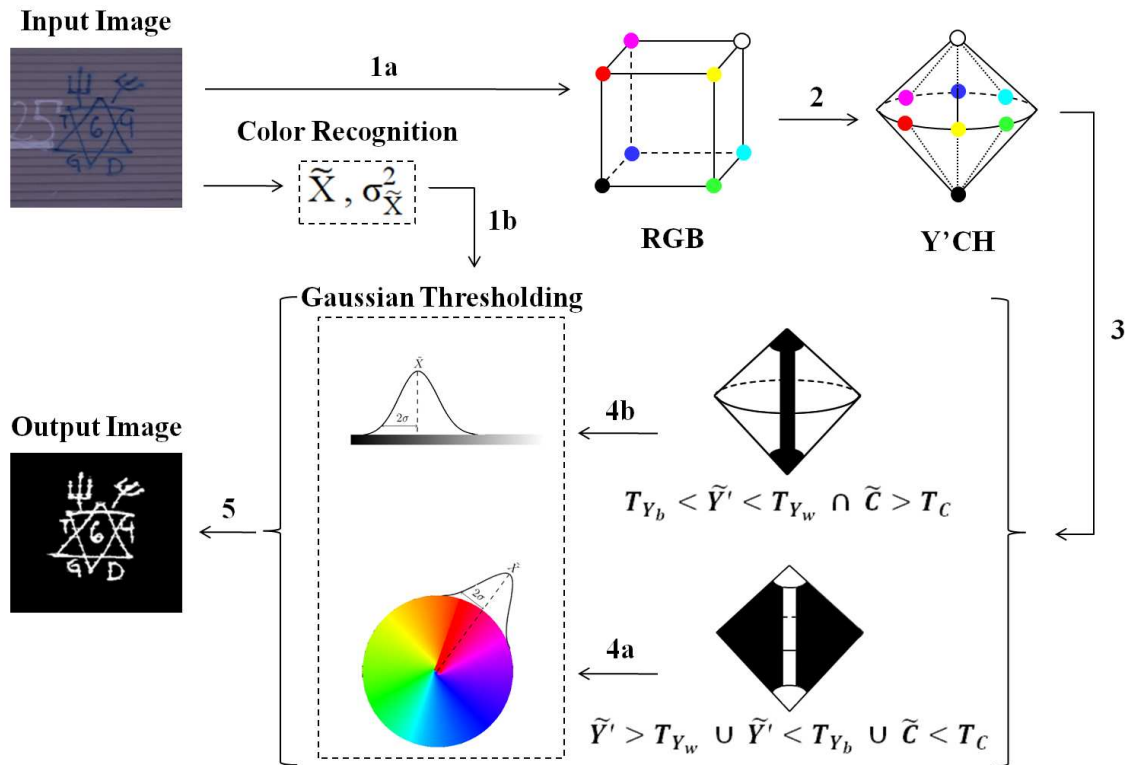


Fig. 3.10.: Color Image Segmentation Using Gaussian Thresholding.

Figure 3.11 shows an example where the color recognition is done by tracing a path along the blue numbers “2” and “5”. Figure 3.12 shows the effect of the Gaussian thresholding process on the letters “Hill”. Note that this method produces a probability map, where the values in a graffiti component decrease as the spray paint fades. This indicates how the graffiti was traced, and it may be useful in future research for shape analysis (Section 6). Appendix B illustrates more examples of our color segmentation method.



Fig. 3.11.: Gaussian Thresholding on Blue.  $(\tilde{H}, \sigma_{\tilde{H}}^2) = (4.19, 0.05)$ .

Table 3.5 shows all the parameters/thresholds we used including empirically derived parameters.



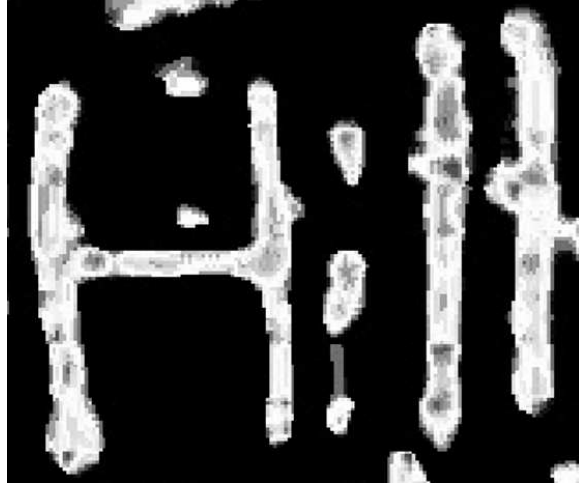


Fig. 3.12.: Probability Map Created By The Gaussian Thresholding.

Table 3.5: Parameters and thresholds used in Color Image Segmentation Based on Gaussian Thresholding.  $W_X$  and  $H_X$  are the width and height of  $X$  respectively.

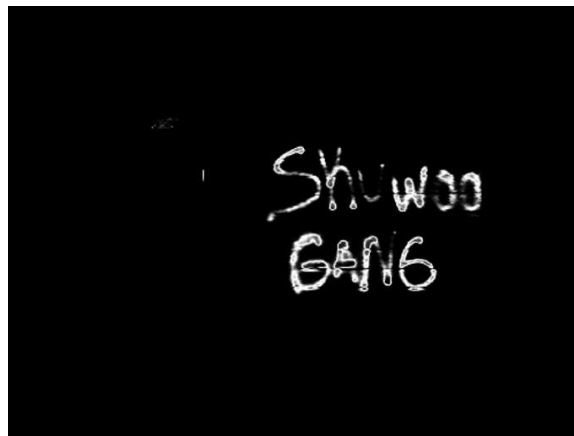
Parameter	Description	Value
$W_X$	Width of resized image for image segmentation	500 px
$T_{Y_w}$	Low luma threshold	0.12
$T_{Y_b}$	High luma threshold	0.85
$T_C$	Low chroma threshold	0.05

### 3.5.2 Block-Wise Gaussian Segmentation Enhancement

Since the median and variance for Gaussian thresholding are obtained from a small sample of the graffiti the resulting probability map  $X_g$  can contain broken or faded graffiti components and noise. These can be caused by either non-uniform scene illumination (Figure 3.13) or foreground-background hue similarity (Figure 3.14).



(a) Original Image. The traced path is marked in green.

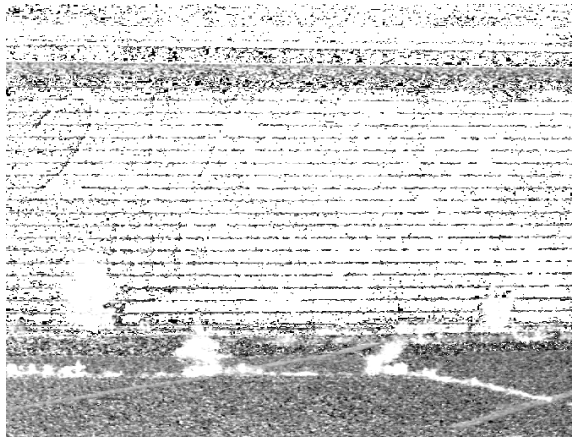


(b) Gaussian Thresholding

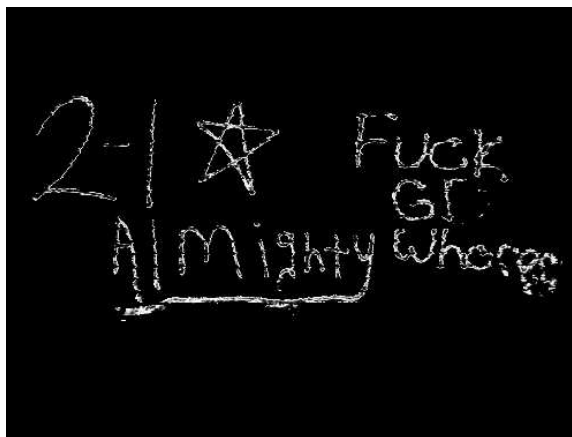
Fig. 3.13.: Gaussian Thresholding results with non-uniform scene illumination.



(a) Original Image. The traced path is marked in green.



(b) Hue Channel



(c) Gaussian Thresholding

Fig. 3.14.: Gaussian Thresholding results with foreground-background hue similarity.

Therefore, we need to enhance  $X_g$  before finding the graffiti components. This can be done by using a block-wise median filter on the luma, chroma and hue channels of the original image  $X$  separately and merging the results. Figure 3.15 shows the process.

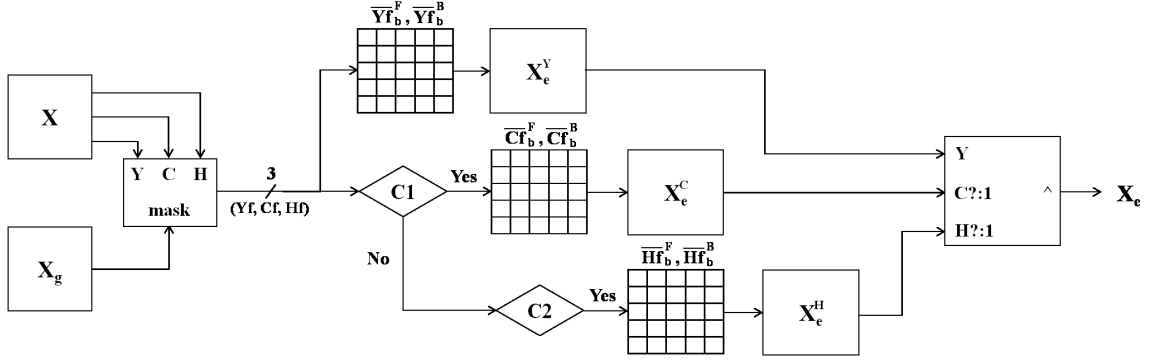


Fig. 3.15.: Block-Wise Gaussian Segmentation Enhancement.

First, we filter each channel on  $X$  with a binary mask created from  $X_g$ , so that

$$Yf(x, y) = \begin{cases} Y(x, y) & \text{if } X_g(x, y) > 0 \\ 0 & \text{else} \end{cases} \quad (3.19)$$

$$Cf(x, y) = \begin{cases} C(x, y) & \text{if } X_g(x, y) > 0 \\ 0 & \text{else} \end{cases} \quad (3.20)$$

$$Hf(x, y) = \begin{cases} H(x, y) & \text{if } X_g(x, y) > 0 \\ 0 & \text{else} \end{cases} \quad (3.21)$$

Then, we divide  $Yf$  in blocks of size  $w_s \times w_s$ , where  $w_s = 0.03 \max(W_X, H_X)$  and  $(W_X, H_X)$  are the width and height of  $X$  respectively. We only consider blocks  $b \in B$ , where  $B$  is the set of blocks containing at least one non-zero valued pixel. For each block  $b \in B$  we compute the luma median of the foreground pixels  $\widetilde{Yf}_b^F$  and the luma

median of the background pixels  $\widetilde{Y}f_b^B$ . Then, we generate the binary image  $X_e^Y$  by evaluating each individual pixel:

$$X_e^Y(x, y) = \begin{cases} 1 & \text{if } b \in B \text{ and } |Y(x, y) - \widetilde{Y}f_b^F| < |Y(x, y) - \widetilde{Y}f_b^B| \\ 0 & \text{else} \end{cases}, \quad (3.22)$$

where  $b$  is the block associated with the coordinates  $(x, y)$ . We use the chroma channel for enhancement if  $\frac{\sum_{b \in B} |\widetilde{C}f_b^F - \widetilde{C}f_b^B|}{n(B)} > T_e^C$  (condition C1 in Figure 3.15), where  $n(B)$  is the cardinality of  $B$ . A value of  $T_e^C = 0.06$  produced the best results after running experiments on more than 700 gang graffiti images. In that case,

$$X_e^C(x, y) = \begin{cases} 1 & \text{if } b \in B \text{ and } |C(x, y) - \widetilde{C}f_b^F| < |C(x, y) - \widetilde{C}f_b^B| \\ 0 & \text{else} \end{cases}, \quad (3.23)$$

If  $\frac{\sum_{b \in B} |\widetilde{C}f_b^F - \widetilde{C}f_b^B|}{n(B)} \leq T_e^C$  we can still use the hue channel for enhancement. If  $X_g$  was obtained using the hue channel during the Gaussian Thresholding (i.e.  $\widetilde{X} = \widetilde{H}$ ) (condition C2 in Figure 3.15) we apply an additional threshold to each pixel. In this case we keep pixels where the hue angular distances satisfy  $\theta(H(x, y), \widetilde{H}f_b^F) < \theta(H(x, y), \widetilde{H}f_b^B)$ , where  $\theta(a, b) = |\text{mod}(a - b, 2\pi) - \pi|$ . That is,

$$X_e^H(x, y) = \begin{cases} 1 & \text{if } b \in B \text{ and } \theta(H(x, y), \widetilde{H}f_b^F) < \theta(H(x, y), \widetilde{H}f_b^B) \\ 0 & \text{else} \end{cases}. \quad (3.24)$$

Therefore,

$$X_e = \begin{cases} X_e^Y \wedge X_e^C & \text{if } \frac{\sum_{b \in B} |\widetilde{C}f_b^F - \widetilde{C}f_b^B|}{n(B)} > T_e^C \\ X_e^Y \wedge X_e^H & \text{if } \widetilde{X} = \widetilde{H} \\ X_e^Y & \text{else} \end{cases}, \quad (3.25)$$

Table 3.6: Parameters and thresholds used in Block-Wise Gaussian Segmentation Enhancement.  $W_X$  and  $H_X$  are the width and height of  $X$  respectively.

Parameter	Description	Value
$w_s$	Block size for segmentation enhancement	$0.03\max(W_X, H_X)$
$T_e^C$	Chroma threshold for channel enhancement	0.06

where  $\wedge$  is the logical conjunction, also known as logical operator *and*. In the block diagram of Figure 3.15 the last module implements Equation 3.25 by doing  $X_e = X_e^Y \wedge X_e^C \wedge X_e^H$ , where  $X_e^C$  and  $X_e^H$  are set to an all-ones matrix 1 of the same size as  $X_e^Y$  if one or more of the conditions ( $C1$ ,  $C2$ ) are not satisfied. That is, if  $C1$  is satisfied  $X_e^H = 1$ ; if  $C1$  is not satisfied and  $C2$  is satisfied  $X_e^C = 1$ ; if both  $C1$  and  $C2$  are not satisfied  $X_e^C = X_e^H = 1$ . Note that if we use the chroma channel enhancement we ignore the hue channel enhancement. This is because our experiments showed that if the condition for hue enhancement is satisfied the chroma enhancement does not improve the output. Also note how when  $X_e = X_e^Y \wedge X_e^C$  the chrome enhancement can introduce some noise, which is removed using luma enhancement. Figures 3.16 and 3.17 show an example of the entire process. Note how  $X_e$  removes noise and enhances the graffiti, but also enhances some non-graffiti areas at the bottom. However, this areas will not be connected to graffiti components and we will be able to discard them in future steps.

Table 3.6 shows all the parameters/thresholds we used including empirically derived parameters.

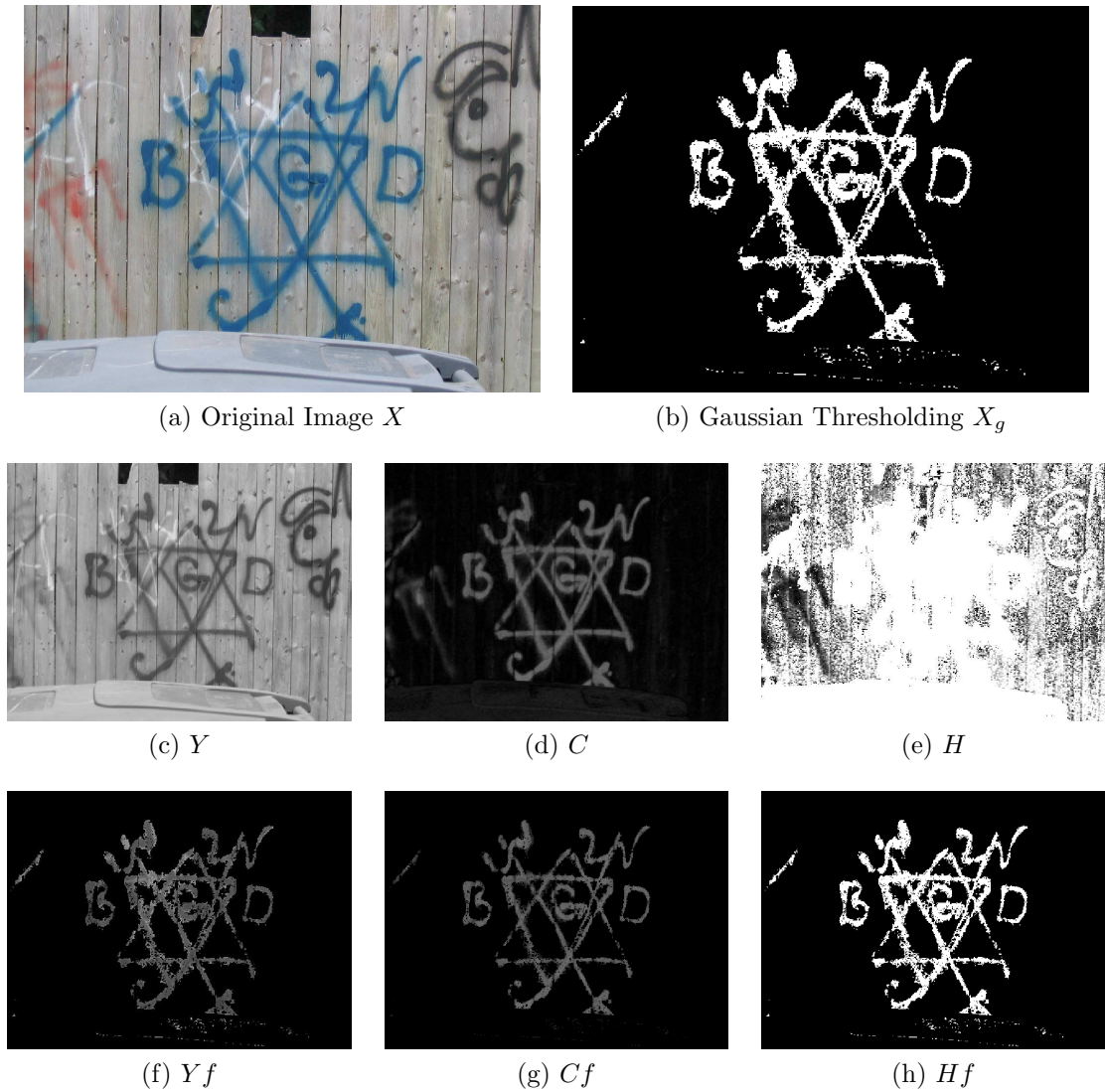


Fig. 3.16.: Example of Block-Wise Gaussian Segmentation Enhancement.

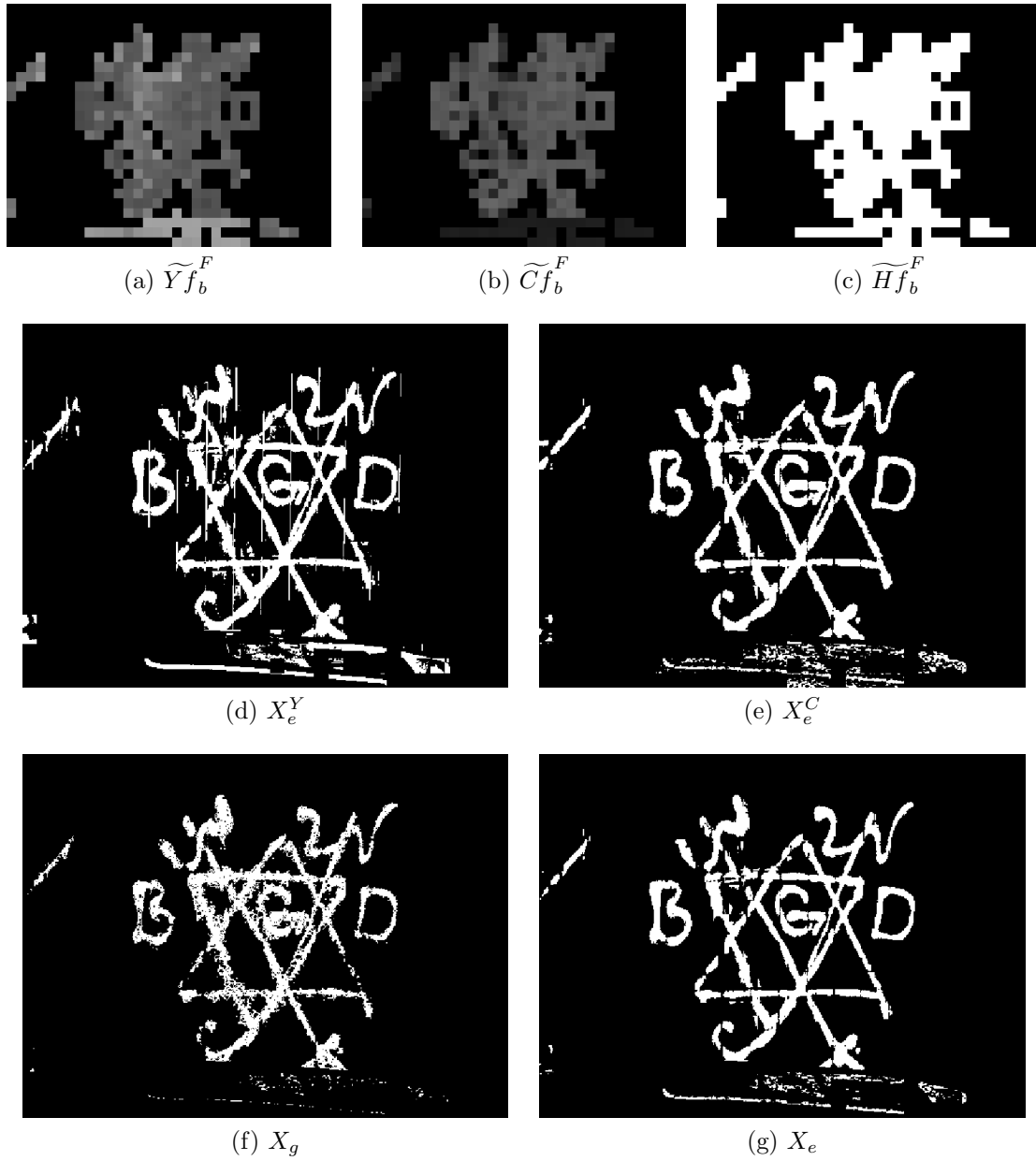


Fig. 3.17.: Example of Block-Wise Gaussian Segmentation Enhancement (continued).



### 3.5.3 Background Stripe Removal

Gang graffiti are sprayed in all kinds of surfaces, including brick walls, garage doors and fences. All these surfaces contain stripes that can affect the graffiti component extraction. Figure 3.18 shows an example of a gang graffiti image after applying Block-Wise Gaussian Segmentation Enhancement. These stripes interfere with the segmentation by linking multiple gang graffiti components. Figure 3.19 shows the process to remove the background stripes. Note that sometimes the color of the background stripes is different from the graffiti itself, and the Color Image Segmentation Based on Gaussian Thresholding step already removes the stripes. Figure 3.20 shows an example.



Fig. 3.18.: Background stripes affecting gang graffiti component segmentation.

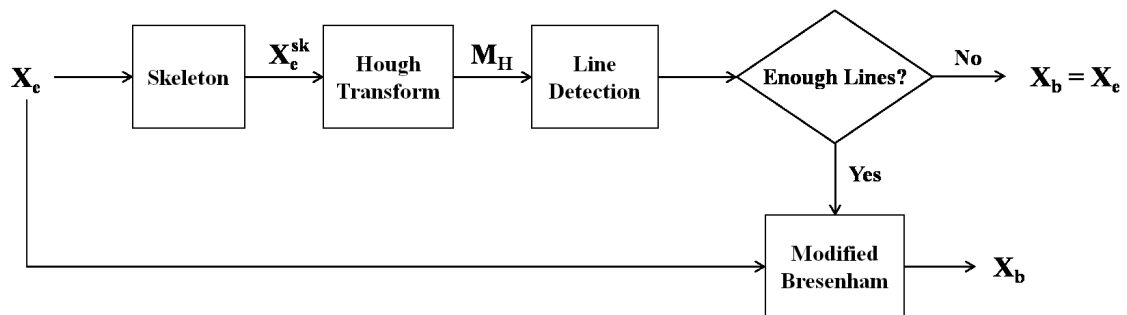


Fig. 3.19.: Background Stripe Removal.

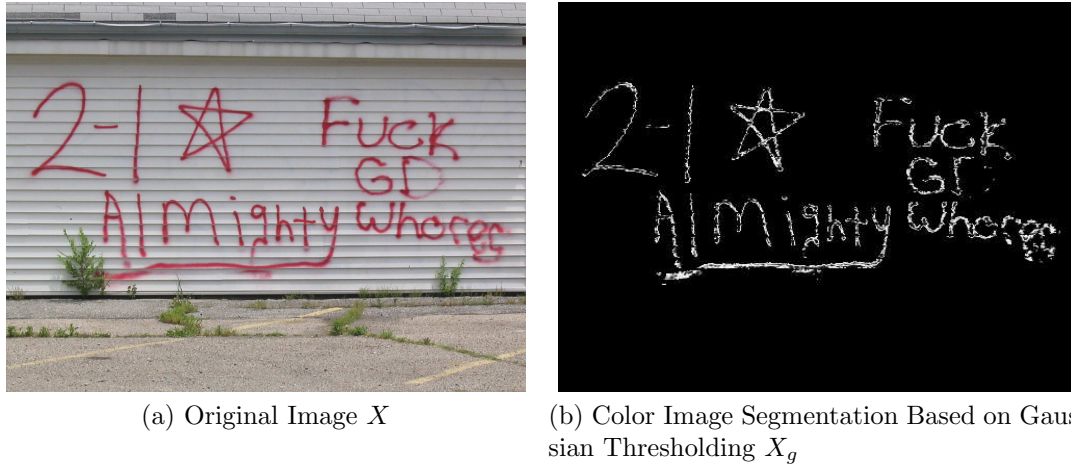


Fig. 3.20.: Example of Background Stripes Removal During the Gaussian Thresholding Step.

First, we compute the skeleton  $X_e^{sk}$  of the input image  $X_e$ , the result of the Block-Wise Gaussian Segmentation Enhancement, which is binary. The skeleton is obtained using parallel thinning [224, 225] as follows. We define the set  $S$  as the set of all 1-valued pixels (ones) of  $X_e^{sk}$  representing objects (connected components) to be thinned. We define the set  $\bar{S}$  as the set of all 0-values pixels (zeros) of  $X_e^{sk}$  representing either the background of or holes in  $S$ . The connectivities for  $S$  and  $\bar{S}$  are set to 8-connectivity and 4-connectivity respectively. Figure 3.21 illustrates the meaning of 8-connectivity and 4-connectivity in a  $3 \times 3$  support around a pixel  $p$ .

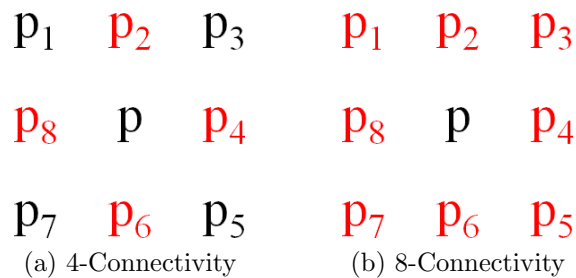


Fig. 3.21.: Connectivity of  $p$ . Pixels are connected to  $p$  if they have the same value as  $p$ . Only pixel locations in red are considered in each connectivity.

We define  $C(p)$  as the number of distinct 8-connected components of ones in  $p$ 's 8-neighborhood.  $C(p) = 1$  implies  $p$  is 8-simple when  $p$  is a boundary pixel [89]. We define  $N(p)$  as

$$N(p) = \min(N_1(p), N_2(p)), \quad (3.26)$$

where

$$N_1(p) = (p_1 \vee p_2) + (p_3 \vee p_4) + (p_5 \vee p_6) + (p_7 \vee p_8) \quad (3.27)$$

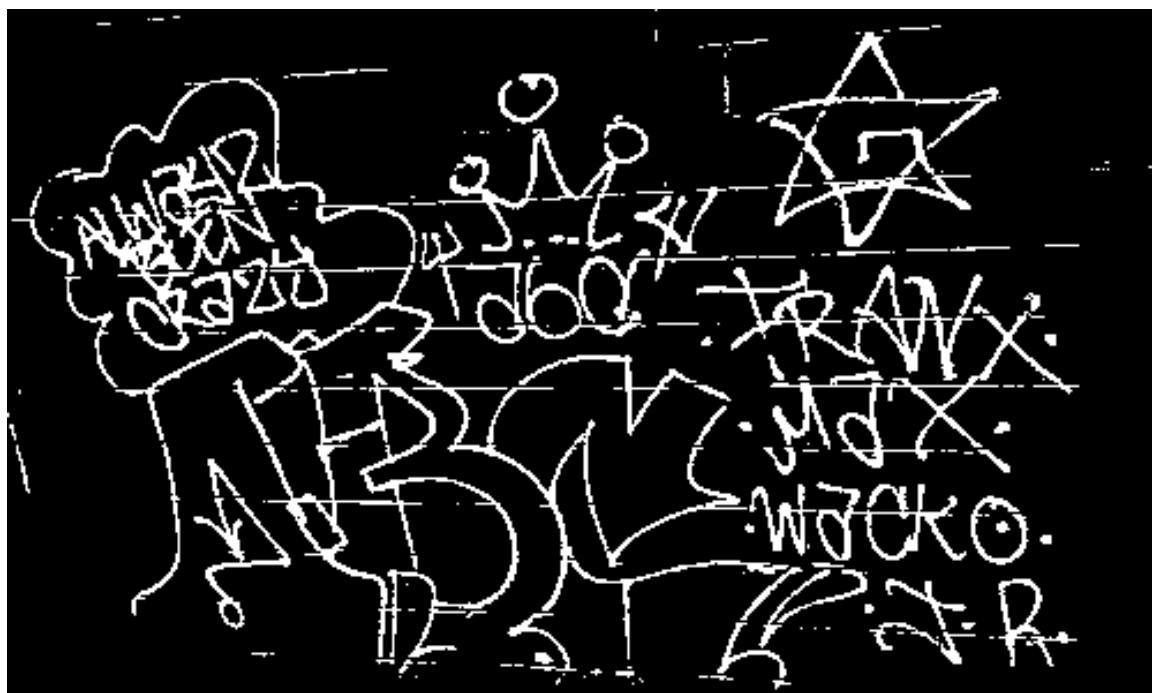
and

$$N_2(p) = (p_2 \vee p_3) + (p_4 \vee p_5) + (p_6 \vee p_7) + (p_8 \vee p_1). \quad (3.28)$$

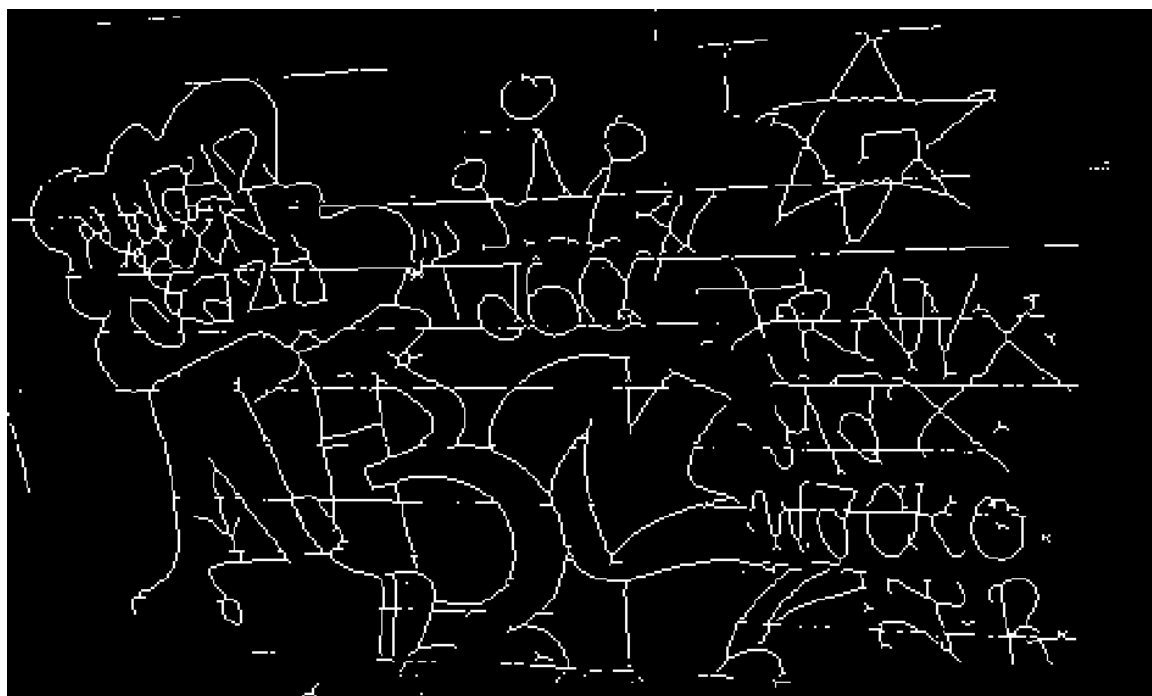
The symbols  $\vee$  and  $+$  are logical OR and arithmetic addition respectively. Note that  $N_1(p)$  and  $N_2(p)$  divide the ordered set of neighbors of  $p$  into four pairs of adjoining pixels and count the number of pairs that contain one or two ones. The thinning process is applied to each pixel  $p \in S$ .  $p$  is deleted (i.e. changing one to zero) if all the following conditions are met:

1.  $C(p) = 1$
2.  $T_{N(p)}^L \leq N(p) \leq T_{N(p)}^H$
3. Either
  - (a)  $(p_2 \vee p_3 \vee \bar{p}_5) \vee p_4 = 0$  in odd iterations
  - (b)  $(p_6 \vee p_7 \vee \bar{p}_1) \wedge p_8 = 0$  in even iterations

where  $T_{N(p)}^L = 2$ ,  $T_{N(p)}^H = 3$ , and  $\bar{p}$  and  $\wedge$  are logical complement and logical AND respectively. The thinning stops when no further deletions are possible. Figure 3.22 shows an example of skeletonization via parallel thinning to obtain  $X_e^{sk}$ .



(a) Binary Image  $X_e$



(b) Parallel Thinning  $X_e^{sk}$

Fig. 3.22.: Skeletonization via Parallel Thinning [225].

The next step is to find straight lines using the Standard Hough Transform (SHT) [226, 227]. The method uses the parametric representations of a line to populate a 2-dimensional matrix  $M_H$  called accumulator array, where its rows and columns correspond to  $\rho$  and  $\theta$  values of  $\rho = x \cos(\theta) + y \sin(\theta)$  respectively. Figure 3.23 illustrates the parametric representation of a line.

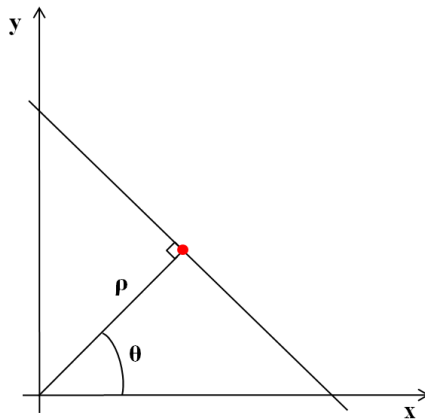


Fig. 3.23.: Parametric Representation of a Line.

First, each cell in  $M_H$  is initialized to zero. For each non-zero pixel in  $X_e^{sk}$  the accumulator cells are updated so that  $M_H(i, j)$  keeps a count of the number of pixels in the  $XY$  plane represented by  $\rho(i)$  and  $\theta(j)$ . Peak values in  $M_H$  represent potential lines in  $X_e^{sk}$ . We Figure 3.24 shows the Hough accumulator array  $M_H$  with highlighted peaks. There are 13 potential lines divided in two sets of  $\theta$  around  $\pi$  and  $-\pi$ , which actually correspond to the same set.

Given the nature of the background stripes in gang graffiti images we limit the number of peaks to  $N_{peaks} = 15$ . For each peak we find the location of all nonzero pixels in the image that contributed to that peak and determine the line segments based on those pixels. Each segment is now represented by a set  $(\theta, \rho, p_i, p_f)$ , where  $(p_i, p_f)$  are the initial and final points of the segment. We discard segments of length less than  $T_{minlen}^W = 0.4W_X$  if the segment is closer to the horizontal plane and less than  $T_{minlen}^H = 0.6H_X$  if the segment is closer to the vertical plane.  $W_X$  and  $H_X$  are the width and the height of the image, respectively. If we have less than  $N_{seg} = 4$

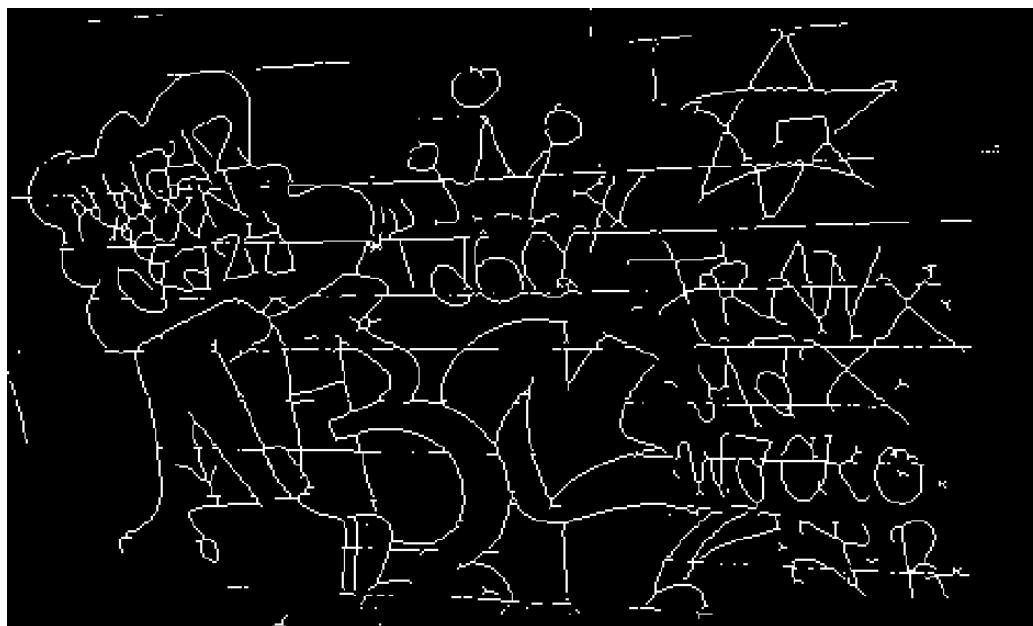
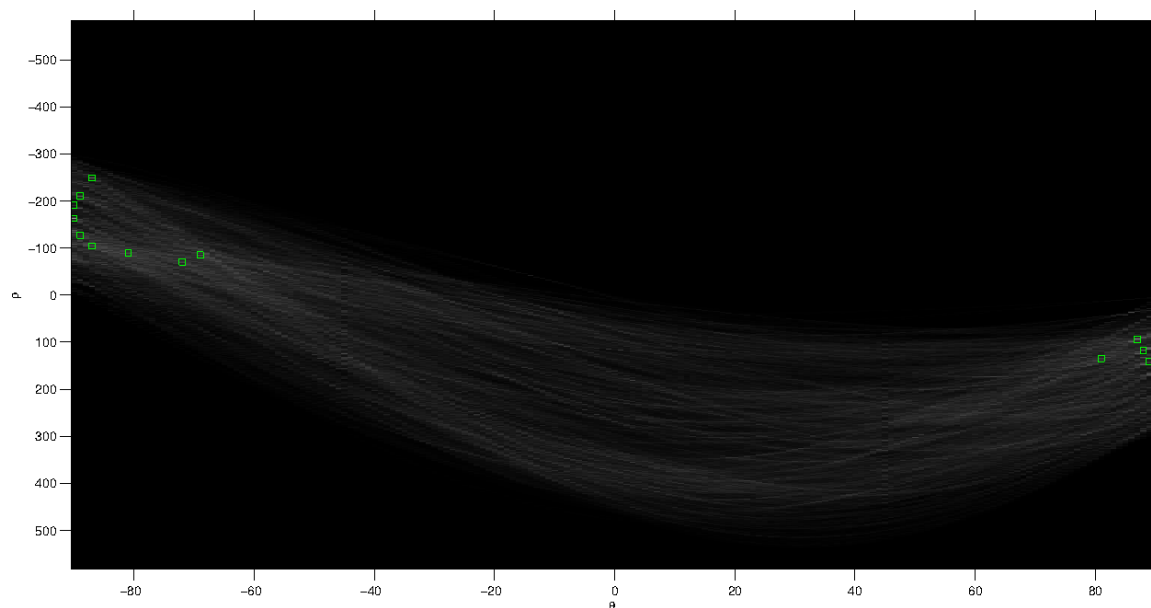
(a) Skeleton  $X_e^{sk}$ (b)  $M_H$ 

Fig. 3.24.: Standard Hough Transform accumulator array. Peaks corresponding to potential lines are marked with green squares.

segments remaining we consider them not to be background stripes, and there is nothing to be done. Else, we need to remove the segments without affecting the graffiti components they may intersect with.

To do that we propose a modification of the Bresenham's technique [228]. The original method retrieves a set of pixels locations  $S_{(x,y)}$  from a given line represented by a set of initial and final points  $(p_i, p_f)$ . Figure 3.25 illustrates the conversion from  $(p_i, p_f)$  to  $S_{(x,y)}$ . The pixels in  $S_{(x,y)}$  are marked in gray. Figure 3.26 shows a step of the process when a pixel location (shown in yellow at  $(x, y)$ ) has been already added to  $S_{(x,y)}$ . Since the line does not fall into the actual pixel grid the next sampled location (shown in yellow at  $(x + 1, y + 1)$ ) will have an error  $\epsilon$  on the  $y$  direction. Note that this error ranges between  $-0.5$  to  $0.5$ . The next point to be added to  $S_{(x,y)}$  can either be  $(x + 1, y)$  or  $(x + 1, y + 1)$ . We choose  $(x + 1, y)$  if  $y + \epsilon + m < y + 0.5$ , and we choose  $(x + 1, y)$  otherwise. By doing so we minimize the total error between the mathematical line segment and what we actually add to  $S_{(x,y)}$ .

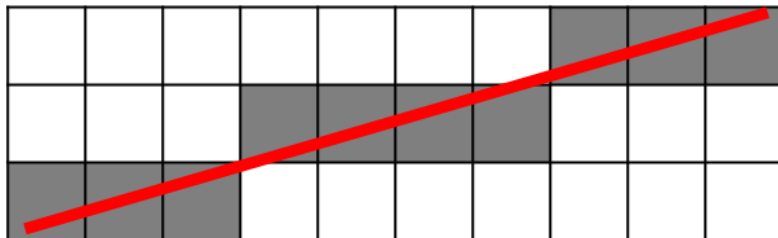


Fig. 3.25.: Bresenham's Technique: mathematical line (red) and elements of  $S_{(x,y)}$  (gray).

Our modification to the original method includes an estimation of the segment width at each new location added to  $S_{(x,y)}$ . For this purpose we need to use the binary image  $X_e$  instead of its skeleton  $X_e^{sk}$ . At each new location  $(x, y)$  we create a window of radius 1 around it and compute the ratio  $R$  as

$$R = \frac{\# \text{ ones inside window}}{\# \text{ pixels inside window}} \quad (3.29)$$

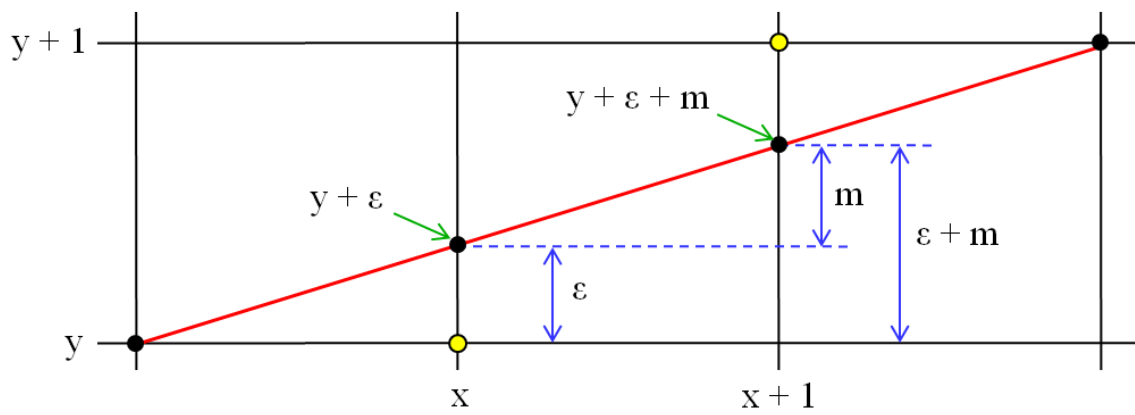


Fig. 3.26.: Step of Bresenham's Technique.



If  $R > T_{rad}^H$  we increase the window size by one and recompute  $R$ . We repeat the process until  $R \leq T_{rad}^H$ . We choose  $T_{rad}^H = 0.6$  as it gave us the best results in our experiments. Figure 3.27 illustrates the final size of the window at different locations. Note that even though the segment can have an arbitrary orientation the window is always aligned with the  $XY$  axes. This is because we just need an estimate of the segment width.

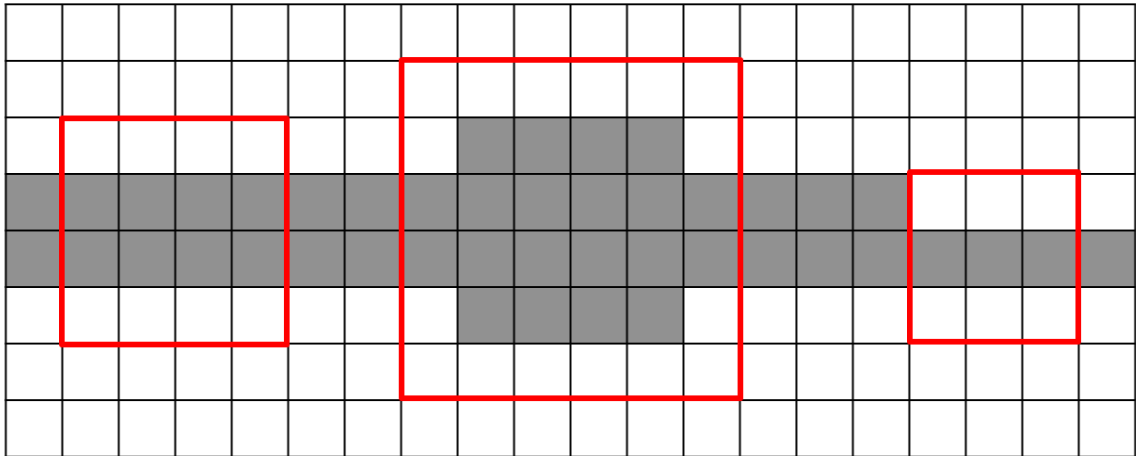


Fig. 3.27.: Final window sizes at different locations using our modified Bresenham's Technique.

Once we have all the segment width estimates for all the pixel locations in  $S_{(x,y)}$  we set the segment width to the most frequent estimated width (i.e. the width mode). The pixel locations with width larger than the mode are considered to be intersections with graffiti components, and they are left untouched. The rest of the pixel locations are removed from the binary image. After all the line segments are processed we obtain the binary image  $X_b$ . Figure 3.28 shows an example of our proposed modified Bresenham's Technique. The green areas correspond to removed line segments, and the blue areas correspond to ignored line segments. Figure 3.29 shows an example of the entire Background Stripe Removal process. Note how some of the line segments actually corresponding to background stripes are not removed. However, we have removed the segments that connect different graffiti components, and they can now

Table 3.7: Parameters and thresholds used in Background Stripe Removal.  $W_X$  and  $H_X$  are the width and height of  $X$  respectively.

Parameter	Description	Value
$T_{N(p)}^L$	Low threshold for thinning	2
$T_{N(p)}^H$	High threshold for thinning	3
$N_{peaks}$	Number of Hough peaks	15
$T_{minlen}^W$	Threshold to discard horizontal segments	$0.4W_X$
$T_{minlen}^H$	Threshold to discard vertical segments	$0.6H_X$
$N_{seg}$	Number of segments to keep	4
$T_{rad}^H$	High threshold for line width	0.6

be separated.

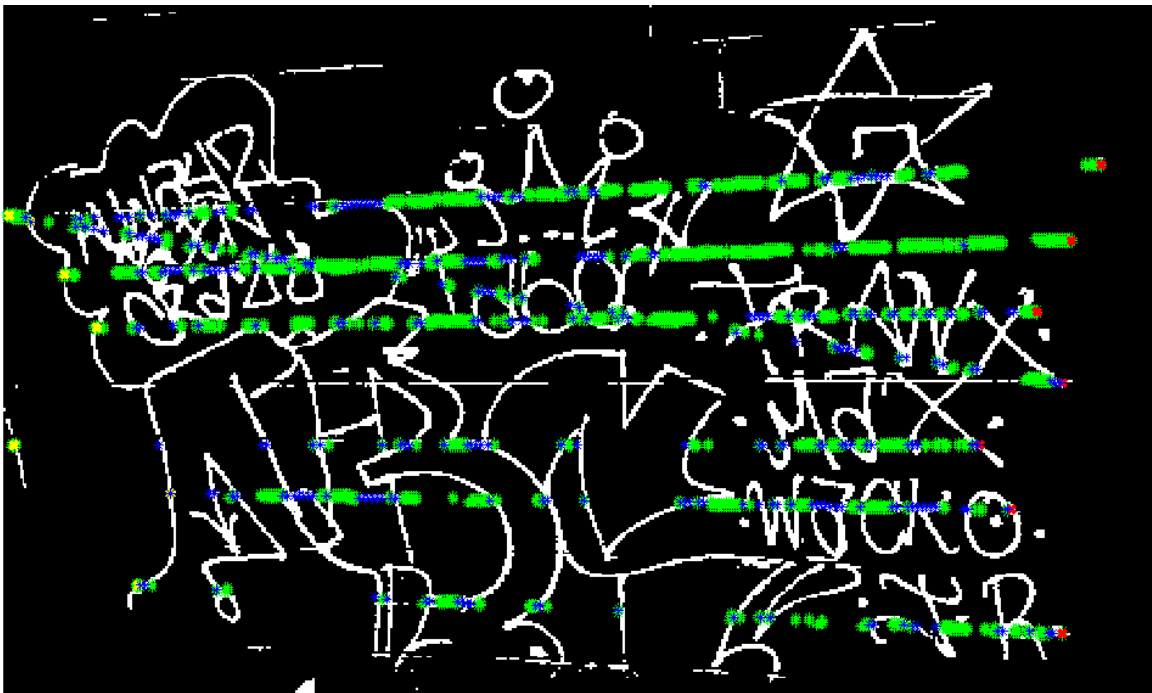


Fig. 3.28.: Modified Bresenham Technique. Green areas correspond to removed line segments; blue areas correspond to ignored line segments.

Table 3.7 shows all the parameters/thresholds we used including empirically derived parameters.

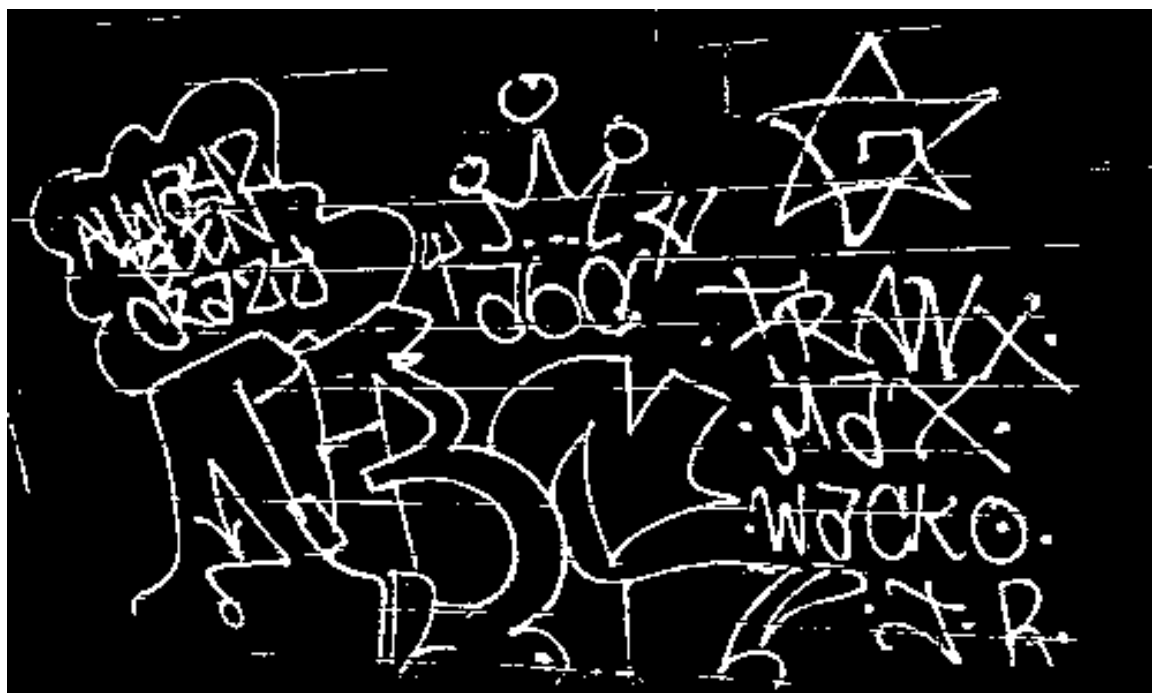
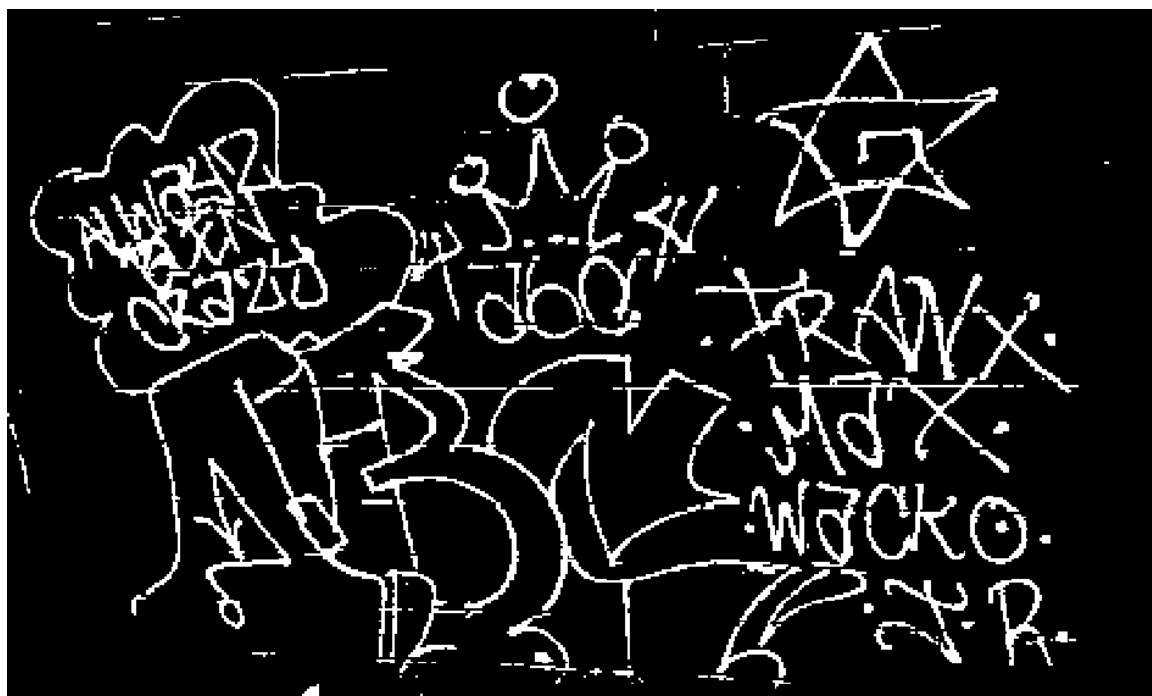
(a) Input:  $X_e$ (b) Output:  $X_b$ 

Fig. 3.29.: Example of Background Stripe Removal.

### 3.5.4 Graffiti Component Reconnection

Even after Block-Wise Gaussian Segmentation Enhancement and Background Stripe Removal there are still broken gang graffiti components that need to be reconnected for efficient segmentation. For this purpose we consider a line reconstruction method used in topographic map enhancement [147, 229]. Figure 3.30 shows the process to reconnect graffiti components.

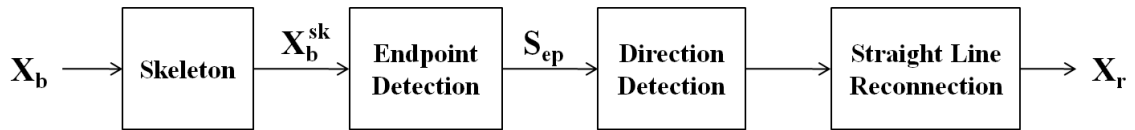


Fig. 3.30.: Graffiti Component Reconnection.

First, we compute the skeleton  $X_b^{sk}$  of the input image  $X_b$ , the result of the Background Stripe Removal, which is already binary. The skeleton is obtained using the method already described in Section 3.5.3. We then detect the endpoints of  $X_b^{sk}$ . An endpoint is defined to have exactly one neighbor pixel. Figure 3.31 illustrates all the possible  $3 \times 3$  templates of an endpoint. Figure 3.32 shows an example of detected endpoints.

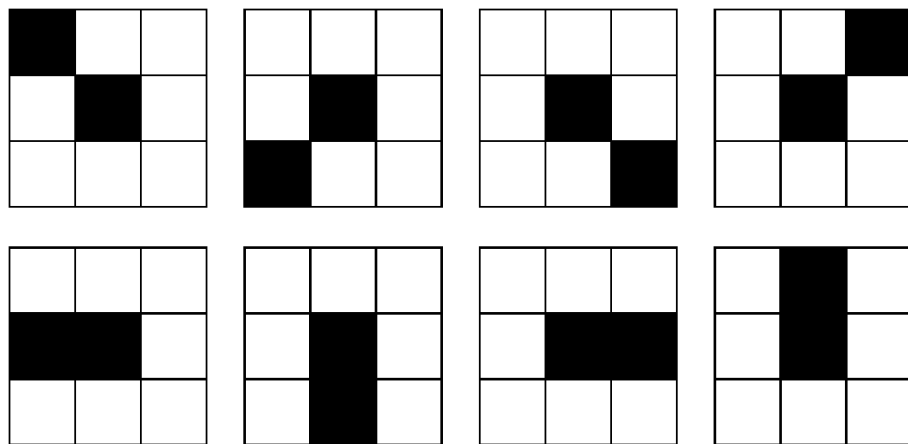


Fig. 3.31.:  $3 \times 3$  templates to detect an endpoint. The endpoint is at the center of the template.

For each endpoint  $e_0$  we create a  $h \times h$  search window around it and build the set  $S_{ep} = \{e_1, e_2, \dots, e_n\}$  with the  $n$  endpoints within the search window. We selected  $h = 20$  as proposed in [147]. Note that we ignore any endpoints that are 8-neighbor connected to  $e_0$  (i.e. part of the same connected component). For each endpoint  $e_i \in S_{ep}$  we detect its direction by constructing a chain code as shown in Figure 3.33. We backtrace  $N_{px}^{bt} = 5$  pixels and assign a zone based on the possible directions 0 – 7 according to Table 3.8.

Table 3.8: Relationship Between Directions and Zones in the Chain Code.

<b>Directions</b>	<b>Zone</b>
1, 2	Zone 1
3, 4	Zone 2
5, 7	Zone 3
7, 8	Zone 4

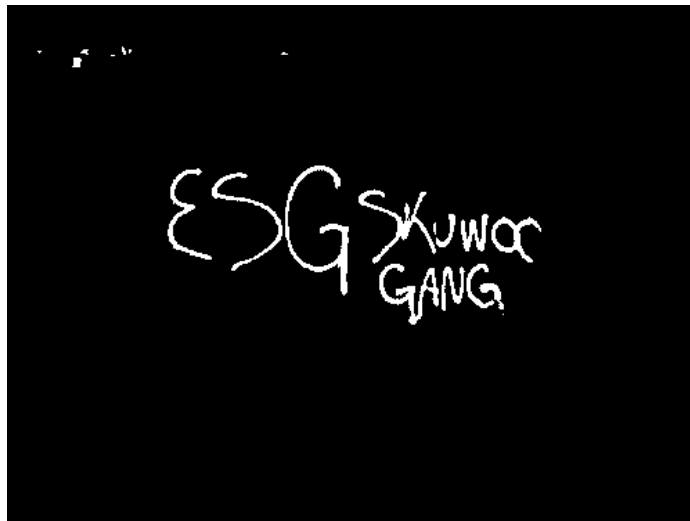
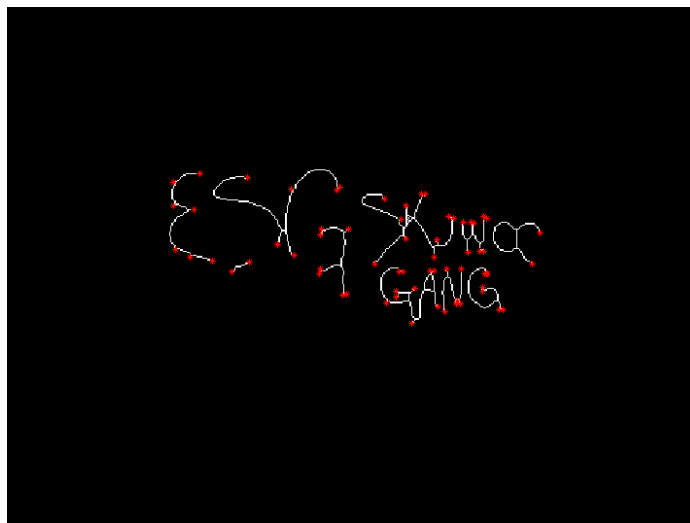
(a) Original Image  $X$ (b)  $X_b$ (c) Endpoints on  $X_b^{sk}$ 

Fig. 3.32.: Endpoint Detection.

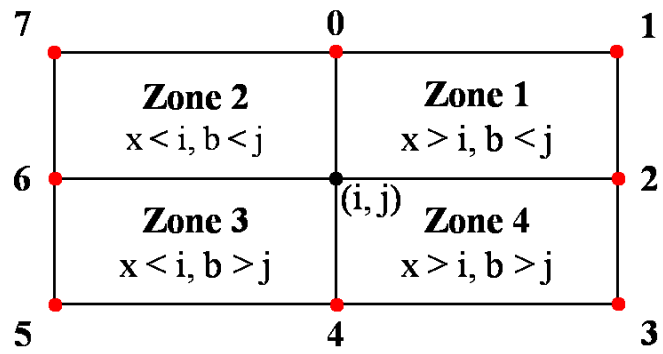


Fig. 3.33.: Chain Code For Endpoint Direction Detection.

We remove from  $S_{ep}$  all the endpoints that do not satisfy the following conditions with respect to  $e_0$ :

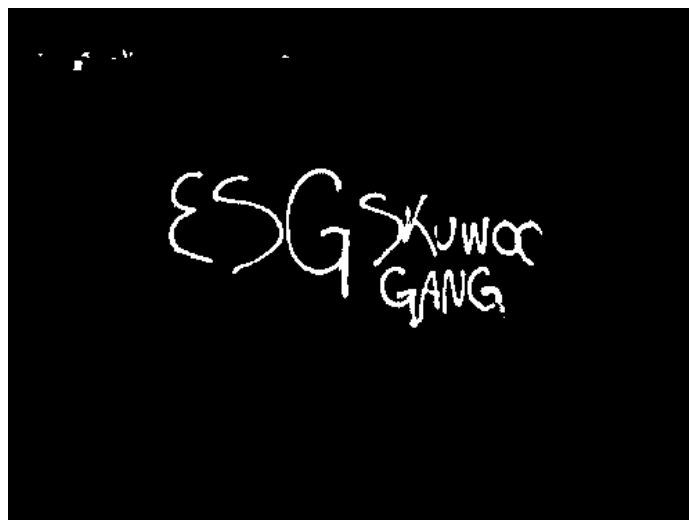
- For opposite directions:
  - Zone 1 opposite to Zone 3
  - Zone 2 opposite to Zone 4
- For parabolic directions:
  - Zone 1 parabolic with Zone 4
  - Zone 4 parabolic with Zone 3
  - Zone 3 parabolic with Zone 2
  - Zone 2 parabolic with Zone 1

If there are more than one remaining endpoints in  $S_{ep}$  we chose the one closest to  $e_0, e_d$ . The method presented in [147] does reconnection between  $e_0$  and  $e_d$  with Cubic Spline Interpolation or Newton Interpolation Method [230]. Since we are just interested in combining disconnected components for classification and not reconstructing them we reconnect  $e_0$  and  $e_d$  with a straight line. After all the endpoints are processed we obtain the binary image  $X_r$ . Figure 3.34 shows an example of the Graffiti Component Reconnection process.

At this point each individual graffiti component corresponds to an 8-neighbor connected component. Figure 3.35 shows an example of the connected component extraction before and after the Automatic Graffiti Component Segmentation. Note that currently we do not try to connect different letters on the same word. Given the handwritten nature of the graffiti (e.g. “y” in Figure 3.35b) it is difficult to discern between words and symbols.

Note that this method can also be used to reconstruct graffiti components that are broken because of being crossed-out by other graffiti component sprayed using



(a)  $X_b$ 

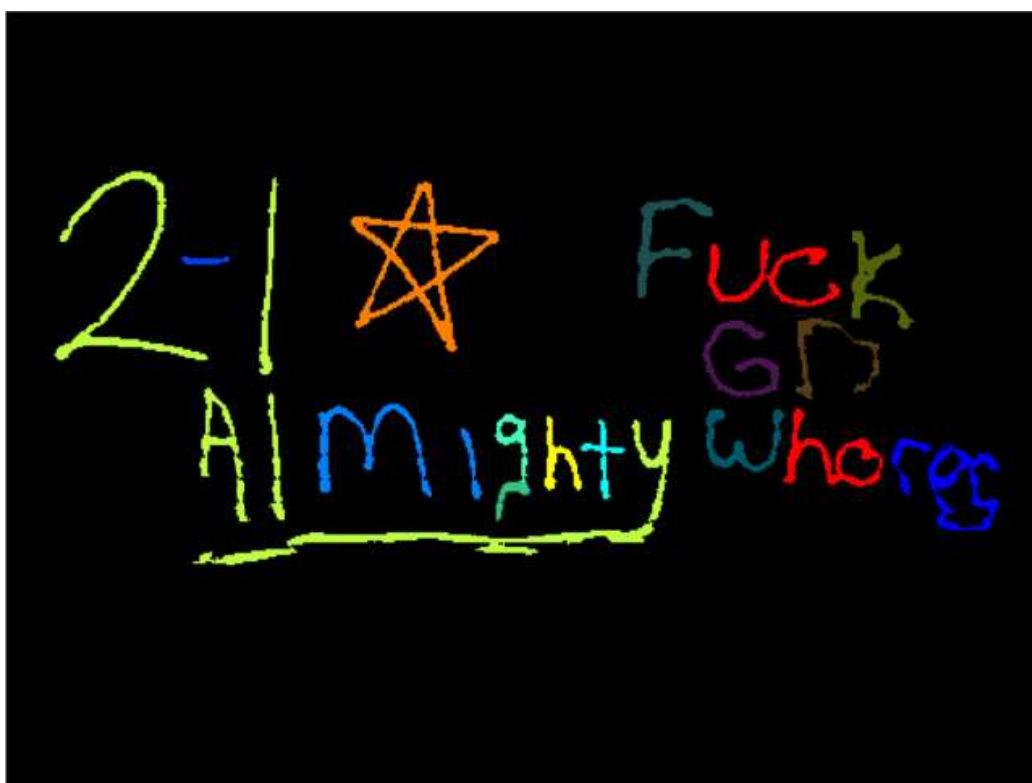
(b) Reconnected Components.

(c)  $X_r$ 

Fig. 3.34.: Example of Graffiti Component Reconnection.



(a) Connected Components of  $X_g$  (Output of Gaussian Thresholding)



(b) Connected Components of  $X_r$  (Output of Graffiti Component Reconnection)

Fig. 3.35.: Example of connected components after Gaussian Thresholding and after Graffiti Component Reconnection.

Table 3.9: Parameters and thresholds used in Graffiti Component Reconnection.

<b>Parameter</b>	<b>Description</b>	<b>Value</b>
$h$	Endpoint search window size	20
$N_{px}^{bt}$	Number of backtracing pixels	5

different color.

Table 3.9 shows all the parameters/thresholds we used including empirically derived parameters.

### 3.6 Gang Graffiti Features

The GARI system provides gang graffiti image retrieval in two scenarios: 1) recognize scenes containing graffiti and 2) classify individual graffiti components. We explain both scenarios in detail in Section 3.7.

For scene recognition we find SIFT features from the entire image, similar to the work done in [7, 8, 10–12, 18] for graffiti and tattoo images. SIFT is invariant to location, scale and rotation, and it is robust to affine transformations and illumination changes and viewpoint. The process to create SIFT descriptors from an image can be summarized as follows.

First, we find all the local extrema in the Difference of Gaussian (DoG) pyramid [15, 231]. A Gaussian pyramid for an image is generated by smoothing it with successively larger Gaussian functions

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (3.30)$$

and arranging the sequence of smoothed images in the form of a stack. Each level of the Gaussian pyramid is one octave above the level below (i.e. doubling the value of  $\sigma$ ). A DoG image  $D(x, y, \sigma)$  at scale  $\sigma$  is defined as

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma), \quad (3.31)$$

where  $L(x, y, k\sigma)$  is the convolution of the original image with  $G(x, y, k\sigma)$ . Figure 3.36 illustrates how the DoG pyramid is generated.

The local extrema (keypoints) are detected from the subpixel minima/maxima in the DoG pyramid by comparing neighboring pixels across scales, as shown in Figure

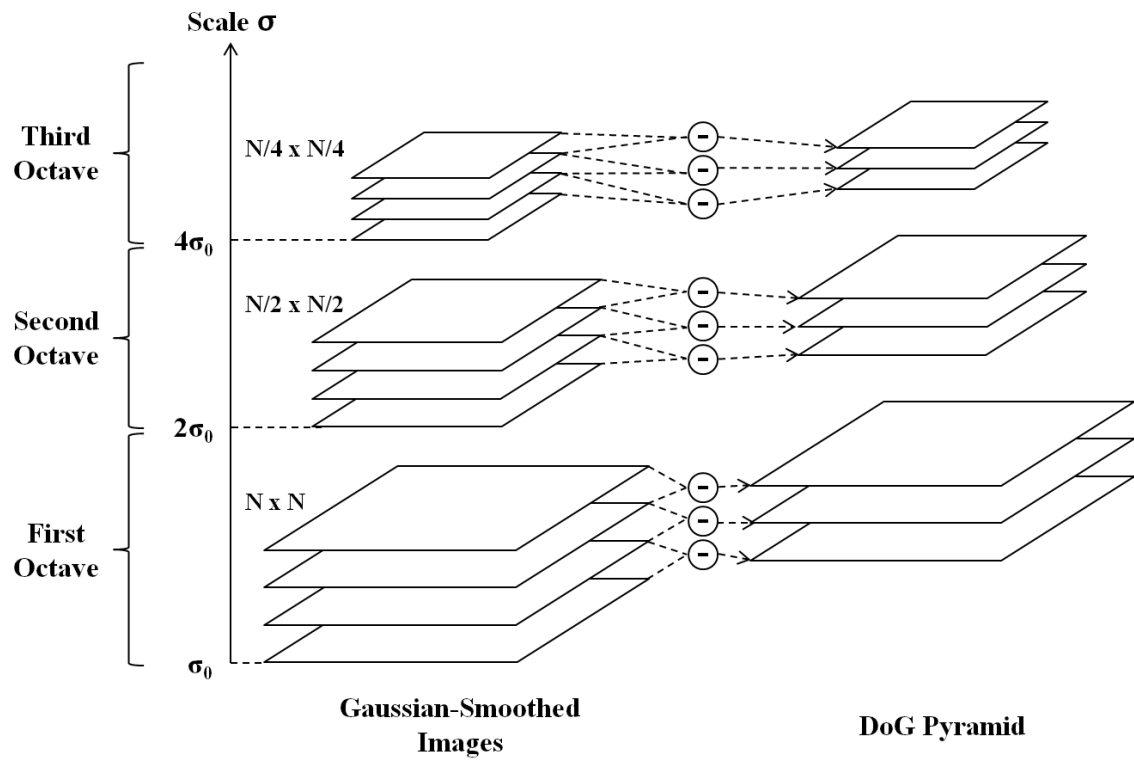


Fig. 3.36.: DoG Pyramid.

3.37. The subpixel accuracy is interpolated using the quadratic Taylor expansion of  $D(x, y, \sigma)$  with the candidate keypoint  $x = (x, y, \sigma)$  as the origin:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D^T}{\partial x^2} x \quad (3.32)$$

Weak extrema are discarded by rejecting keypoints that satisfy  $|D(x)| < 0.03$ .

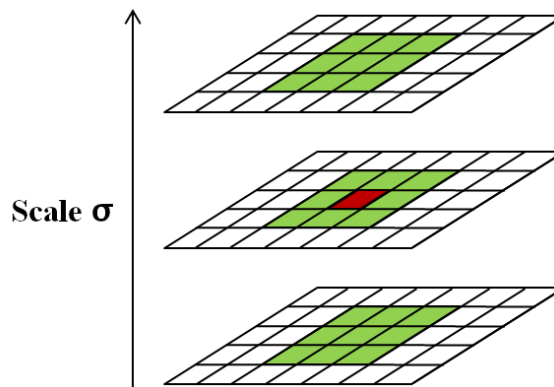
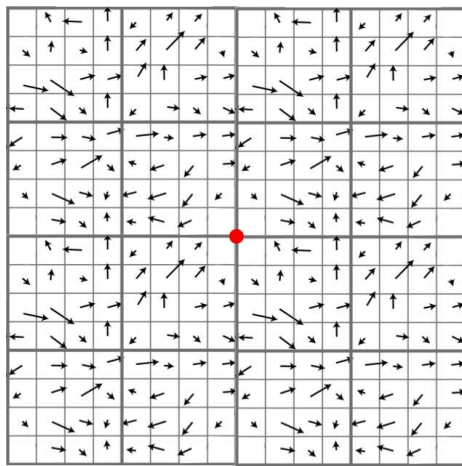


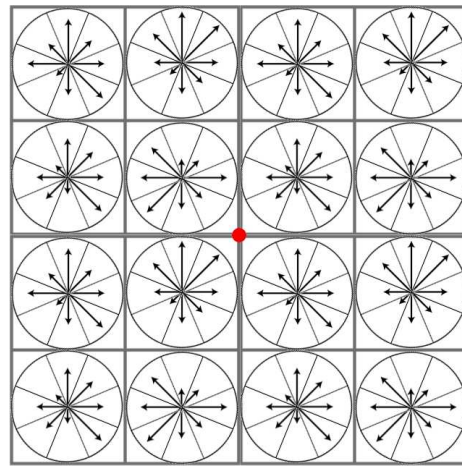
Fig. 3.37.: Neighboring Pixels (green) For Keypoint Extraction (red).

We then associate a dominant local orientation to a keypoint by constructing a histogram of gradient orientations using 36 bins spanning  $360^\circ$ . The bin where the histogram peak occurs decides the dominant local orientation. By representing the keypoint relative to its orientation the SIFT descriptor achieves rotation invariance.

Finally, the SIFT descriptor is created by surrounding each keypoint with a  $16 \times 16$  descriptor window divided into  $4 \times 4$  cells. The gradient magnitudes in the descriptor window are weighted by a Gaussian function with  $\sigma$  equal to half the width of the neighborhood. For each of the 16 cells an 8-bin orientation histogram is determined, thus creating a 128-dimensional descriptor with its length normalized to make it robust to changes in illumination. Figure 3.38 shows a graphical representation of the keypoint descriptor generation. Figure 3.39 shows some examples of extracted SIFT keypoints overlapped on the input images.



(a) Gradient Magnitudes Around Descriptor Window



(b) 16 8-Bin Orientation Histograms (128-dimensional)

Fig. 3.38.: Keypoint Descriptor Generation. The red dot represents the location of the keypoint.



(a) SIFT Descriptors



(b) Gradient Magnitude Histograms

Fig. 3.39.: 25 SIFT descriptors selected at random. Each keypoint is represented by a set of gradient magnitude histograms (green) rotated to its dominant local orientation (yellow). The size of the green grid represents the scale of the descriptor.



For individual gang graffiti component classification we do not use SIFT descriptors directly, but the spatial locations of the SIFT keypoints to create Local Shape Context (LSC) descriptors similar to the work proposed in [30, 183]. We do this because graffiti components are handwritten shapes with intra-class inconsistencies and small shape distortions that are not fully captured with SIFT descriptors. Also, SIFT descriptors accommodate for illumination changes and complex textures, which are not present in binarized graffiti components.

First, we find the gang graffiti components as individual connected components from the output of the Automatic Graffiti Component Segmentation in Section 3.5. For each graffiti component we then find  $N_f$  SIFT keypoint locations. Each location  $f_i$  needs to be compared against the other  $N_f - 1$  locations to create a LSC descriptor. This is done by binning the locations into a histogram, where its bins are broad enough to allow for small shape distortions and orientation variation. Our proposed histogram is defined with  $n_r = 3$  concentric circumferences representing log-radial distance bins and  $n_\theta = 16$  equally spaced sectors representing angles. We use a log-radial increment because we want to give more importance to the neighbor features than the rest. A histogram is centered at  $f_i$  and its bins are populated by calculating the distances

$$r_i^j = \frac{\|f_i - f_j\|_2}{\bar{r}_i} \quad (3.33)$$

and the angles

$$\bar{\theta}_i^j = \theta_i^j - \theta_i, \quad (3.34)$$

Table 3.10: Parameters and thresholds used for the Gang Graffiti Features.

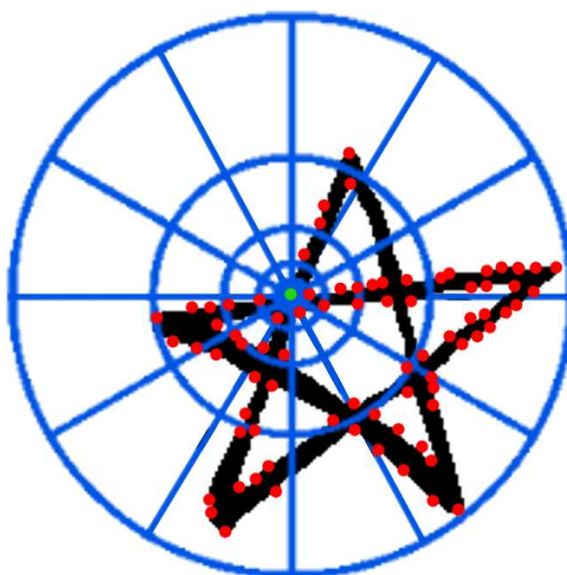
Parameter	Description	Value
$n_r$	Number of log-radial distance bins	6
$n_\theta$	Number of angular bins	19

for all  $j \in [1, N_f]$  and  $j \neq i$ , where  $\bar{r}_i$  is the average distance between  $f_i$  and the rest of locations,  $\theta_i^j$  is the angle between  $f_i$  and  $f_j$ , and  $\theta_i$  is the dominant local orientation already described. Note that  $\theta_i^j$  can be determined by

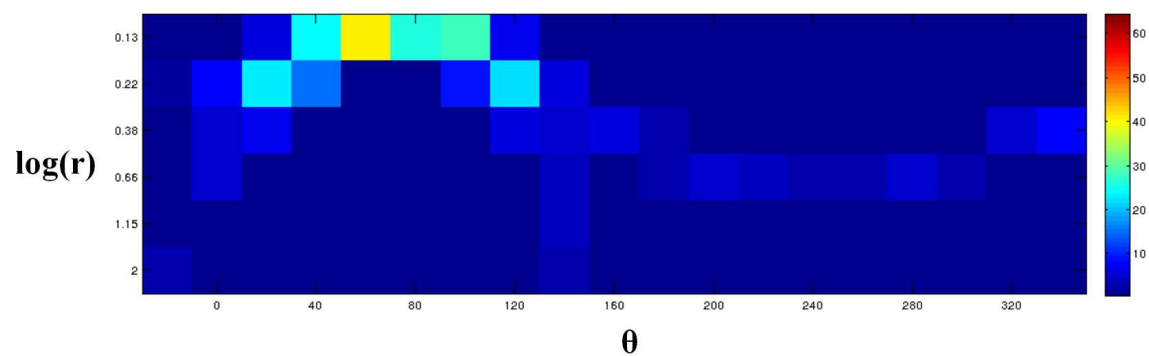
$$\arctan \frac{f_{iy} - f_{jy}}{f_{ix} - f_{jx}}, \quad (3.35)$$

where  $f_{kx}$  and  $f_{ky}$  are the  $x$  and  $y$  components of the  $k^{th}$  location. By normalizing  $r_i^j$  by  $\bar{r}_i$  and subtracting  $\theta_i$  from  $\bar{\theta}_i^j$  we achieve scale invariance and rotation invariance in the LSC descriptor respectively. Each LSC histogram is then represented a normalized  $n_r \times n_\theta$  matrix, which can be flattened to a  $n_r n_\theta$ -dimensional descriptor. Figure 3.40 illustrates the histogram and the distributions of the bins overlaid on a gang graffiti component.

Table 3.10 shows all the parameters/thresholds we used including empirically derived parameters.



(a) LSC Log-Radial Histogram



(b) LSC Normalized Matrix

Fig. 3.40.: Local Shape Descriptor histogram for a specific keypoint and its matrix representation. The matrix holds the count distribution of SIFT keypoint locations relative the specific keypoint.

### 3.7 Content Based Gang Graffiti Image Retrieval

We describe a method to recognize gang graffiti by matching image features from query images against our database of gang graffiti. The method is currently used in two scenarios: 1) “Gang Graffiti Scene Recognition” to recognize scenes containing graffiti (Figure 3.41) and 2) “Gang Graffiti Component Classification” to classify individual graffiti components (Figure 3.42). In both cases we use a vocabulary tree [196] to retrieve input images.

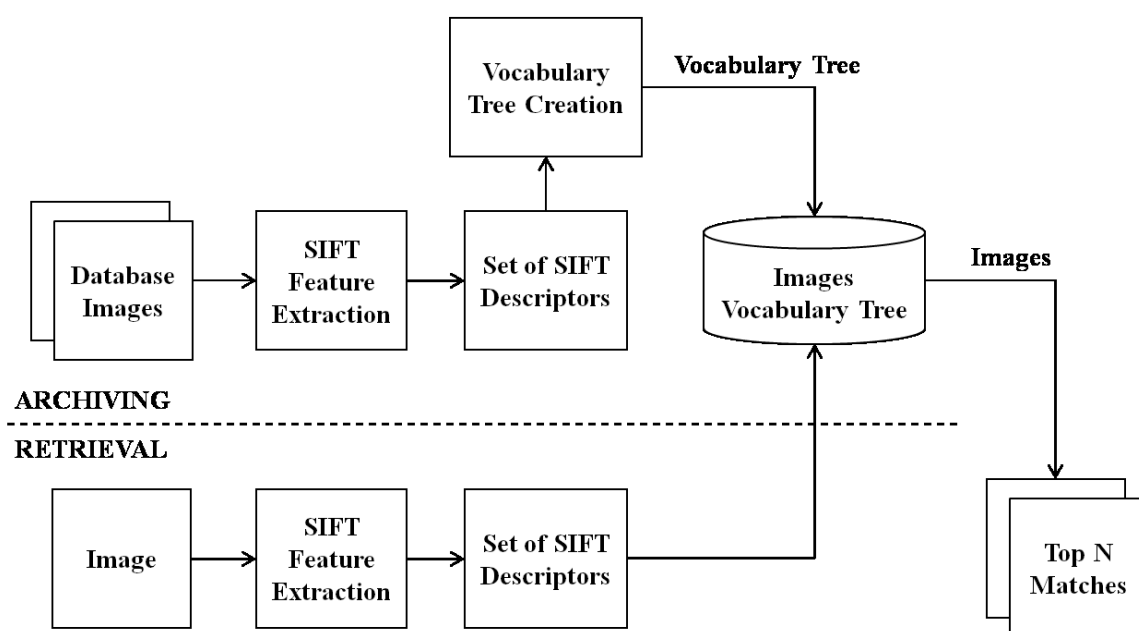


Fig. 3.41.: Gang Graffiti Scene Recognition.

The vocabulary tree is obtained as follows. First, we find features from a set of database images to get  $N$   $D$ -dimensional vectors (i.e. descriptors), where  $D$  will depend on the type of feature [15, 24, 232, 233]. All the  $N$   $D$ -dimensional descriptors populate the  $\mathbb{R}^D$  space, which we then recursively divide into sub-clusters using hierarchical  $k$ -means clustering [234].

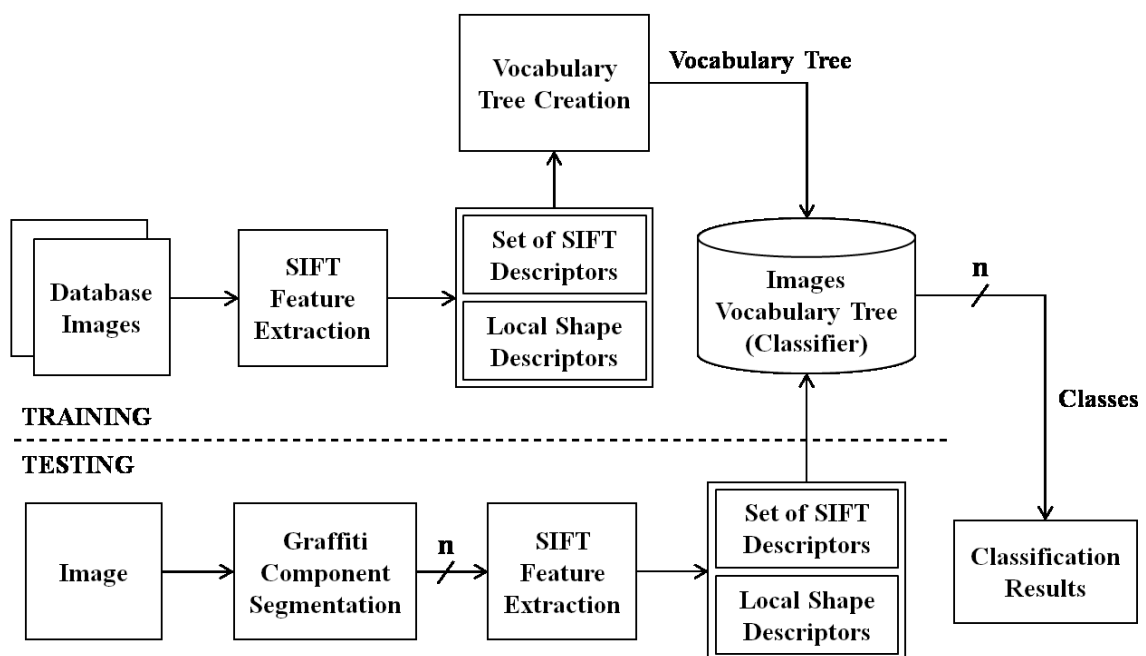


Fig. 3.42.: Gang Graffiti Component Classification.

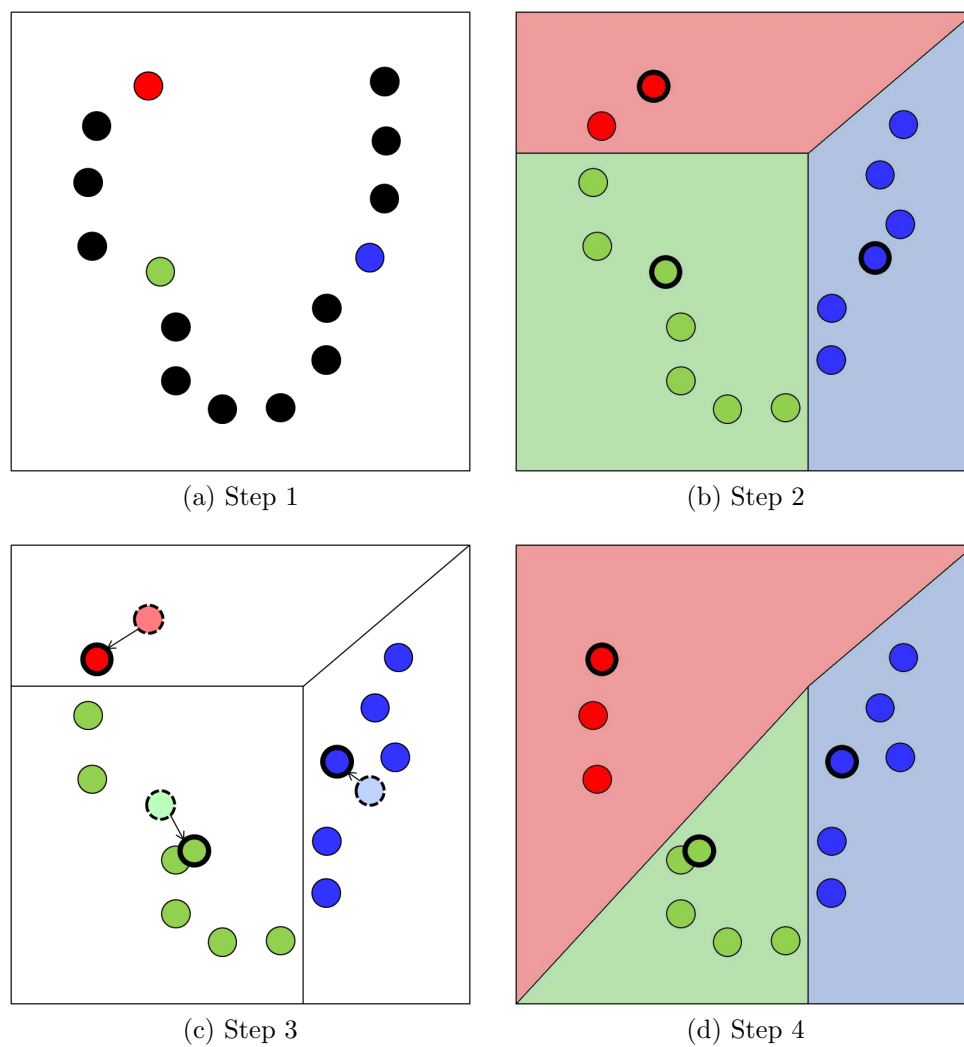
At each recursion level  $k$ -means is used in four steps. First,  $k$  initial “means” are randomly chosen among all the data in the cluster. Second,  $k$  clusters are created by associating every data sample to its nearest mean. Third, each cluster is given a new mean computed as the centroid of all the data points associated with it. Finally, the second and third steps are repeated until convergence is reached (no data sample moves from one cluster to another). Figure 3.43 illustrates the entire process. Since  $k$ -means is greedy for minimizing the sum of squared errors (SSE) it may not converge to the global optimum. Its performance strongly depends on the initial guess of the partition. To escape from getting stuck at a local minimum we can use  $r$  random starts. Specifically, we can repeat the process  $r$  times and select the final clustering with the minimum SSE from the  $r$  runs [235, 236].

We keep clustering until we have a total of  $n_w$  sub-clusters, each of which contains the set of descriptors closest to its center. We call each of these sub-clusters a word. This clustering can be interpreted as a vocabulary tree, where  $k$  corresponds to the branching factor at each level, and each word corresponds to a path from root to leaf. Figure 3.44 illustrates this equivalence. Note that we keep track of the image corresponding to each descriptor.

At the end of the process each image  $i$  can be represented as an  $n_w$  dimensional vector  $d_i$ , where  $n_w$  is the total number of words in the tree. At each index  $j \in [1, n_w]$  in  $d_i$  an entropy weighting [196] is applied so that

$$d_i[j] = \frac{N_j^i \ln \frac{M}{M_j}}{N_i}, \quad (3.36)$$

where  $N_j^i$  is the the number of descriptors of the  $i$ -th database image associated with the  $j$ -th word,  $M$  is the total number of database images,  $M_j$  is the number of database images with at least one descriptor belonging to the  $j$ -th word, and  $N_i$  is the total number of descriptors found on the  $i$ -th image. Based on the results of [196] we chose  $k = 3$  and  $n_w = 10,000$  to create our vocabulary tree.

Fig. 3.43.: Four Main Steps in  $k$ -Means.

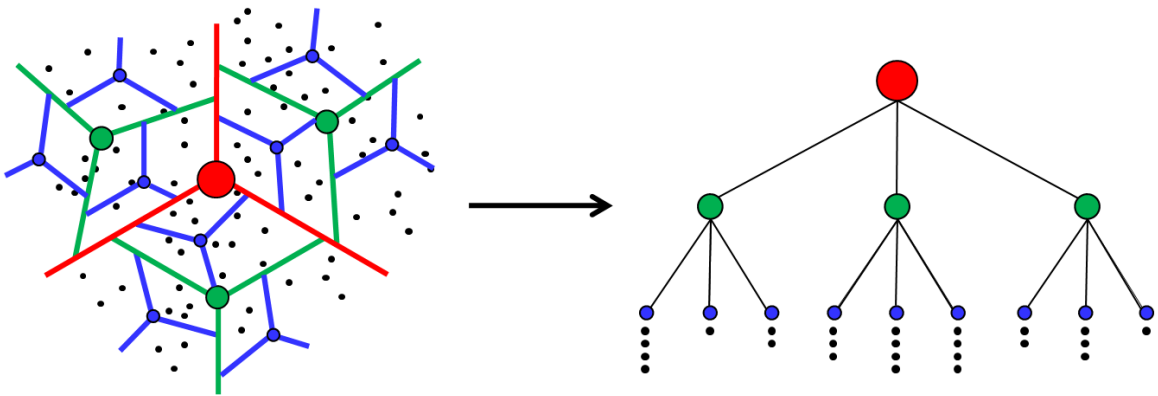


Fig. 3.44.: Vocabulary Tree Built From Hierarchical k-Means. Each black dot corresponds to a descriptor from a database image.



In order to match an input image  $I$  against an image in our database we first extract descriptors from  $I$ . Each of the input descriptors is pushed down the vocabulary tree to find its closest word and an  $n_w$  dimensional vector  $q$  is created following the same criteria explained above, such that

$$q[j] = \frac{N_j^q \ln \frac{M}{M_j}}{N_q}, \quad (3.37)$$

The method in [196] proposes a scoring method to find the closest match to  $I$  based on normalized differences, such that the closest match  $CM$  is

$$CM = \underset{i}{\operatorname{argmin}} \|q - d_i\|_2^2 \quad (3.38)$$

However, in high-dimensional spaces (e.g.  $n_w = 10,000$ ) the Euclidean distance exhibits properties of the phenomenon known as curse of dimensionality [237, 238]. The estimate of  $CM$  can be very poor if “boundary effects” are not taken into account. The boundary effect shows how the query region (i.e. a sphere whose center is the query point) is mainly outside the hyper-cubic data space. One way of illustrating this effect is to compare the volume ratio between a hypersphere with and a hypercube [239, 240]. The volume of a hypersphere with radius  $r$  and dimension  $d$  is

$$V_{hs} = \frac{2r^d \pi^{d/2}}{\Gamma(d/2)}, \quad (3.39)$$

where  $\Gamma()$  is the Gamma function defined as

$$\Gamma(m) = 2 \int_0^\infty e^{-r^2} r^{2m-1} dr. \quad (3.40)$$

The volume of a hypercube with radius  $r$  and dimension  $d$  is

$$V_{hc} = (2r)^d. \quad (3.41)$$

Therefore, it can be seen that

$$\lim_{d \rightarrow \infty} \frac{V_{hs}}{V_{hc}} = \lim_{d \rightarrow \infty} \frac{\pi^{d/2}}{2^{d-1} d \Gamma(d/2)} = 0. \quad (3.42)$$

This shows how nearly all the high-dimensional space is contained in the “corners” of the hypercube.

Note that most average-case analyses of nearest neighbor searching techniques are made under the simplifying assumption that  $d$  is fixed and that the number of descriptors is so large relative to  $d$  that the boundary effects can be ignored. In Gang Graffiti Scene Recognition we find hundreds of high-dimensional descriptors from an input image, so we can use this assumption. However, in Gang Graffiti Component Classification we only extract dozens of high-dimensional descriptors, and making this assumption can be dangerous. Instead, we propose a majority voting matching approach, where  $CM$  is computed as

$$CM = \operatorname{argmax}_i \sum_{j=1}^{n_w} (N_j^q)^i, \quad (3.43)$$

where  $(N_j^q)^i$  is the number of descriptors from  $q$  associated with the  $j$ -th leaf that match the  $i$ -th database image. Figure 3.46 illustrates the majority voting matching approach. Note that a drawback of the basic majority voting classification occurs when the class distribution is skewed. That is, samples of a more frequent class (i.e. graffiti component) tend to dominate the prediction of the query [241]. Therefore we need to make sure that the training data for Gang Graffiti Component Classification contains the same number of samples for each class.

The main advantage of using a vocabulary tree for image retrieval is that its leaves define the quantization, thus making the comparison dramatically less expensive than

Table 3.11: Parameters and thresholds used in Content Based Gang Graffiti Image Retrieval.

Parameter	Description	Value
$k$	Branching factor	3
$n_w$	Number of leaves	10,000

previous methods in the literature [196, 242, 243]. Also, once the vocabulary tree is built, new images can be added by just pushing down its descriptors.

The scalability of the vocabulary tree can be inferred from the results of [196], shown in Figure 3.45. The retrieval performance increases significantly with the number of leaf nodes, the branch factor, and the amount of training data.

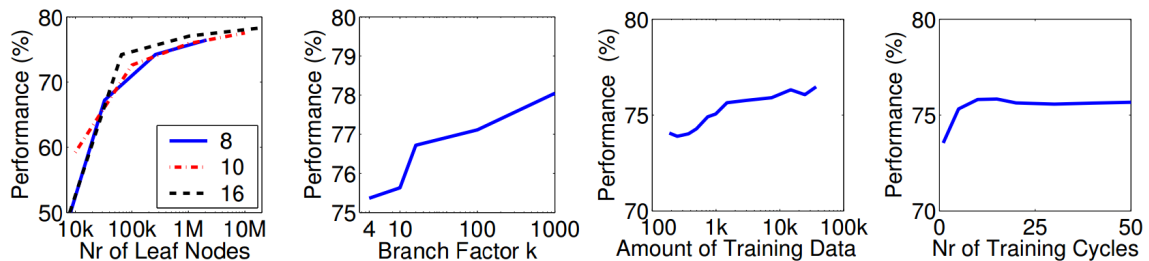


Fig. 3.45.: Scalability Results of Vocabulary Tree tested on a 6,376 ground-truth image dataset [196]. From left to right: Performance vs number of leaf nodes with branch factor  $k = 8, 10$  and  $16$ . Performance vs  $k$  for one million leaves. Performance vs training data volume in  $720 \times 480$  frames, run with 20 training cycles and  $k = 10$ . Performance vs number of training cycles run on 7K frames of training data and  $k = 10$ . The image belongs to [196].

Currently, SIFT features are used for both Gang Graffiti Scene Recognition and Gang Graffiti Component Classification. However, note that the  $k$ -means clustering approach accepts any type multi-dimensional vector.

Table 3.11 shows all the parameters/thresholds we used including empirically derived parameters.

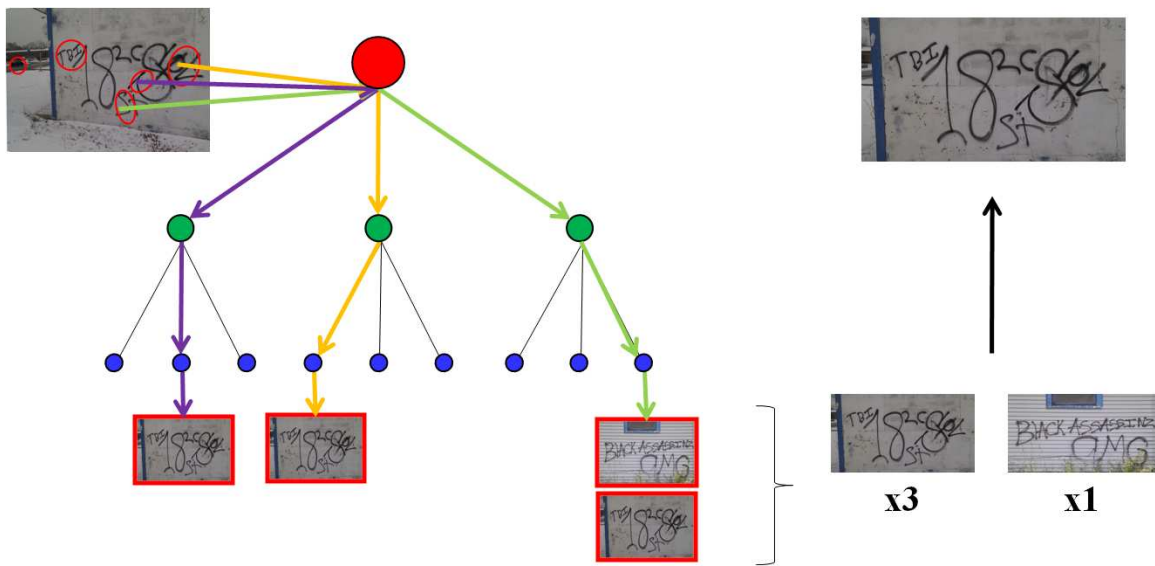


Fig. 3.46.: Majority Voting Matching.

## 3.8 System Implementation

### 3.8.1 System Architecture

We implemented the “mobile” part of the GARI system as an application for Android and iOS devices. We also have a web-based interface accessible from any web browser. Figure 3.47 illustrates the GARI system, which is divided in two groups:

1. **Client-side:** Implemented operations on the mobile device and communicate with the database (server) of gang graffiti through either WiFi or 4G/3G networks.
2. **Server-side:** Implemented operations on the database of gang graffiti and communicate with the client.

The client-side includes the device and methods available to the users, either to operate without the use of a network connection (offline services) or to make queries to the database (online services). The offline services are only available from Android devices (Section 3.8.3). The online services are available from both Android devices or any web browser (e.g., Internet Explorer, Mozilla Firefox, Google Chrome). This includes desktop and laptop computers as well as Blackberry smartphones (Section 3.8.4). The server-side includes all operations done on the server, including image analysis and queries to the database from both the Android application and the web-based interface. The database comprises gang graffiti images and metadata information for each entry, such as EXIF data, image geolocation and the results of the image analysis on each image whether it was done on the server or client.

### 3.8.2 GARI Databases

In this section we describe how the image database is organized. We will first describe the database schema and then show by an example how the information

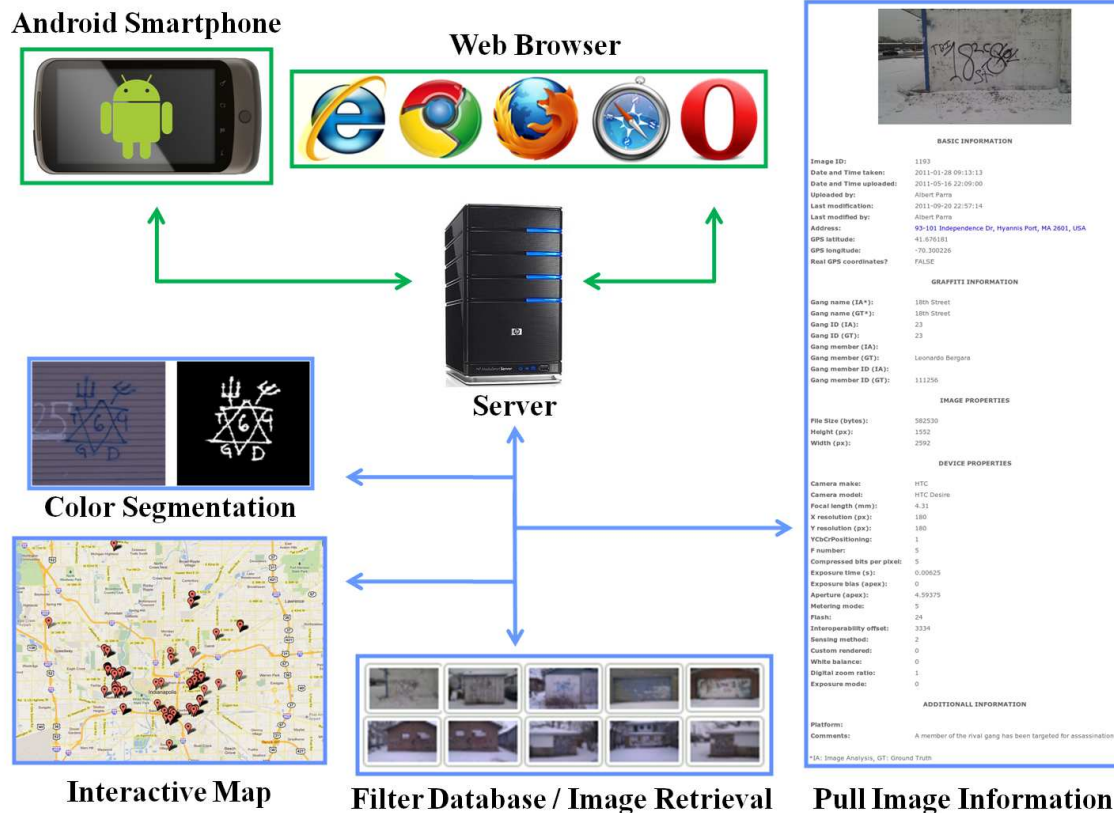


Fig. 3.47.: Overview of The GARI System - Client-Side Components (green) and Server-Side Components (blue).

GARI acquires is added to the database. The database of gang graffiti was deployed for three uses:

1. To collect and organize graffiti images acquired by first responders. This includes the images, metadata, and any interpretation or other information provided by the first responder.
2. To store the results of the image analysis.
3. To manage first responders' credentials, allowing them to access the services available through the Android/iOS applications and the web based interface.

Our database is implemented in PostgreSQL [244] on a Linux server. It consists of eight tables structured as shown in Figure 3.48. Note that the schema does not show all the fields in all the tables but just the relevant fields to indicate the association between the tables. Also the various IDs mentioned below (e.g. image ID) will be discussed in more detail after the tables are described in the following list.

1. **images**: Stores EXIF data from the images along with image location and general image information and the results from the image analysis. The fields related to this table are shown in Tables D.1, D.2, D.3 and D.4 in Appendix D.
2. **imageColors**: Stores all color IDs related to each image ID. This table is especially useful when more than one color is found in the same graffiti image.
3. **colors**: stores the relationship between color IDs and color names.
4. **imageBlobs**: Stores the number of blobs in each graffiti, the ID of each graffiti component for each blob, and the color ID of each graffiti component. This also stores special attributes of graffiti components. These attributes may include a specific graffiti component being crossed-out, upside-down, etc. Table D.6 in Appendix D describes the fields of this table.
5. **blobComponents**: stores the relationship between graffiti component IDs and graffiti component names, as well as the type ID for each graffiti component. Each graffiti component belongs to any of the following types: symbol, character, number, acronym, nickname, string.
6. **componentTypes**: stores the relationship between type IDs and type names.
7. **gangComponents**: stores the relationship between gang IDs and gang names, as well as the graffiti component ID (or multiple graffiti component IDs) associated with each gang. This table is especially useful when more than one graffiti component is associated with the same gang name.

8. **users**: Stores users' credentials to access to the system services as well as information concerning administrative privileges, email addresses, and registration and login status. Table D.5 in Appendix D describes the fields of this table.

Note that currently we only populate the tables **images** and **users**. The database relationships between all the tables are implemented and are ready to be used in the future (see Section 6).

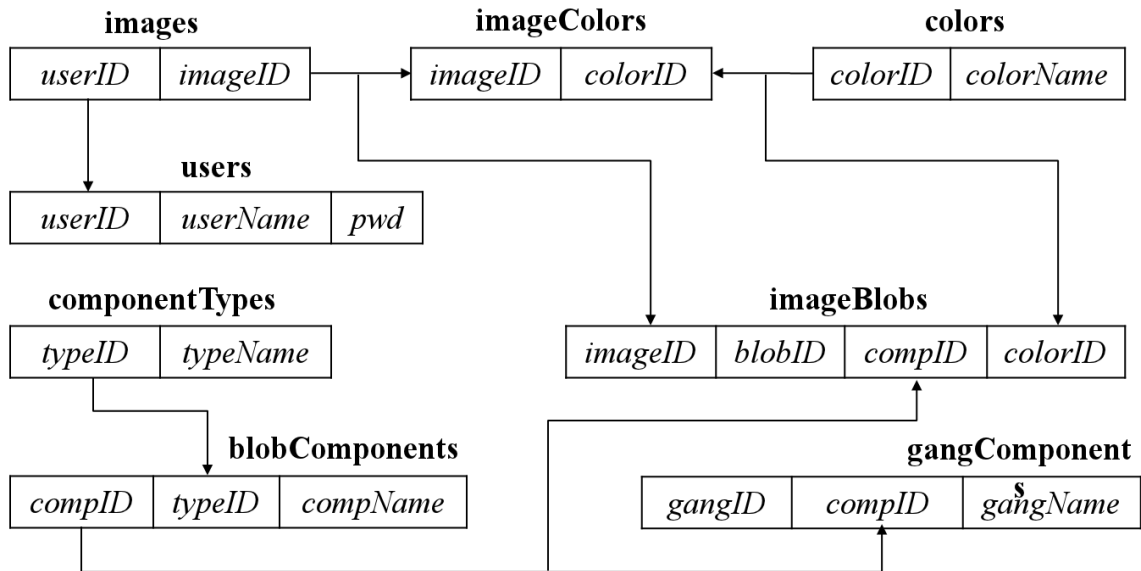


Fig. 3.48.: Database Schema Showing The Associations Between the Tables in the Database.



## Adding Images to the Database

The following example illustrates the process of adding a graffiti image to the database. The image analysis is assumed to have been completed. Figure 3.49 shows the example image that has been manually labeled to facilitate the explanation. Each labeled circle represents a blob and each blob contains a distinguishable graffiti component. The blob labeling of the image corresponds with the field *blobID* from table *imageBlobs* in the database.

First, we fill table *imageColors* with the colors found in the graffiti. This is, black, green, and blue. Second, we analyze the blobs separately:

1. Color: black. Graffiti component: X3.
2. Color: green. Graffiti component: SPV.
3. Color: blue. Graffiti component: X3.
4. Color: blue. Graffiti component: LK. Crossed-out in green.
5. Color: blue. Graffiti component: ES. Crossed-out in green.

Note that the meaning of the acronyms and the type of the graffiti components is not addressed here. This information is assumed to already exist in the database.

Once the image analysis is complete the image, along with the blob information, is added to the database. Figure 3.50 shows the database fields filled with the information obtained from the graffiti in Figure 3.49. First, the user ID of the first responder who captured the image and the image ID are added to the *images* table. The image ID is a unique identifier of the graffiti image and it is automatically updated every time an image is uploaded to the server. Although it is not shown in Figure 3.50, some additional image information (i.e., EXIF data, GPS coordinates) is extracted from the uploaded image and added to the *images* table. Second, the color IDs for the three colors found in the graffiti, which are obtained by checking the color description field, (labeled *colorName* in Figure 3.50), are added to the *imageColors* table, and

linked to the graffiti ID. At the same time, the five blobs are added to the *imageBlobs* table. Each blob has a corresponding graffiti component ID, which is obtained by checking the graffiti component description field, (labeled *compName* in Figure 3.50), of the *blobComponents* table. Each graffiti component has a color associated with it and can activate one or many attributes in the same table (see Table D.6 for all the attributes). In this example, blobs one to three do not have any additional attribute. Blobs four and five have activated the crossed-out attribute.

Note that this process is totally objective. That is, the information uploaded to the database does not require any interpretation from the first responder. With all the objective information available in the tables and the associations between the data one can produce an informed graffiti interpretation. For example, we have added graffiti components with IDs 27 (*SPV*) and 29 (*LK*). These IDs are associated with specific gang names in the *gangComponents* table. The same reasoning could be used if the graffiti did not contain any specific content with just the graffiti color being identified. Additional tables can relate gang IDs with color IDs effectively providing the results of gangs matching the specific color or colors.

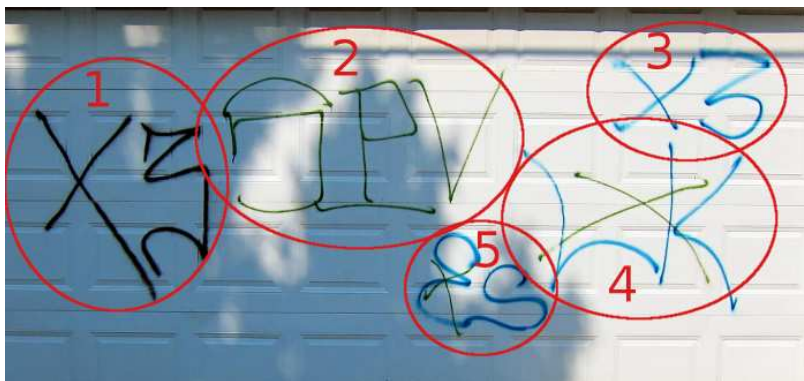


Fig. 3.49.: Example of Graffiti (Manually Labeled).

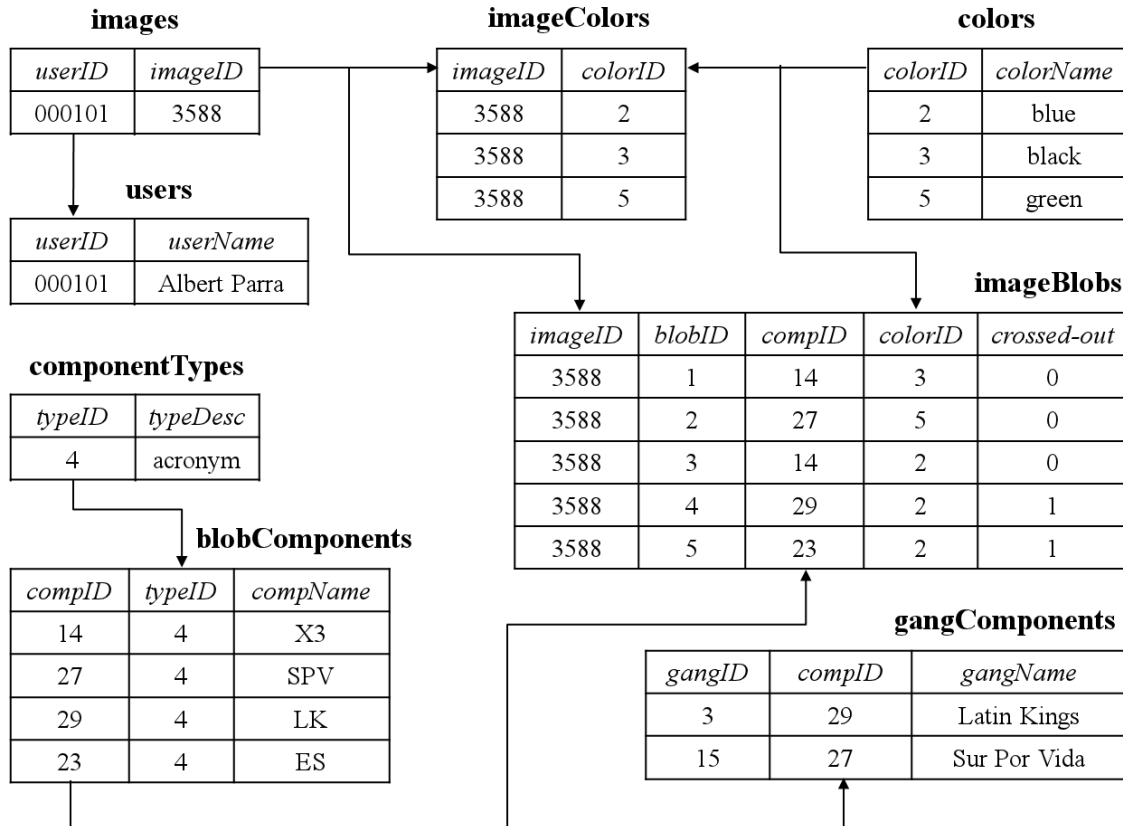


Fig. 3.50.: Database Fields With Information From The Graffiti in Figure 3.49.

### 3.8.3 Android/iOS Implementation

We implemented the GARI system on Android and iOS devices as summarized in Figure 3.51. We called this application Mobile GARI. In this section we describe how the application works and describe its user interface.

#### Overview

A user takes an image of the gang graffiti using the embedded camera on the device via the Graphical User Interface (GUI). The EXIF data of the image, including GPS location and date and time of capture, is automatically added to the image header.

The user can then choose to upload the image to the server to be included in the database of gang graffiti, find similar images in the database of gang graffiti, or do color recognition. The first option, uploading to the server, allows the user to send the image and the EXIF data to the server creating a new entry in the database. The second option, find similar images, allows the user to send the image to the server and find gang graffiti images that match part or all of the contents of the image. The third option, color recognition, allows the user to trace a path in the current image using the device's touchscreen. The color in the path is then automatically detected (Section 3.4) and the result is shown to the user. The database of gang graffiti can then be queried to retrieve graffiti images of the same color.

Another option is to browse the database of gang graffiti given various parameters such as the distance from current location or date and time. The thumbnail images that match the query are downloaded from the server and shown to the user on the mobile telephone. The user can then browse the results to obtain more information about the specific graffiti. Note that in order to browse the database of gang graffiti a network connection is required.

We implemented the system on different smartphones makes and models, but always targeting version 3.2 of the Android operating system (OS). We chose Android OS version 3.2 to cover as much user market as possible while still being able to include the necessary features. Since Android applications are generally forward-compatible with new versions of the Android platform, by choosing OS version 3.2 we cover 78.7% of the market (as of March 2014) [245].

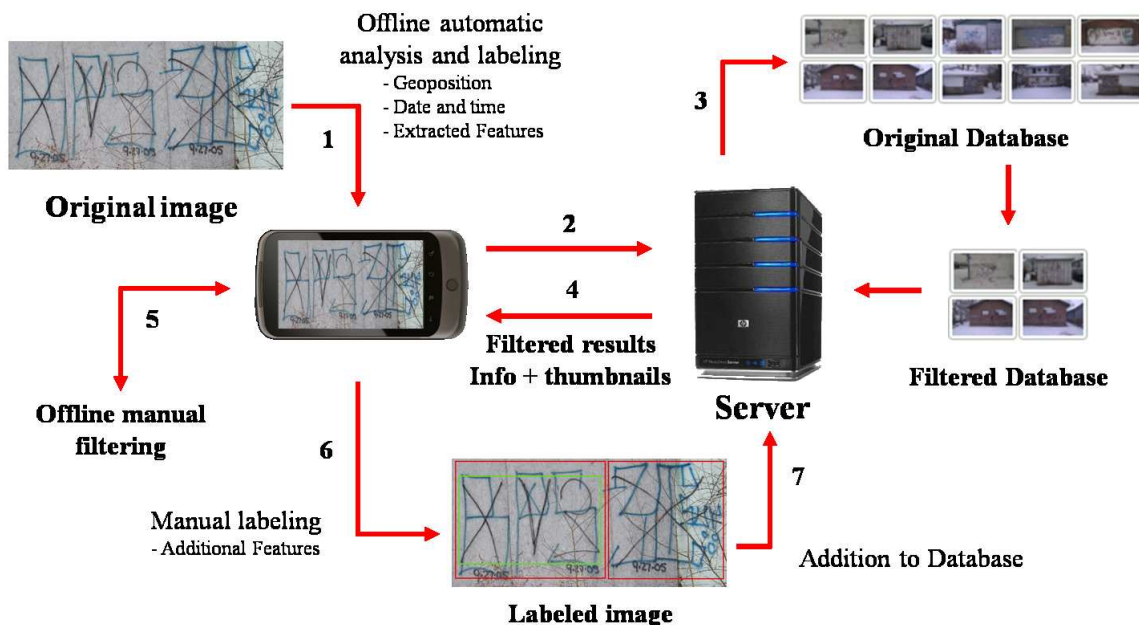


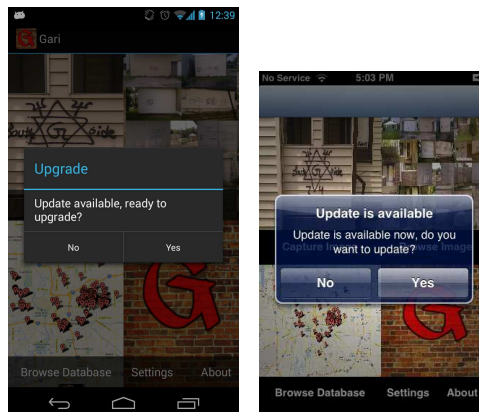
Fig. 3.51.: Overview of the GARI System.

## User Interface

Our Android application does not require the use of a network connection. However it is mandatory if the user wants to browse the graffiti database or upload images to the graffiti database. The application automatically checks for updates when launched, notifying the user if a new version is available (Figure 4.25). A user must be assigned a User ID (equivalent to a First Responder ID) and a unique password in order to use GARI. Once the User ID and password has been entered, the main screen is presented. The menu options are displayed on the main screen (Figure 4.26a/3.53c) and on the secondary screen (Figure 4.26b/4.26b) when an image is captured or browsed. In Android devices, the menu button brings additional options when available. Note that the menu button can be a hardware key (Figure 3.54a) or a software key (Figure 3.54b) depending on the device used. In iOS devices, the

additional options are presented on the screen as buttons. The main screen includes the following options:

- Browse Image
- Browse Database
- Capture Image
- Send to Server (available after browsing or capturing an image)
- Analyze Image (available after browsing or capturing an image)
- Settings
- About



(a) Android

(b) iPhone

Fig. 3.52.: Automatic updates.

### Browse Image

The user has the option to browse images stored on the Android device, to later upload them to the server or analyze them. Note that the entire phone image gallery is browsed, including images that have not been taken using the Mobile GARI application. When the option “Browse Image” is tapped, a directory browsing window is

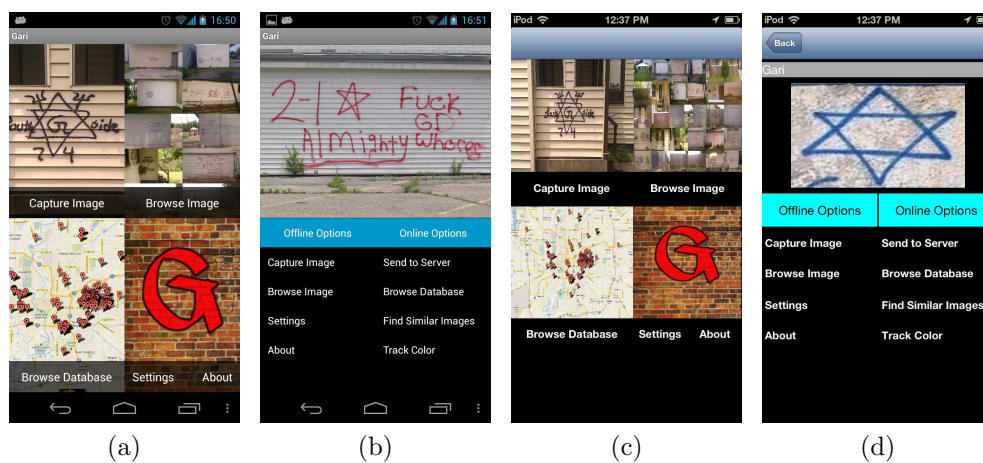


Fig. 3.53.: User options screens for Android (4.26a, 4.26b) and iPhone (3.53c, 3.53d).

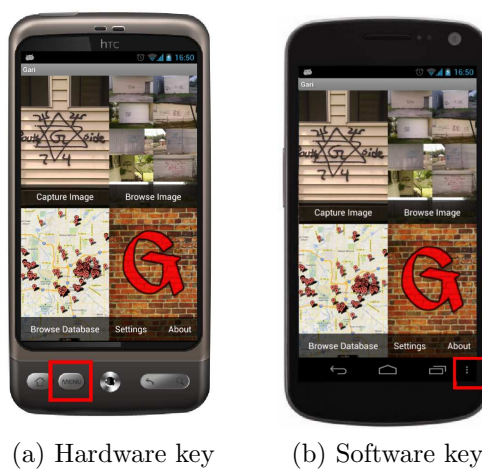


Fig. 3.54.: Examples of location of the menu button (red square) on Android devices.

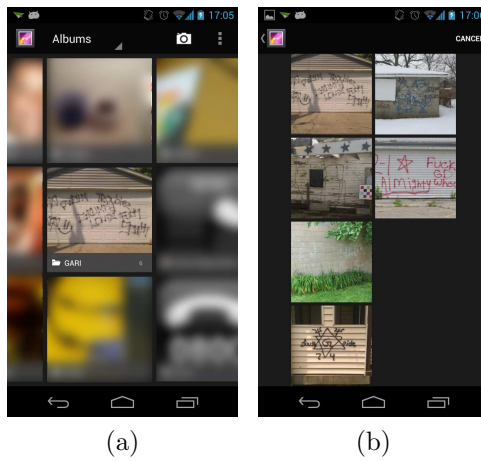


Fig. 3.55.: Example of image browsing.



opened, and the user can search and select the desired image. Figure 3.62 shows an example of browsing.

### Browse Database

The menu option “Browse Database” allows the user to browse the database by radius. That is, it extracts from the database all the images in a given radius from the current location. Figure 3.56 shows the dialog where the user can select a radius between 1 mile and 20 miles.

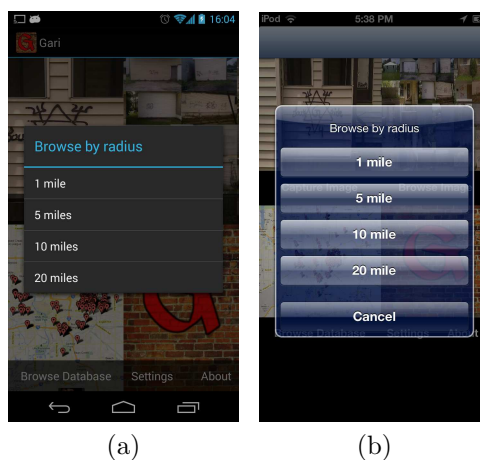


Fig. 3.56.: Browse by radius screen for Android (left) and iPhone (right).

When a specific radius is chosen, the application checks for the device location automatically, in order to add the GPS coordinates to the image. Depending on the system used (Network (3G/4G or WiFi) or GPS), it can take up to 30 seconds to acquire the location. The user is notified during the period, as shown in Figure 3.57.

In Android devices, if the location system is not enabled on the device, the user is notified and taken to the location settings (Figure 3.58), where the location systems can be enabled.

Once the location is locked, the application contacts the image database and checks how many thumbnails have to be downloaded (Figure 3.59a/3.59c). If the user accepts, the information that matches the query is retrieved (Figure 3.59b/3.59d). Figure 3.60 shows an example of the results, where each line contains a thumbnail of

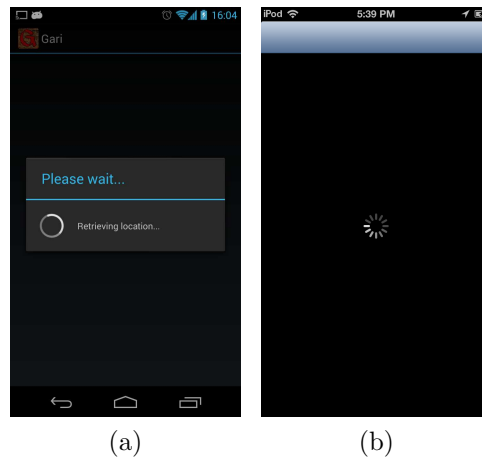


Fig. 3.57.: Progress dialog notifying the user of a location retrieval, for Android (left) and iPhone (right).

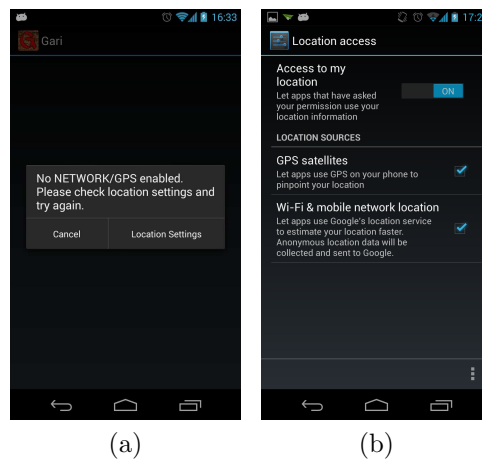


Fig. 3.58.: 3.58a Dialog notifying the user that no Network or GPS systems are enabled, and 3.58b location settings of the device, for Android.

a graffiti or tattoo and basic information about it, including the date and time the image was taken, and its GPS latitude and longitude.

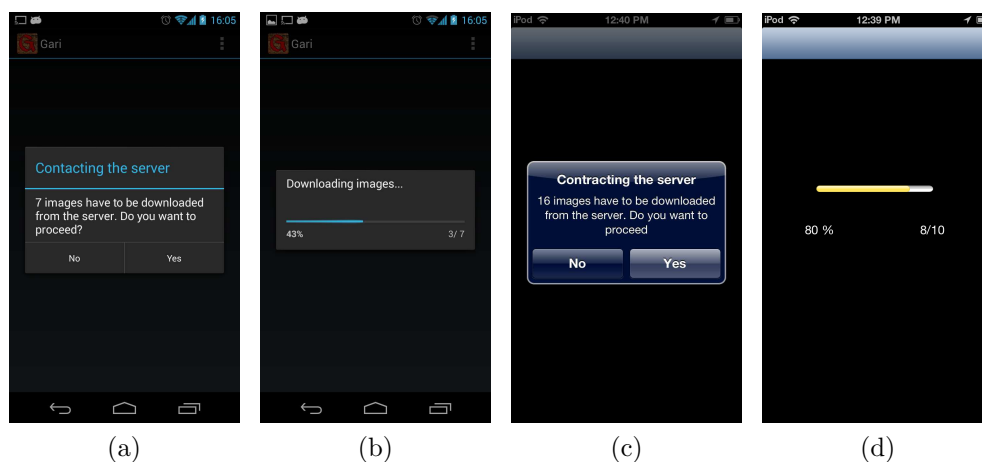


Fig. 3.59.: Screen notifications during database browsing for Anroid (3.59a, 3.59b) and iPhone (3.59c, 3.59d).

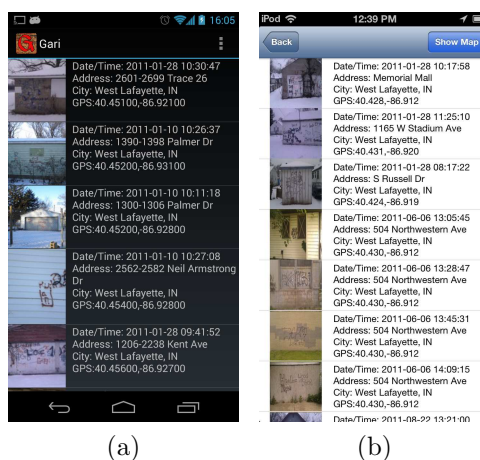


Fig. 3.60.: Results after querying the image database for Android (left)) and iPhone (right).

To obtain more information about a particular graffiti or tattoo, the user can tap on either the thumbnail or the text field, and the application will contact the server, extracting a larger image and the information available. Figure 3.61 shows an example of the extended results. The text field includes information about the

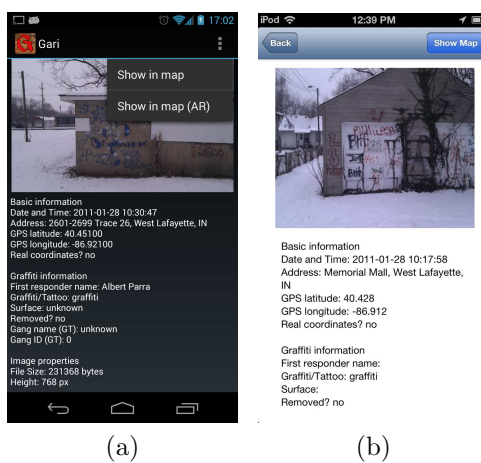


Fig. 3.61.: Extended results after querying the image database for Android (left) and iPhone (right).

graffiti or tattoo taken from the database tables. The available fields are detailed in Appendix D.

Whether the user is in the results view or in the extended results view, the menu key will have the option “Show in map.” It allows the user to display the position of multiple graffiti or tattoo or focus on a single image (green marker on Figure 3.62a/3.62b), depending on the current layout. In Android phones the user can choose to display his/her current location via the “My Location” option, and switch between normal and hybrid maps via the “Hybrid” option. In iOS devices the user can switch between normal and hybrid using the buttons on the map.

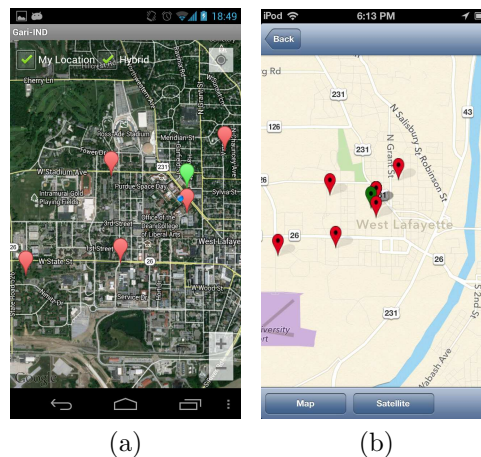


Fig. 3.62.: Graffiti locations displayed on a map for Android (left) and iPhone (right)

Similar to the “Show in map” option, the menu key will have to option “Show in map (AR).” AR stands for Augmented Reality. It allows the user to display the position of graffiti and tattoo locations on top of the camera feed on the mobile phone. Figure 3.63 shows an example. As the user moves the mobile phone around, the screen gets updated and shows graffiti and tattoo locations in the camera range as pins. When tapping on a pin, a dialog appears at the bottom displaying the address, city and distance of the graffiti/tattoo from the mobile phone. Also, the image thumbnail is shown in the bottom right. When tapped, the user is redirected to the extended results page (Figure 3.61).

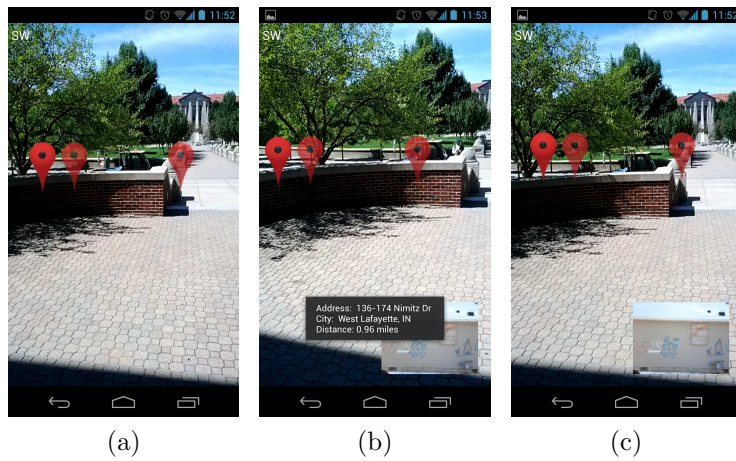


Fig. 3.63.: Graffiti locations displayed on an Augmented Reality feed for Android

### Capture Image

The menu option “Capture Image” starts the image acquisition. The user just has to point to the graffiti or tattoo and wait for the three second countdown followed by automatic image acquisition. The countdown is shown in the center of the screen, as illustrated in Figure 3.64. The countdown is automatically restarted if the smartphone registers a considerable amount of shaking, in order to minimize the risk of taking blurred images. After the image is automatically captured the application checks for motion blur and lack of illumination, and restarts the counter to take a new image if necessary. The application automatically checks the user’s current location after acquiring an image.



Fig. 3.64.: Camera Activity.

### Send to Server

The menu option “Send to Server” allows the user to send the current image to the server. First, the user will be prompted to select the source of the image, either graffiti or tattoo (Figures 3.65a/3.65c and 3.65b/3.65d). After tapping on “Send” the image is uploaded to the server on the background. While an image is being uploaded, the user can keep using the application and send more images. A queue will be automatically created and the images will be sent sequentially. If the Internet connection is lost, the application will wait until the connectivity is restored to restart the uploading process. If the application is closed or the mobile device is shut down during an upload, the file will be automatically uploaded next time the user launches the application. Figure 3.66 illustrates the process. An icon on the notification bar (top of the screen) shows the status of the upload. By dragging down the notification

bar the user can see more information about the upload progress. If the image is successfully added to the database, the application will also extract the information uploaded, and will display it to the user (Figure 3.67).

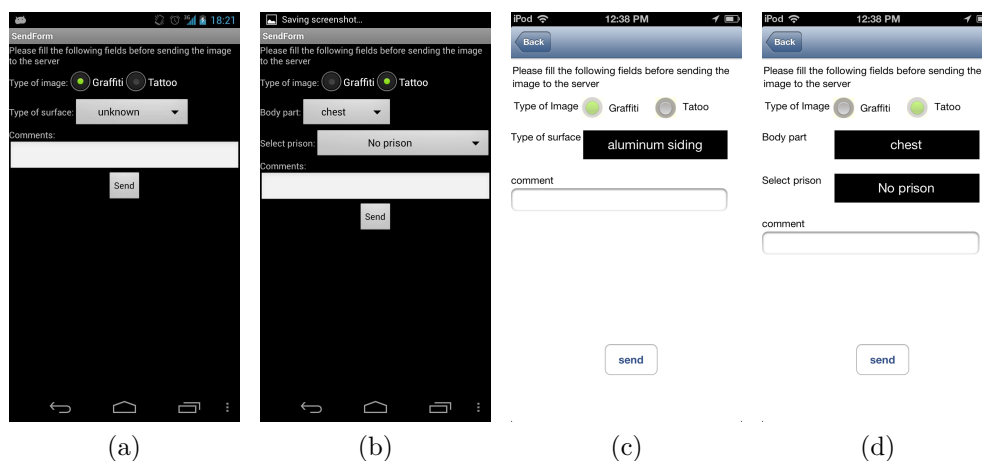


Fig. 3.65.: Result of uploading an image to the server for Android (3.65a and 3.65b) and iPhone (3.65c and 3.65d).

### Find Similar Images

The menu option “Find Similar Images” allows the user to find similar images to the current image being displayed on the secondary screen. The image is sent to the server and analyzed. When the analysis is done, the server sends back a list of matching candidates. Figure 3.68 shows the process. The options for this list are the same as the ones described for the results from browsing the database. Note that the matching candidates in the list are sorted by score, where the first entry corresponds to the most similar image to the query.

### Analyze Image

The menu option “Analyze Image” allows the user to aid the application in detecting the gang graffiti components. This option is only enabled once an image has been captured or browsed. First, the user has to select a region of the image containing the graffiti color, as shown in Figure 3.69a/3.69c. When the desired area is selected and “Save” is tapped, the user can create a path on the image using their finger, as shown



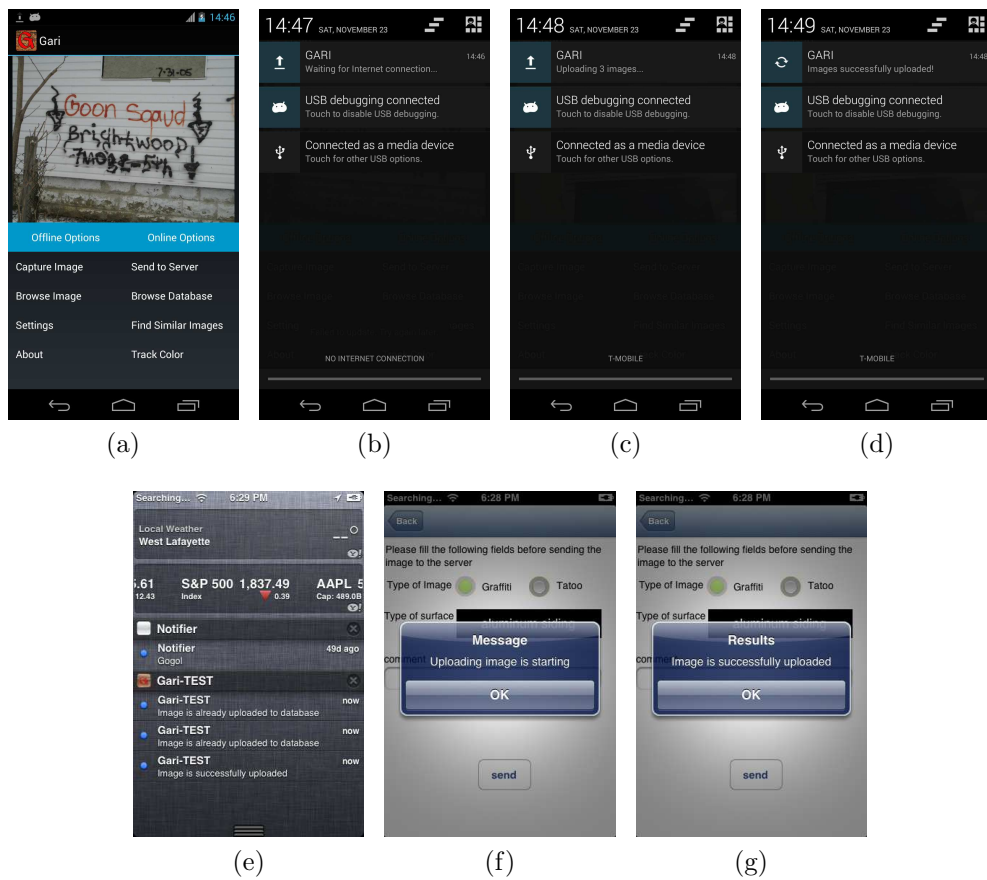


Fig. 3.66.: Image uploading on the background on Android (top) and iPhone (bottom). From left to right (Android): Uploading image (icon), waiting for Internet connection, uploading 3 images, image successfully uploaded. From left to right (iPhone): Messages on the notification bar, Uploading image (message), image successfully uploaded (message).

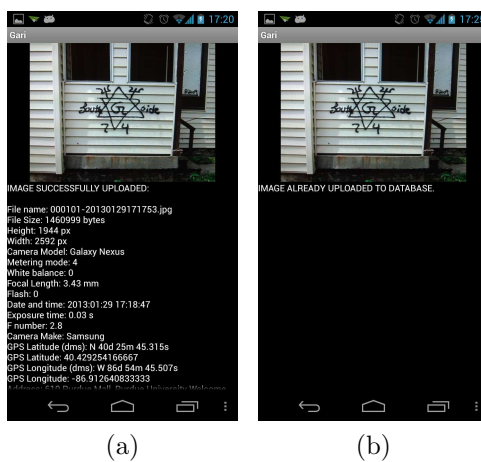


Fig. 3.67.: Image upload successfully (3.67a) and image already uploaded to database (3.67b).

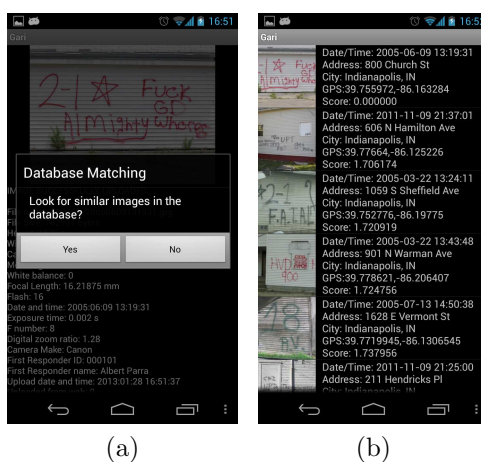


Fig. 3.68.: Screen notifications when finding similar images (Android).

in Figure 3.69b/3.69d. There is no need to trace the entire content of the area with the same color. Just a significant sample is enough to determine the color. Figure 3.69b/3.69d also shows the available options. In Android devices the “Undo” option removes the last path created; the “Clear” option clears all the paths created; and the “Analyze” option obtains the current path and analyzes the color. In iOS devices the “Analysis” option obtains the current path and analyzes the color. The image and the recognized color are then sent to the server for analysis, and the results are given back to the user as a list of thumbnails, classification results and gang graffiti colors, as shown in Figure 3.70.

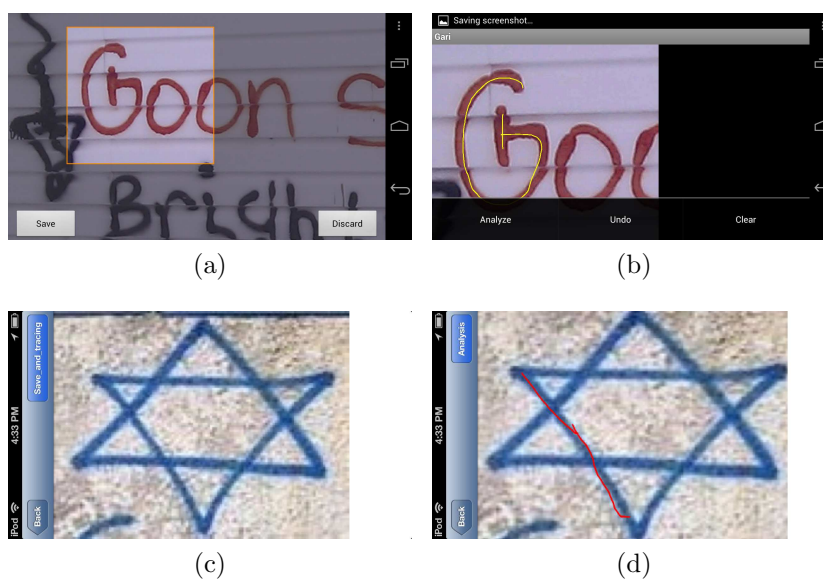


Fig. 3.69.: Steps to follow when selecting the region to analyze the color for Android (top) and iPhone (bottom).

Figure 3.71 shows the result of the color tracing. The application then extracts from the database all the gangs that match the detected color. There is also the option “Browse database by color”, which queries the database and extracts all the images in the database that match the traced color. Figure 3.71b shows an example. After color recognition the user can send the image to the server for automatic graffiti analysis.

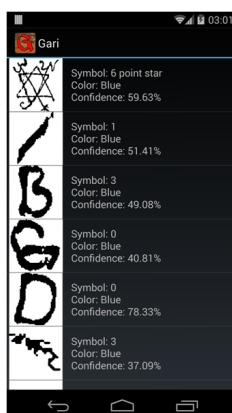


Fig. 3.70.: Image Analysis Results.

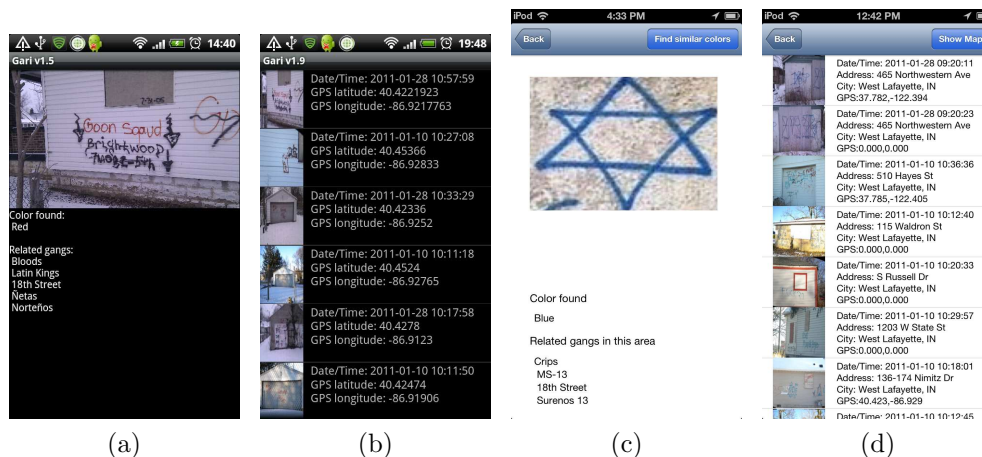


Fig. 3.71.: Gangs related to the traced color and images in the database that match the traced color for Android (3.71a, 3.71b) and iPhone (3.71c, 3.71d).

## Security

Our Android application is used by first responders from multiple agencies. Therefore, it is mandatory to ensure that only authorized users can access and use the application. The connections to the server must be secure and all the information transmitted to and from the server must be encrypted (using the SSL/TLS protocol). The user credentials are sent every time the application contacts the server to make sure the connection is made by an authorized user. In the Android version we use ProGuard [246], a code optimizer and obfuscator for the Android SDK. It reduces the application size up to 70% and makes the source code more difficult to reverse engineer. It also improves the battery life by removing verbose logging code in a background service. An additional level of security includes the creation of two types of users:

- Regular users: Can switch between users, change their password, delete specific images only taken by themselves, and send crashlogs to the server.

- Administrative users: Can modify the server domain name/IP address, change user IDs, change passwords, delete specific images from any user, delete all images of any specific user, and send crashlogs to the server.

When launching the GARI application a dialog box automatically prompts the user for login credentials (Figure 4.33). The user is required to input a user ID and a password.

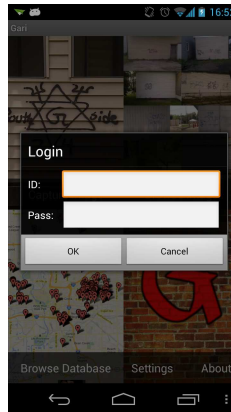


Fig. 3.72.: User ID Prompt.

The first time a user logs in the credentials are checked with the server and once they are validated they are stored in the device in an encrypted file. This allows the user to use the application without needing a network connection. Note that passwords are never stored as plaintext, neither on the device or the server. They are hashed using an MD5 cryptographic hash function [247]. We also use a login system in which the application creates a session for an authorized user that lasts 24 hours. After that period of time the user is required to login again.

All authorized users can access the “Settings” option from the main screen of the application. Figure 3.73 shows the various options. Note that no one can delete images from the server. At this time no one can edit the attributes of images retrieved from the server.

- Server domain/IP: the the address of the server to be changed by domain name or IP address (only available to administrative users).

- Switch user: allows one to open sessions for other users. Note that switching to another user ends the session for the current user.
- Change password: allows one to change the password used to access the application. Note that the password is changed for both the Android application and the web-based application.
- Send crashlog: allows system crash feedback to be sent to the server.

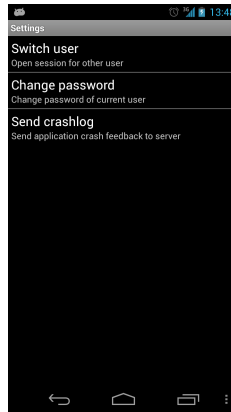


Fig. 3.73.: “Settings” Dialog, Showing the Various Options.

### 3.8.4 Web Interface

#### System Overview

We also implemented our system as a web interface that gives a user access to the graffiti in the database and provides the ability to upload, modify and browse most database contents as summarized in Figure 3.74. We called this application Desktop GARI. The user logs in into the “Archive” using authorized credentials. Note that the credentials are the same for both the Android application and the web services. The user can then either browse the database of gang graffiti or upload an image. If the choice is to browse the database, the user can check the graffiti images and their attributes or filter the database using parameters such as radius from a specific

location or address, capture data, upload data, or modified date. The results are shown as a list of thumbnail images with basic information that identifies the graffiti image. The user can then browse specific images and place them on a map, so to visually track gang activity. If the choice is to upload an image, the user can select a graffiti image from their local system (i.e., any device with a web browser). Some attributes can be adjusted through guided steps before adding the information to the database, such as location, gang information, or additional comments.

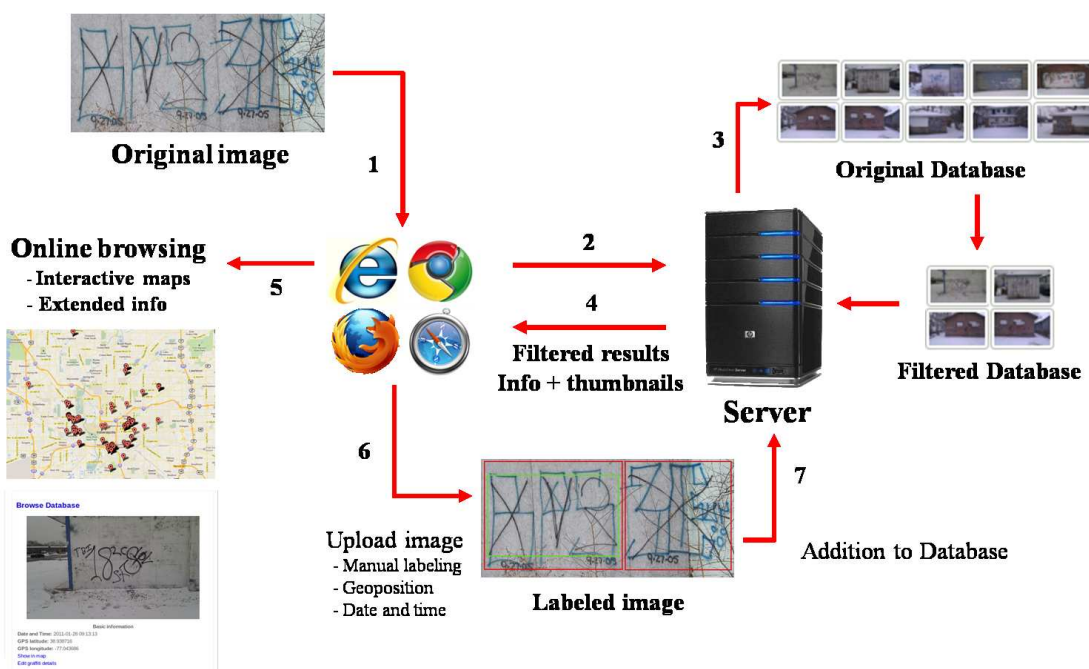


Fig. 3.74.: Overview of the Web Interface of the GARI System.

The web interface is available from any device with a web browser. This includes all desktop and laptop machines and all mobile telephones capable of browsing the web (e.g., iOS, Blackberry, Android devices). In some cases, the current location of the user is required in order to retrieve results from the database of gang graffiti such as when using the “radius” function to display graffiti on a map. Geolocation was introduced with HTML5 and it is widely implemented by many modern browsers.



However, only the latest browsers support this service. Table 3.12 lists the browsers and their support level for Geolocation.

Table 3.12: Web Browsers Supporting HTML5 Geolocation Service.

<b>Browser</b>	<b>Version</b>
Firefox	3.5+
Internet Explorer	9+
Google Chrome	5+
Safari	5+
iPhone Safari	+3.0 OS
Android	Through Gears API
Opera	10.6+

## User Interface

As of March 2014 the GARI website is located at [www.gang-graffiti.org](http://www.gang-graffiti.org). The main page contains information about the GARI project, its principal investigators, and the graduate students involved. Figure 3.75 shows a snapshot.

The “Archive” page (Figure 3.76) displays the options available a user. These include:

- Browse database
- Upload image
- Upload multiple images
- Create database report

A username and password is required to access the database contents. A user can use the same username and password used for the mobile application.

### Browse database

The “Browse database” page (Figure 3.77) allows the user to either browse the entire database or to do a specific search. This includes:

- **Browse all database/graffiti/tattoo:** Retrieves from the database either images, only graffiti images, or tattoo images.
- **Search by radius:** Retrieves from the database all the graffiti and tattoos in a specific radius, from a specified location from the list. The locations in the list include the user’s current location, the Video and Image Processing Laboratory (VIPER) at Purdue University, and the Indianapolis Metropolitan Police Department (IMPD). The “Current location” option requires the user to share their current location, as shown in Figure 3.78.
- **Search by Date:** Retrieves from the database the graffiti and tattoo images captured, uploaded or modified in a specific period of time.
- **Search by address:** Retrieves from the database the graffiti and tattoo images in a specific radius, from a specified address. Provides more flexibility than the “Search by radius” option.

The search results are shown in Figure 3.79. At first, only a small-scale image and basic information is displayed. Depending on the search various parameters are shown, including:

- **Date/Time captured (uploaded, modified):** date and time the image was acquired, uploaded or modified, depending on the search.
- **Address:** address where the image was acquired. A map showing the graffiti or tattoo location when clicked is available.
- **More information:** link to show additional information about the graffiti or tattoo.

- Image ID: image identifier in the database.
- Distance: distance from the user’s current location to the graffiti or tattoo. Only available when searching by radius or address.

Each image or group of images can be displayed on an interactive map. Figure 3.80 shows an example of the interactive map when a single image is displayed. The image is placed on a map, and a balloon pops out, showing a thumbnail and some information about the image, including the date and time it was acquired, and its location in GPS coordinates. Figures 3.81 and 3.82 show an example of the interactive map when multiple images are displayed. Each marker represents the location of a graffiti or tattoo from the search results. From this map the user can click on any of the markers to see a thumbnail of the graffiti or tattoo, its location in GPS coordinates, and a link to obtain more information about the graffiti or tattoo. Figure 3.83 shows an example.

In the “More information” section, the user can see the information available in the database for a specific graffiti or tattoo. Figure 3.84 shows an example. The image can be clicked to enlarge it in a new window. Also, there are two additional options: “Show in map”, and “Edit image details”.

### **Upload Image**

The “Upload image” feature (Figures 3.85 and 3.86) allows a user to upload an image to the database.

Once the image is uploaded, fields can be filled in by the user. These include:

- Assign GPS coordinates
  - By known address
  - By clicking on map
- User information
  - First responder name


- First responder ID
- Graffiti/Tattoo information
  - Image Type
  - Surface type (if graffiti)
  - Body part (if tattoo)
  - Prison (if tattoo)
- Additional information
  - Gang name: from drop-down menu of known gangs or user’s input
  - Gang member: gang member involved in the graffiti
  - Comments

Figures 3.87 and 3.88 show examples of filled fields adding information to the graffiti.

Clicking on “Submit Image” completes the editing and shows the user the final output of the image uploading session. Figure 3.84 is an example of this (the same information as clicking on “More information” when browsing the graffiti database).

## Upload Multiple Images

The “Upload multiple images” feature (Figure 3.89) allows a user to upload multiple images to the database at the same time. By clicking on “Select files” the user can browse the computer to select one or multiple images to upload to the server (Figure 3.90). Multiple images can be selected using the SHIFT or CTRL buttons on the keyboard. By holding SHIFT when clicking on two files, it will select everything in between them. By holding CONTROL when clicking on files, it will select individual images. Once the images are selected a list of files to upload will be created as shown in Figure 3.91. By clicking on “Upload selected files” the images are uploaded to the server. As the images are being uploaded, the progress is shown to the user (Figure 3.92). Once all the images are uploaded, a preview screen is shown to the user, where basic information is automatically populated for each image (Figure 3.93). For each image, the user can populate the same fields as when using the feature “Upload image”. After populating all the necessary fields, the user can click on “Submit images” located below the last image to update the information on the server. The results of the submission are shown as seen on Figure 3.79. Note that until the user clicks on “Submit images” no images are added to the database.



**VIPER**  
Video and Image Processing Laboratory

**VACCINE**  
Visual Analytics for Criminal, Civil, and Terrorism-related Environments  
A/C/E Department of Homeland Security Center of Excellence

[Main](#)  
[Archive](#)

### Gang Graffiti Recognition and Analysis Using a Mobile Telephone

Gangs are a serious threat to public safety throughout the United States. Gang members are continuously migrating from urban cities to suburban areas. They are responsible for an increasing percentage of crime and violence in many communities.

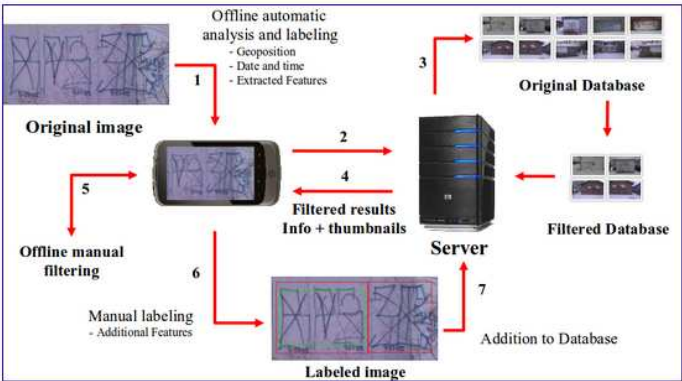
According to the National Gang Threat Assessment, approximately 1 million gang members belonging to more than 20,000 gangs were criminally active within all 50 states and the District of Columbia as of September 2008. Criminal gangs commit as much as 80 percent of the crime in many communities according to law enforcement officials throughout the nation.

Street gang graffiti is their most common way to communicate messages, including challenges, warnings or intimidation to rival gangs. It is, however, an excellent way to track gang affiliation and growth, or even sometimes to obtain membership information.

The goal of this project is to use the knowledge gained from our work in mobile devices and applications and leverage it towards the development of a mobile-based system capable of image analysis. This system will provide an accurate and useful output to a user based on a database of gang graffiti images.

The image analysis includes obtaining the metadata (geoposition, date and time) and extracting relevant features (e.g., color, shape) from the gang graffiti image. The information is sent to a server and compared against the graffiti image database. The matched results are sent back to the device where the user can then review the results and provide extra inputs to refine information. Once the graffiti is completely decoded and interpreted, it is labeled and added to the database.

This project is funded by the Department of Homeland Security's Visual Analytics for Command, Control and Interoperability Environments Center of Excellence (VACCINE) at Purdue University.



**System Overview**

#### Principal Investigators

**Edward J. Delp**, The Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering, Purdue University

**Mireille Boutin**, Assistant Professor of Electrical and Computer Engineering, Purdue University

#### Graduate Students

**Albert Parra Pozo**, Graduate Student, School of Electrical and Computer Engineering, Purdue University

[Main](#) | [About](#) | [Publications](#) | [News](#)

Last modified: Thu, 15 Sep 2011 00:27:51 EDT

Fig. 3.75.: Main Page of the Web Interface of GARI.



Fig. 3.76.: “Archive” Section of Desktop GARI.



Fig. 3.77.: “Browse database” section of the web-based interface for GARI.

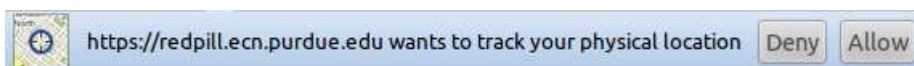


Fig. 3.78.: The current location of the user is only acquired upon request.



**VIPER**  
Video and Image Processing Laboratory

**VACCINE**  
Video and Image Processing Laboratory  
A U.S. Department of Homeland Security Center of Excellence

- [Main](#)
- [Archive](#)
- [Browse database](#)
- [Upload image](#)
- [User settings](#)
- [Logout](#)
- [GM State Colors](#)

**Browse Database**

Total: 302 graffiti

[Show all images in map](#)  
[Show images with real GPS in map \(208\)](#)



**Date/Time captured:** 2011-01-28 10:47:47  
**Address:** 2413 Yandes St, Indianapolis, IN 46205, USA  
[More information](#)  
**Image ID:** 1682



**Date/Time captured:** 2011-06-07 11:44:30  
**Address:** 280 N Holmes Ave, Indianapolis, IN 46222, USA  
[More information](#)  
**Image ID:** 1877



**Date/Time captured:** 2011-01-28 10:30:47  
**Address:** 2601-2699 Trace 26, West Lafayette, IN 47906, USA  
[More information](#)  
**Image ID:** 1637

Fig. 3.79.: Results of browsing the database.

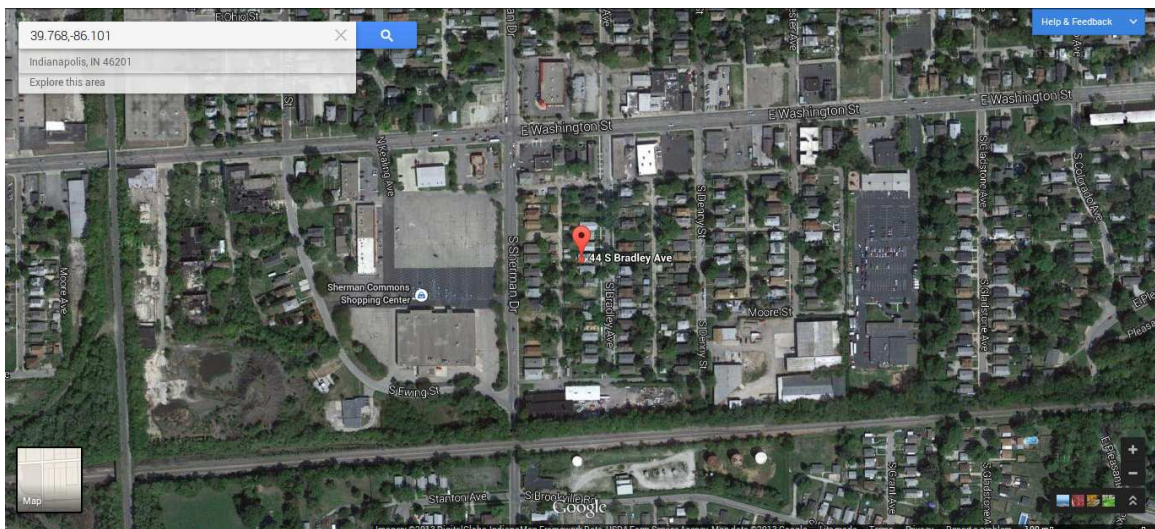


Fig. 3.80.: Example of the interactive map when a single image is displayed.



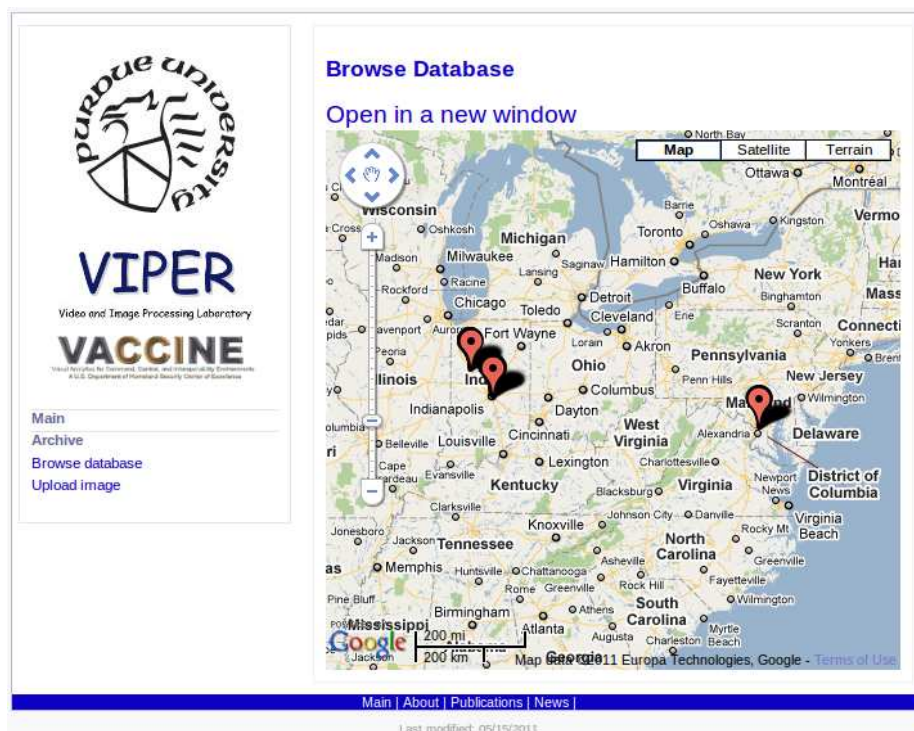


Fig. 3.81.: Example of the interactive map when multiple images are displayed.

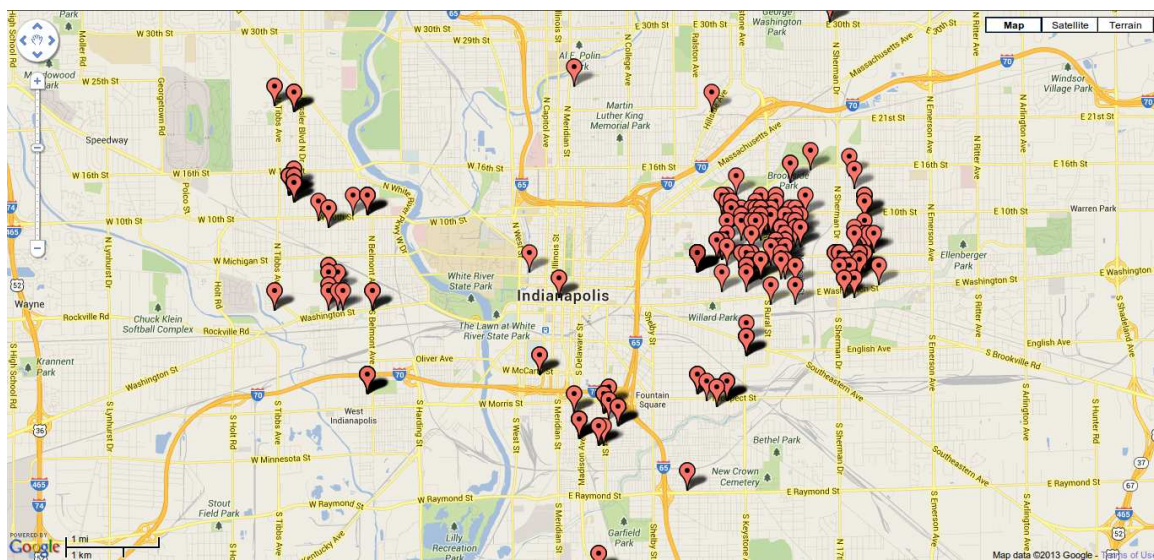


Fig. 3.82.: If “Open in a new window” is clicked, the interactive map expands to a full screen to make navigation easier.

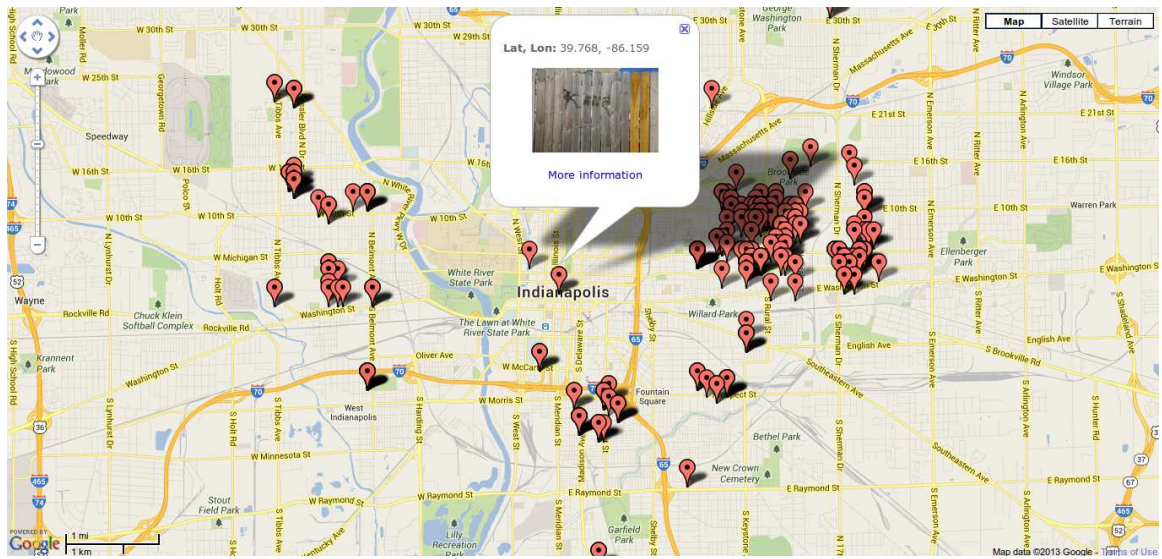



Fig. 3.83.: Example of a popped out balloon on the interactive map when a marker is clicked.




**VIPER**  
Video and Image Processing Laboratory

**VACCINE**  
Video and Image Processing Laboratory  
A/CIS Department of Human & Security Center of Excellence

- [Main](#)
- [Archive](#)
- [Browse database](#)
- [Upload image](#)
- [User settings](#)
- [Logout](#)
- [GM State Colors](#)

### Browse Database



**BASIC INFORMATION**

**Image ID:** 1193  
**Date and Time taken:** 2011-01-28 09:13:13  
**Date and Time uploaded:** 2011-05-16 22:09:00  
**Uploaded by:** Albert Parra  
**Last modification:** 2011-09-20 22:57:14  
**Last modified by:** Albert Parra  
**Address:** [93-101 Independence Dr, Hyannis Port, MA 2601, USA](#)  
**GPS latitude:** 41.676181  
**GPS longitude:** -70.300226  
**Real GPS coordinates?** FALSE

**GRAFFITI INFORMATION**

**Gang name (IA\*):** 18th Street  
**Gang name (GT\*):** 18th Street  
**Gang ID (IA):** 23  
**Gang ID (GT):** 23  
**Gang member (IA):**  
**Gang member (GT):** Leonardo Bergara  
**Gang member ID (IA):**  
**Gang member ID (GT):** 111256

**IMAGE PROPERTIES**

**File Size (bytes):** 582530  
**Height (px):** 1552  
**Width (px):** 2592

**DEVICE PROPERTIES**

**Camera make:** HTC  
**Camera model:** HTC Desire  
**Focal length (mm):** 4.31  
**X resolution (px):** 180  
**Y resolution (px):** 180  
**YCbCrPositioning:** 1  
**F number:** 5  
**Compressed bits per pixel:** 5  
**Exposure time (s):** 0.00625  
**Exposure bias (apex):** 0  
**Aperture (apex):** 4.59375  
**Metering mode:** 5  
**Flash:** 24  
**Interoperability offset:** 3334  
**Sensing method:** 2  
**Custom rendered:** 0  
**White balance:** 0  
**Digital zoom ratio:** 1  
**Exposure mode:** 0

**ADDITIONALL INFORMATION**

**Platform:**  
**Comments:** A member of the rival gang has been targeted for assassination

\*IA: Image Analysis, GT: Ground Truth

[Main](#) | [About](#) | [Publications](#) | [News](#)

Fig. 3.84.: Example of “More information” result for a specific search in the database.



Fig. 3.85.: “Upload Image” Section of Desktop GARI.



Fig. 3.86.: Preview of an Image Before Uploading It to the Graffiti Database.

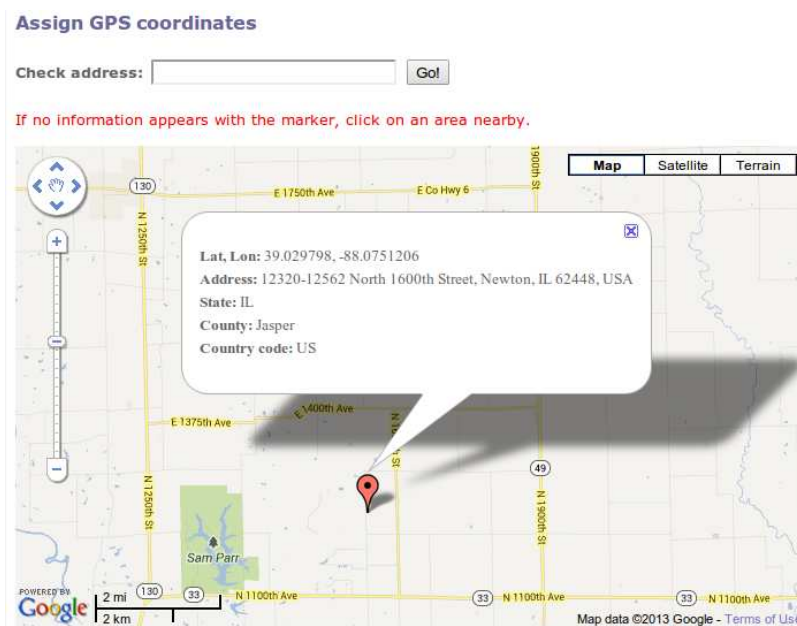


Fig. 3.87.: After uploading the image to the database, the user can select where the image was taken using an interactive map.

**User information**

First responder name:

First responder ID:

**Graffiti/Tattoo information**

Image type:  Graffiti  Tattoo

(Graffiti) Surface Type:

(Tattoo) Body Part:

(Tattoo) Prisons:

**Additional information**

Real GPS:

Gang name:  other:

Gang member:

Comments:

Fig. 3.88.: After uploading the image to the database, information can still be added.

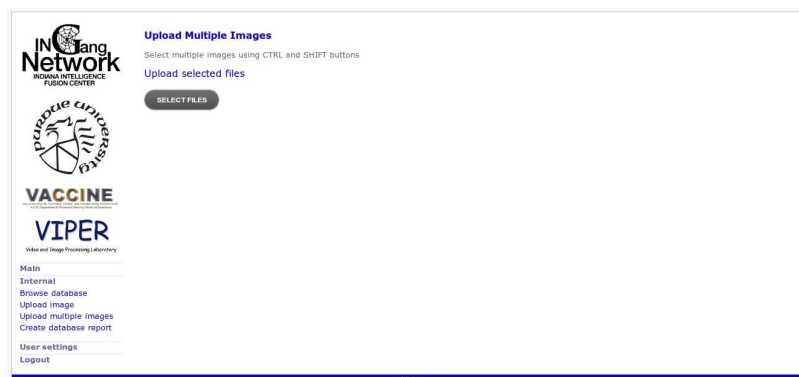


Fig. 3.89.: Upload multiple images: Main screen.

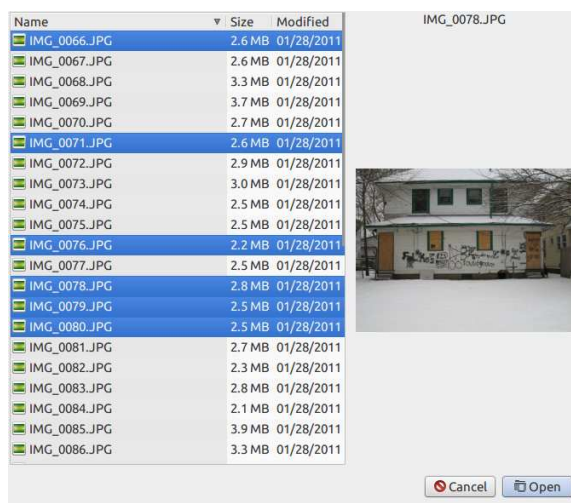


Fig. 3.90.: Upload multiple images: Select multiple files. Note that the appearance of this screen may vary depending on the operating system used.

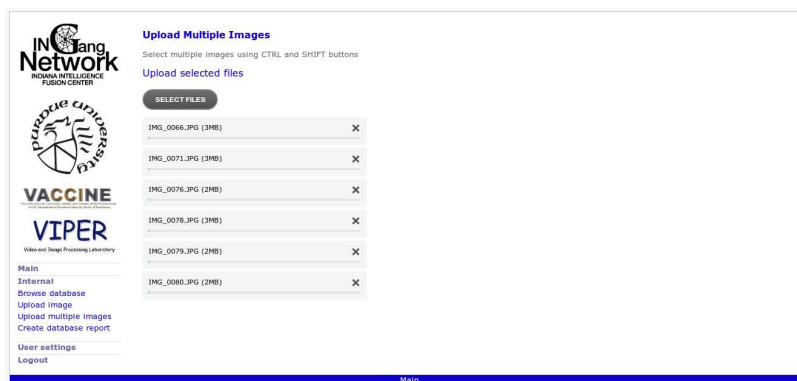


Fig. 3.91.: Upload multiple images: List of images to upload.



Fig. 3.92.: Upload multiple images: Upload progress.

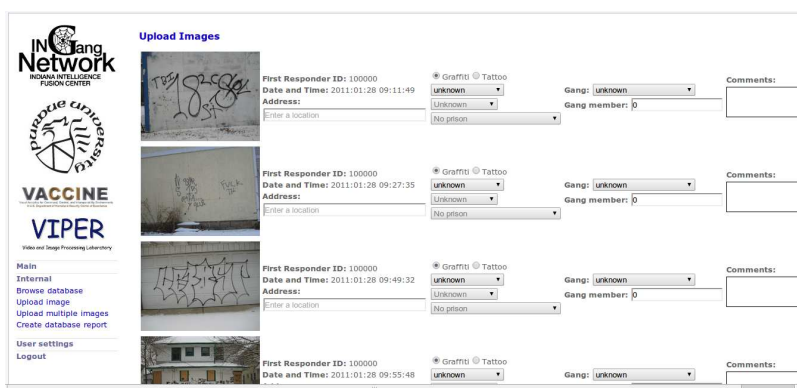


Fig. 3.93.: Upload multiple images: Review screen.

### **Create Database Report**

The “Create database report” feature (Figure 3.94) allows a user to download a spreadsheet containing information from the database.

The available fields are:

- Image ID
- Path to the image file
- First responder name
- First responder ID
- Upload date and time
- Image size
- Image height
- Image width
- Camera make
- Camera model
- GPS longitude
- GPS latitude
- Address
- City
- County
- State
- ZIP code



- Country
- Comments

Multiple fields can be selected using the SHIFT or CTRL buttons on the keyboard. By holding SHIFT when clicking on two fields, it will select everything in between them. By holding CONTROL when clicking on fields, it will select individual fields. The number of entries to be downloaded range from 200 to all the entries on the database (i.e. all images on server). The entries to be downloaded can also be sorted by date and time. After clicking on “Submit” a spreadsheet is automatically created, and a link to the download is provided to the user, as shown in Figure 3.95.

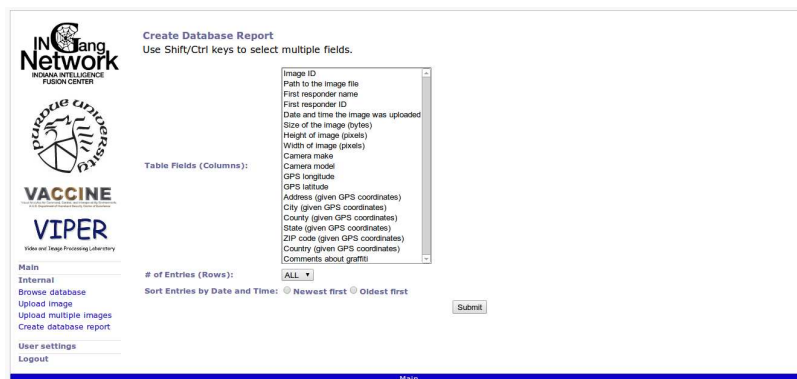


Fig. 3.94.: Create database report.



Fig. 3.95.: Create database report: download screen.

## Security

Access and navigation to the web interface are established and managed using encrypted Secure Sockets Layer (SSL) sessions. SSL encrypts information both during the transmission. The user must log in using authorized credentials before entering the archive. Figure 3.96 shows the login page. Once successfully logged in an SSL session is created and maintained for the current user. The user account can be managed by clicking on the “User Settings” link on the left sidebar. Note that currently the only option available is password change.



The screenshot shows a web interface for the VIPER VACCINE archive. On the left side, there is a logo for Purdue University, followed by the text "VIPER Video and Image Processing Laboratory" and "VACCINE". Below this, there is a sidebar menu with links: "Main", "Archive", "Browse database", and "Upload image". The main content area contains a login form with two input fields: "User ID:" and "Password:". A "Submit" button is located to the right of the "Password:" field. At the bottom of the page, there is a blue navigation bar with links: "Main | About | Publications | News |".

Fig. 3.96.: Login Page for Accessing the Gang Graffiti Archive.

## 4. MOBILE EMERGENCY RESPONSE GUIDE (MERGE)

### 4.1 Review of Existing Methods

In this section we review some relevant literature in the areas of sign location detection and sign recognition.

#### 4.1.1 Sign location detection

Sign location detection methods can be classified into three main categories: shape-based [248], color-based [249] and saliency-based [250].

Shape-based approaches first generate an edge map and then use shape information to find objects. For example, in [251] triangular, square and octagonal road signs are detected exploiting properties of symmetry and edge orientations exhibited by equiangular polygons. In [252] a road-sign detection system is based on support vector machines (SVM). It uses shape classification using linear and Gaussian-kernel SVMs. In most cases, the methods are invariant to translation, rotation, scale, and, in many situations, to partial occlusions. In [253] the authors present a system for detection and recognition of road signs with red boundaries and black symbols inside. Pictograms are extracted from the black regions and then matched against templates in a database. They propose a fuzzy shape detector and a recognition approach that uses template matching to recognize rotated and affine transformed road signs. In [254] the authors propose a system for automatic detection and recognition of traffic signs based on maximally stable extremal regions (MSERs) and a cascade of support vector machine (SVM) classifiers trained using histogram of oriented gradient (HOG) features. The MSER offers robustness to variations in lighting conditions. The system works on images taken from vehicles, operates under a range of weather

conditions, runs at an average speed of 20 frames per second, and recognizes all classes of ideogram-based (nontext) traffic symbols from an online road sign database.

Other shape-based approaches use “shape descriptors”, which can be generally classified into two methods: contour-based methods and region-based methods [255, 256]. Contour-based methods only exploit the boundary information while region-based methods exploit all the pixels within a region. Contour-based methods are widely used in many applications because of their simplicity [168]. Although shape signatures obtained through contour-based methods are not generally robust to noise [168] the Fourier descriptor (FD) overcomes noise sensitivity by usually using only the first few low frequency coefficients to describe shape. The FD is also compact and easy to normalize. Because of its properties the FD is one of the most used shape descriptors [255–259]. In addition, it has been shown that the FD outperforms many other shape descriptors [168, 260].

Previous work on FDs includes methods for generating descriptors invariant to geometric transformations and matching methods for shape similarity and image retrieval. For example, in [261] a new Fourier descriptor is proposed for image retrieval by exploiting the benefits of both the wavelet and Fourier transforms. A complex wavelet transform is first used on the shape boundary, and then the Fourier transform of the wavelet coefficients at multiple scales is examined. Since FDs are used at multiple scales, the shape retrieval accuracy improves with respect to using ordinary FDs. FDs are analyzed as feature vectors in [262] for pedestrian shape representation and recognition. The results showed that only ten descriptors of both low and high frequency components of pedestrian and vehicle shapes are enough for accurate recognition. Shape context from [185] is used in [263] to generate descriptors and proposed a matching method that uses correspondences between two shapes based on ant colony optimization. In [264] the authors describe simple shapes using FDs based on chain codes and the Fourier transform. The first ten coefficients are used to approximate the shapes. In [257] the authors use the Fourier transform of local regions on the output of a MSER detector. They propose a FD matching method that

uses the phase information to extract the orientation of the shape and used the FDs for recognizing road signs. However, this method fails when signs have low resolution.

Color-based approaches overcome the problems of shape variation, partial occlusion, and perspective distortion. However, colors are sensitive to lightning conditions and illumination changes. To deal with these disadvantages, some color spaces that keep sign color almost invariant are used in existing methods. For example, in [265] sign detection is done using a color-based segmentation method as a preprocessing step for shape detection. Color-based segmentation is used to achieve real time execution, since color-based segmentation is faster than shape-based segmentation. In [266] several color components are used to segment traffic signs under different weather conditions. Various color spaces are analyzed to detect traffic prohibitive signs, alert signs and guide signs.

Saliency-based approaches utilize selective visual attention models, which imitate human early visual processing in order to overcome the above problems in complex scenes. This paper makes use of the saliency-based visual attention models to construct a hazmat sign saliency map as a sign localization method. Visual saliency is closely related to how we perceive and process visual stimuli and it is often characterized by variant object features, like color, contrast, gradient, edge, and contour. Theories of human visual attention hypothesize that the human vision system only processes parts of an image in detail while leaving others nearly unprocessed [267]. A saliency-based visual attention (SBVA) model was presented in [250] using images features with a Gaussian pyramid. A graph-based visual saliency (GBVS) method was proposed in [268], to highlight conspicuous regions. This method allows combinations with other visual attention maps. A dynamic visual attention (DVA) model based on the rarity of features is proposed in [269]. A histogram-based contract (HC) method and a region-based contract (RC) method were introduced in [270] to construct saliency maps. HC-maps produce better performance over RC-maps but at

the expense of increasing the computation time. A multi-scale dissimilarity aggregation (MSDA) method is used to estimate the saliency of regions in [271]. A saliency map generation method was described in [272] using image signature (IS) to highlight sparse salient regions based on RGB or Lab color spaces. An saliency detector based on hypercomplex Fourier transform (HFT) is presented in [273] using the convolution of the image amplitude spectrum with a low-pass Gaussian kernel.

#### 4.1.2 Sign recognition

Sign recognition methods can be classified into: geometric constraint methods, boosted cascades of features, and statistical moments [274–276].

Methods based on geometric constraints include the use of Hough-like methods [277,278], contour fitting [279,280], or radial symmetry detectors [281,282]. These approaches apply constraints on the object to be detected, such as little or no affine transformations, uniform contours, or uniform lightning conditions. Although these conditions are usually met, they cannot be generalized. For example, [278] presents an analysis of Hough-like methods and confirms that the detection of signs under real-world conditions is still unstable. A novel Hough-like technique for detecting circular and triangular shapes is also proposed, in order to overcome some of the limitations exposed.

Methods based on the boosted cascades of features commonly use the Viola-Jones framework [283–285]. These approaches often use object detectors with Haar-like wavelets of different shapes, and produce better results when the feature set is large. For example, in [284] a system for detection, tracking, and classification of U.S. speed signs is presented. A classifier similar to the Viola-Jones detector is used to discard objects other than speed signs in a dataset of more than 100,000 images. In [285] the detection is based on a boosted detectors cascade, trained with a version of Ad-

aboost, which allows the use of large feature spaces. The system is robust to noise, affine deformation, partial occlusions, and reduced illumination.

Methods based on statistical moments [286–288] use the central moments of the projections of the object to be detected. They can be used to check the orientation of the object, or to distinguish between different shapes such as circles, squares, triangles, or octagons. These methods are not robust to projective distortions or non-uniform lighting conditions. For example, in [288] a mobile-based sign interpretation system uses detection of shapes with an approximate rotational symmetry, such as squares or equilateral triangles. It is based on comparing the magnitude of the coefficients of the Fourier series of the centralized moments of the Radon transform of the image after segmentation. The experimental results show that the method is not robust to projective distortions.

## 4.2 Segment Detection Using Geometric Constraints

Figure 4.1 shows the block diagram of the proposed method. We find edges in the image using the Canny edge detector. Since hazmat signs can be present at various distances, we use median auto-thresholding. To deal with non-uniform illumination changes in the scene, we also grayscale histogram equalize the image. We assume: 1) any sign in the image has to be approximately upright with its major axes aligned with the  $XY$  axis; and 2) the projective distortion has to be small. (i.e., edges have to be approximately at  $\pm 90^\circ$  with respect to each other).

Given these assumptions, we use morphological filters to eliminate edges not belonging to a hazmat sign. We create flat linear structuring elements of length  $L_{se} = 10$  pixels at  $\theta_{se} = \pm 45^\circ$  and use them separately to erode the Canny edge map. Figure 4.2 shows the structuring elements used for erosion.

The resulting edge map is the superposition of the two erosions. We then find line segments using the Standard Hough Transform [226, 227] (already explained in

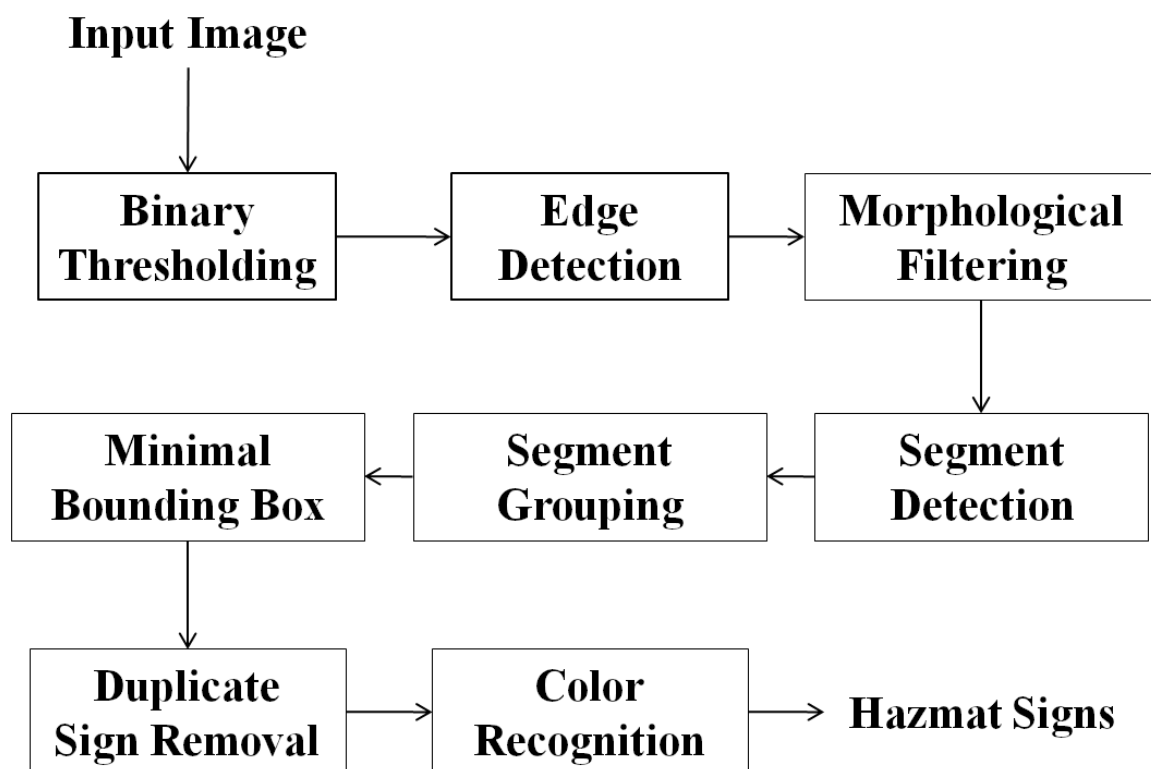


Fig. 4.1.: Segment Detection Using Geometric Constraints.

0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

(a) Linear Structuring Element at  $+45^\circ$ (b) Linear Structuring Element at  $-45^\circ$ 

Fig. 4.2.: Structuring Elements Used for Erosion.



Section 3.5.3). We set the minimum gap allowed between points on the same line to  $N_{gap}^L = 5$  pixels and the maximum gap to  $N_{gap}^H = 0.05 \max(W_X, H_X)$ , where  $(W_X, H_X)$  are the width and height of the image respectively.

We next proceed to group the segments into candidates. Each candidate consists of a set of segments having one reference segment, at least one parallel segment, and two orthogonal segments (one to the left and one to the right of the reference segment). The reference segment is chosen at random from the list of segments that have not been grouped yet. Parallel segments need to have similar slope and length relative to the reference segment. The thresholds are set so that  $|m_p - m_r| < T_m$  and  $|l_p - l_r| < T_l$ , where  $m_p$  and  $m_r$  are the slopes of the parallel and reference segments respectively,  $l_p$  and  $l_r$  are the lengths of the parallel and reference segments respectively,  $T_m = 0.1$ ,  $T_l = 0.75e$  and  $e = \max(l_p, l_r)$ . The distance  $d$  between the reference and the parallel segments has to be in the range  $T_d^L < d < T_d^H$ , where  $T_d^L = 0.5e$  and  $T_d^H = 2.5e$ . This distance is defined between the middle points of the parallel and the reference segments. Also, the angle between the reference and the parallel segments has to be less than  $\theta_{RP} = 20^\circ$ . This angle is defined by the normal of the parallel segment at its middle point and the vector joining the middle points of the parallel and the reference segments. Orthogonal segments need to have opposite slope and similar length to the reference segment, that is,  $|m_o + 1/m_r| < T_m$  and  $|l_o - l_r| < T_l$ , where  $m_o$  and  $l_o$  are the slope and the length of the orthogonal segment. The distance  $d$  between the reference and the orthogonal segments has to be in the range  $T_d^L < d < T_d^H$ . The angle between the reference and the orthogonal segments is defined as positive when the orthogonal segment is to the right of the reference segment, and defined as negative when the orthogonal segment is to the left of the reference segment.

For each candidate set satisfying the geometric constraints we compute its minimal bounding box. We then discard any candidate with a bounding box aspect ratio smaller than  $T_{BB} = 1.3$ .

Finally, we check the remaining candidates and remove those that correspond to the same sign. This can be done by first dividing all bounding boxes that overlap more than  $T_{overlap} = 50\%$  into groups, and then finding the optimal bounding box for each group. We consider the optimal bounding box to be the one with its nodes closest to its centroid (i.e. closest to a square).

Figure 4.3 illustrates an example of the complete process. Once a hazmat sign is segmented, its color is set to the average hue inside the optimal bounding box and the color is used to identify the sign. We also do basic text recognition inside the detected region using the open source Optical Character Recognition (OCR) engine OCRAD [289]. Although the accuracy of OCRAD is far below other state-of-the-art OCR engines, it was chosen for its speed [290]. Note that the text recognition step is applied just for testing purposes. Other text recognition methods will be investigated in the future (see Section 6).



Fig. 4.3.: First method (left to right): original image, segments at  $\pm 45^\circ$ , grouped segments, optimal bounding box.

Table 4.1 shows all the parameters/thresholds we used including empirically derived parameters.

### 4.3 Convex Quadrilateral Detection Based on Saliency Map<sup>1</sup>

Our first method described above has some drawbacks:

---

<sup>1</sup>The work presented in this section was done by the author jointly with Bin Zhao.

Table 4.1: Parameters and thresholds used in Segment Detection Using Geometric Constraints.  $W_X$  and  $H_X$  are the width and height of  $X$  respectively.  $e = \max(l_p, l_r)$

Parameter	Description	Value
$L_{se}$	Length of structuring elements for erosion	10 px
$\theta_{se}$	Orientation of structuring elements for erosion	$\pm 45^\circ$
$N_{gap}^L$	Maximum gap for Standard Hough Transform	5 px
$N_{gap}^H$	Minimum gap for Standard Hough Transform	$0.05 \max(W_X, H_X)$
$T_m$	Slope threshold	0.1
$T_l$	Length threshold	$0.75e$
$T_d^L$	Low distance threshold between segments	$0.5e$
$T_d^H$	High distance threshold between segments	$2.5e$
$\theta_{RP}$	Angular threshold between segments	$20^\circ$
$T_{BB}$	Bounding box ratio threshold	1.3
$T_{overlap}$	Bounding box overlap threshold	50%

- **Grayscale:** By converting the original RGB image to grayscale we lose color information. This can cause the hazmat sign to have similar intensity values as the background given specific illumination conditions. Figure 4.4 illustrates an example. The edge detection process cannot separate the top corner of the sign from the background, thus losing the necessary edges to continue the recognition process.
- **Low resolution/Blurry:** With low resolution or blurry images, the resulting edge map will not contain straight edges at  $\pm 45^\circ$  and the erosion process will then delete most of them. Figure 4.5 shows an example.
- **Distortion:** Hazmat signs not satisfying the two assumptions of the first method will be removed during the erosion process. Figure 4.6 shows an example.
- **Line overlap:** The gap threshold of the Standard Hough Transform may cause the segment grouping process to merge two segments from two close signs, as shown in Figure 4.7.
- **Shade:** The image contains shade that can alter the color of the sign. Figure 4.8 illustrates an example. The result is an unsuccessful color recognition once the hazmat sign is detected.

Our second technique replaces the initial edge detection with a saliency map to detect regions potentially containing hazmat signs<sup>2</sup>. The block diagram in Figure 4.9 shows the block diagram of the proposed method. Figures 4.10 and 4.11 illustrate examples of the saliency maps obtained on the Lab and RGB color spaces. Note how the saliency map applied on the RGB color space does better on black or white signs (low chroma region), while the Lab color space does better on the rest of the signs.

We apply visual saliency models to the input images represented in both RGB and Lab color spaces. In each color space, two saliency maps are constructed using

---

<sup>2</sup>This work was done by Bin Zhao.



Fig. 4.4.: Issue With First Method: Grayscale. Sign Is Lost On Line Detection Process.

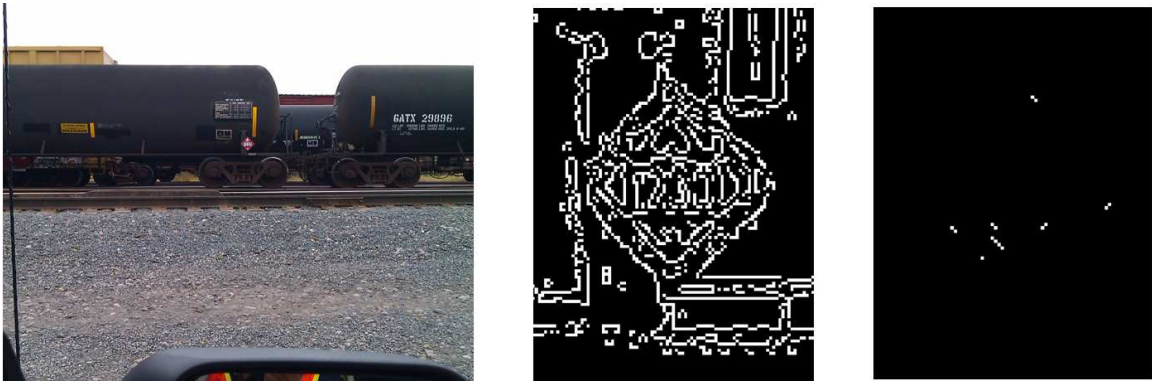


Fig. 4.5.: Issue With First Method: Low Resolution. Sign Is Lost On Erosion Process.

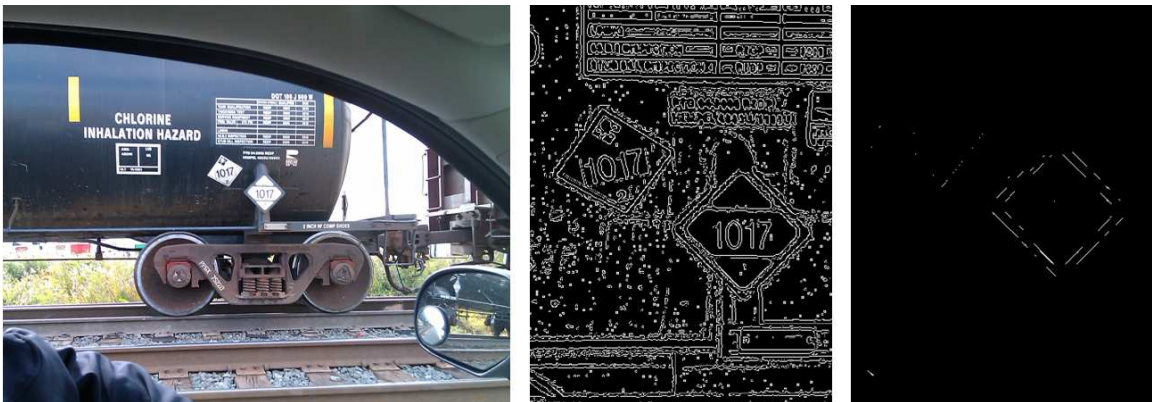


Fig. 4.6.: Issue With First Method: Sign Distortion. Sign Is Lost On Erosion Process.



Fig. 4.7.: Issue With First Method: Segment Merging. Sign Is Lost On Segment Grouping Process.



Fig. 4.8.: Issue With First Method: Shade. Sign Color Is Not Recognized Properly.

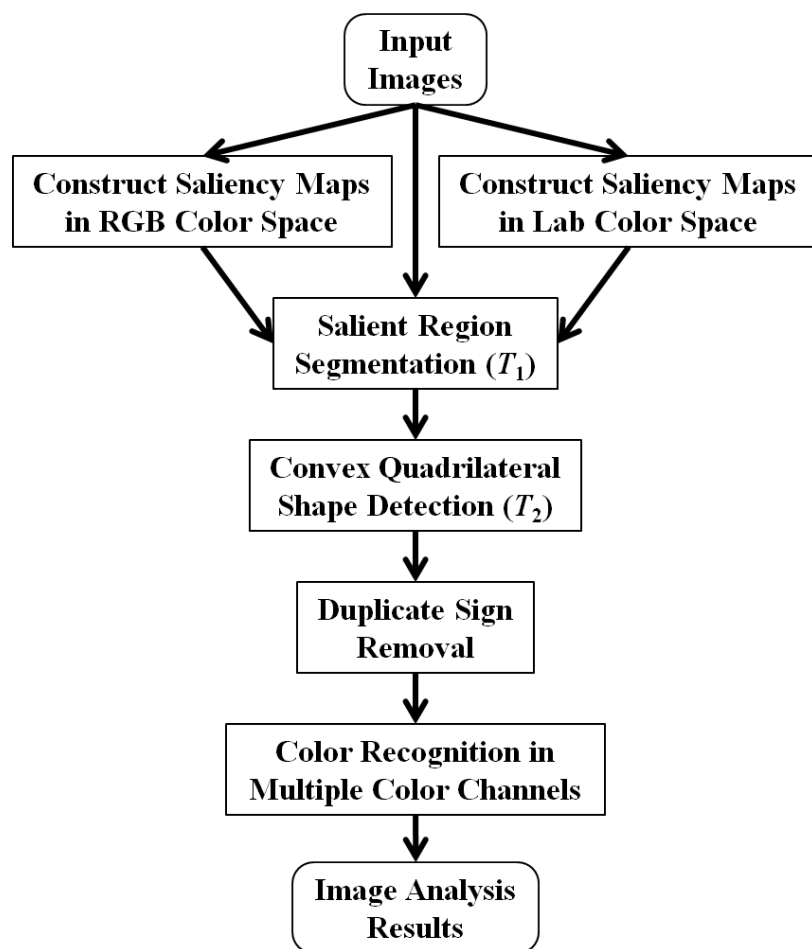


Fig. 4.9.: Proposed Hazmat Sign Detection and Recognition Method.



Fig. 4.10.: Saliency Map Method Obtained On Lab (Middle) and RGB (Right) Color Spaces.

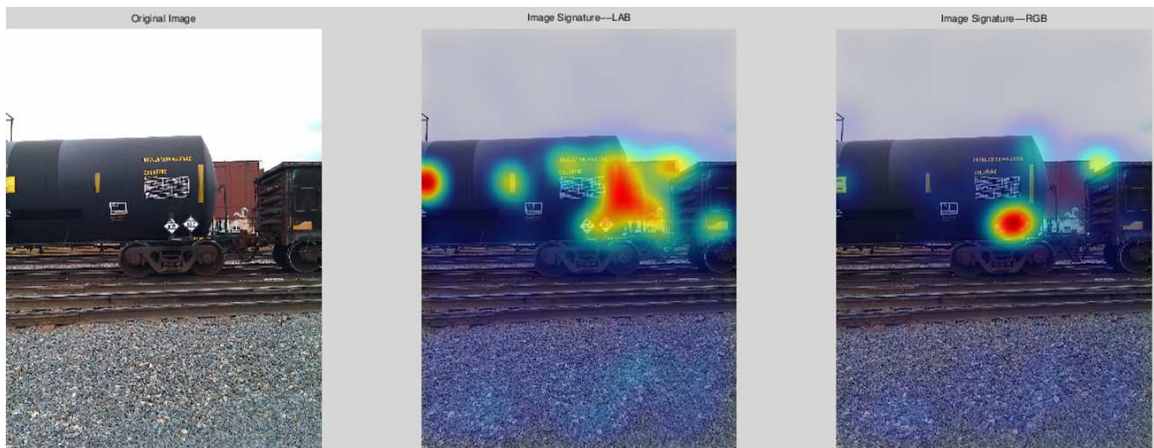


Fig. 4.11.: Saliency Map Method Obtained On Lab (Middle) and RGB (Right) Color Spaces.



two visual saliency models separately, i.e. IS [291] and HFT [273]. The saliency maps assign higher saliency value to more visually attractive regions. Note that the original HFT method uses the I-RG-BY opponent color space. We modified this method to use RGB and Lab color components with different weights ( $W_{RGB} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$  for RGB and  $W_{Lab} = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$  for Lab). The combined saliency map method, denoted as IS+HFT(RGB+Lab), generates four saliency maps (two for RGB and two for Lab) and produces the best results in the experiments (see Section 5.2.2). We threshold each saliency map to create a binary mask to segment the salient regions from the original image. The threshold  $T_1$  is determined as  $k$  times the average saliency value of a given saliency map. That is,  $T_1 = \frac{k}{W \times H} \sum_{x=1}^W \sum_{y=1}^H S(x, y)$ , where  $W$  and  $H$  are the width and height of the saliency map,  $S(x, y)$  is the saliency value at position  $(x, y)$  and  $k$  is empirically determined for the combined saliency map method ( $k = 4.5$  for IS and  $k = 3.5$  for HFT).

For each salient region found, we detect signs using specific color channels. Hazmat signs in our datasets contain either one or two of the following colors: black, white, red, blue, green, yellow. We then divide the input image into six color channels and we process them as separate images. The red, green and blue channels are obtained from the RGB color space. The yellow channel is obtained from the CMYK color space. The black and white channels are obtained by thresholding the Y channel.

This allows us to do both sign detection and color recognition at the same time, since we will assume that the color of any hazmat sign found in the region will correspond to the color channel associated to it. Note that although our dataset does not contain orange hazmat signs, they exist and can appear in the future. We would then be able to extract a seventh channel by transforming the image from RGB to a hue-based color space and then segment the hue channel.

The grayscale and the color channels are thresholded to account for highly chromatic areas using an empirically determined threshold  $T_2$  (85 for black, 170 for white, and 127 for color). Note that this last threshold can be avoided by working with a hue-based color space. Each of the thresholded images is binarized, and morpholog-

ically opened to remove small objects containing less than  $N_{px}^O = 0.05\%WH$ . We also use dilation with a flat, disk-shaped structuring element of size  $S_{se} = 7$  to merge areas that may belong to the same object. Figure 4.12 shows the structuring element used for dilation.

0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

Fig. 4.12.: Structuring Element Used for Dilation.

We then retrieve the contours from the resulting binary image [292]. For each contour, we use the Standard Hough Transform [226, 227] to find straight lines that approximate the contour as a polygon. The intersections of these lines give us the corners of the polygon, which can be used to discard non-quadrilateral shapes. If the contour is approximated by four vertices, we find its convex hull [293]. If the convex hull still has four vertices, we check the angles formed by the intersection of its points. If each of these angles is in the range  $T_\theta^v = 90^\circ \pm 1.5^\circ$ , and the ratio of the sides formed by the convex hull is in the range  $T_r^e = 1 \pm 0.5$ , we can assume that we have found a convex quadrilateral.

Finally, we use the same technique as in the first method to remove quadrilaterals that correspond to the same hazmat sign. Figure 4.13 illustrates a successful detection of two signs, one is affected by rotation and perspective distortion. Figure 4.14 illustrates a successful detection of one sign and also a false positive. In this particular

case the issue could be addressed by using an optical character recognition to detect the text inside the sign candidate.



Fig. 4.13.: Second Method: True Positives.



Fig. 4.14.: Second Method: True Positive/False Positive.

Our second method offers multiple advantages. First, it is robust to rotation, since there is no erosion at  $\pm 45^\circ$ . Second, it is robust to perspective distortion, since convex quadrilaterals can be skewed. Third, it is able to detect signs close to each other, since there is no overlapping of line segments caused by the Standard Hough Transform. Fourth, it is more robust to blurred and low resolution images, since there is no edge detection is on the sign recognition step. Lastly, it is more robust

Table 4.2: Parameters and thresholds used in Convex Quadrilateral Detection Based on Saliency Map.  $W$  and  $H$  are the width and height of the saliency map.  $S(x, y)$  is the saliency value at  $(x, y)$

Parameter	Description	Value
$W_{RGB}$	RGB weights for saliency model	$[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$
$W_{Lab}$	Lab weights for saliency model	$[\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$
$T_1$	Saliency map threshold	$\frac{k}{W \times H} \sum_{x=1}^W \sum_{y=1}^H S(x, y)$
$k$	Weight included in $T_1$ (IS)	4.5
$k$	Weight included in $T_1$ (HFT)	3.5
$T_2$	Color channel threshold (black)	85
$T_2$	Color channel threshold (white)	170
$T_2$	Color channel threshold (color)	127
$N_{px}^O$	Number of pixels for opening	0.05% $WH$
$S_{se}$	Size of structuring elements for dilation	7
$T_\theta^v$	Angular threshold between convex hull vertices	$90^\circ \pm 1.5^\circ$
$T_r$	Ratio threshold between convex hull edges	$1 \pm 0.5$

to color recognition, since it detects signs already in specific color channels. The only disadvantage is its execution time. The first method uses basic geometry to find potential candidates, while the second method needs to compute a saliency map as a preprocessing step, which takes more time than the first process itself.

Table 4.2 shows all the parameters/thresholds we used including empirically derived parameters.

#### 4.4 Sign Location Detection Based on Fourier Descriptors<sup>3</sup>

The second method is robust to geometric distortions and illumination changes. However, it relies on the detection of straight edges and the relationship between their lengths and angles. This causes the process fails on low resolution images, signs with partial occlusions and deteriorated signs. We propose a third method to overcome the drawbacks caused by detections based on geometric constraints. Figure 4.15 shows the block diagram of the proposed method. We use contour shape representation and

<sup>3</sup>The work presented in this section was done by Kharittha Thongkor jointly with the author and Bin Zhao.

matching based on Fourier descriptors. Note that we do not use a saliency map to get an initial sign location estimation. Instead we use the original image as input to our system.

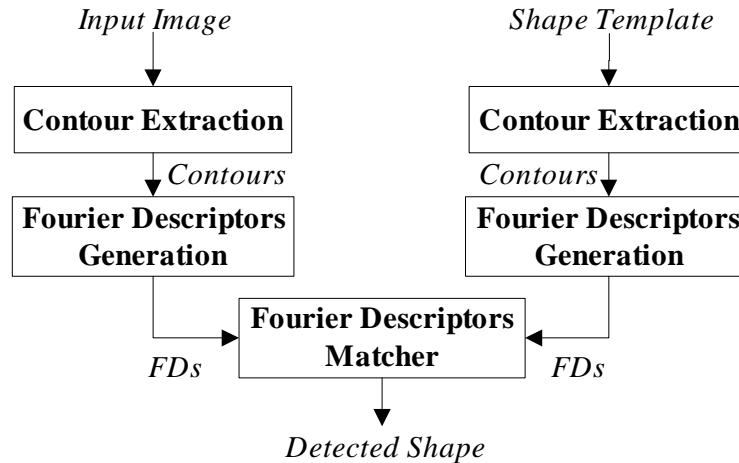


Fig. 4.15.: Sign Location Detection Based on Fourier Descriptors.

In this method we also detect hazmat sign locations in specific color channels, so no further color recognition is not required after detecting the location of the sign. As opposed to our second method, each of the six images extracted from each color channel is binarized separately. For this purpose we propose the use of color channel thresholding followed by Otsu's thresholding technique [294] to obtain the final binary image. For each of the six color channel images,  $I_i$ ,  $i \in [1, 6]$ , we first select two parameters for channel thresholding,  $T_{i_1}$  and  $T_{i_2}$ . The reason why we need  $T_{i_1}$  and  $T_{i_2}$  is that directly using Otsu's thresholding method on a channel does not produce accurate results when images contain variable illumination [295]. Histogram of each color channel can be analyzed for minima/valleys which can then be used to determine two thresholds as follows.  $T_{i_1}$  is set to

$$T_{i_1} = \min\left(\frac{255}{4}, h_{i_1}\right), \quad (4.1)$$

where  $H_{i_1}$  is the location of the first valley of the histogram of the  $i^{\text{th}}$  color channel. The first valley is the minimum point between the first two significant peaks. The

set of significant peaks  $P_1$  of a histogram  $h$  is defined as the set of points with a histogram value greater than their local maximum neighbors [296]. That is,

$$P_1 = \{(p_i, h(p_i)) | h(p_i) > \{h(p_{i-1}), h(p_{i+1})\}, p_i \in P_0\}, \quad (4.2)$$

where

$$P_0 = \{(i, h(i)) | h(i) > \{h(i-1), h(i+1)\}, 0 \leq i \leq 255\}, \quad (4.3)$$

$T_{i_2}$  is set to

$$T_{i_2} = \max\left(3\frac{255}{4}, H_{i_2}\right), \quad (4.4)$$

where  $H_{i_2}$  is the location of the last valley of the histogram of the  $i^{th}$  color channel. The color channel image  $I_i$  is then thresholded by:

$$I'_i(x, y) = \begin{cases} 0 & I_i(x, y) \leq T_{i_1} \text{ or } I_i(x, y) \geq T_{i_2} \\ I_i(x, y) & \text{otherwise} \end{cases} \quad (4.5)$$

Each image  $I'_i$  is then used as input for Otsu's thresholding method to automatically generate a threshold  $T_{i_b}$ . Finally, each original color channel image  $I_i$  is then binarized using  $T_{i_b}$ . Figure 4.16 illustrates a comparison using Otsu's method with and without our proposed color channel thresholding method. Note how Otsu's method fails to find the optimal threshold because of the high density of pixels in the sky region having high intensity values in the red channel.

As we mentioned above we use morphological techniques to merge areas in the binary image found above that may belong to the same hazmat sign. First, we use a flood-fill operation to fill holes in the binary image [297]. A hole is a set of background pixels surrounded by foreground pixels. Next, we use morphological dilation with a flat, diamond shape structuring element of size  $S_d = 5$  pixels to enlarge the boundaries of foreground areas [256, 298]. Then, we remove small objects by using morphological opening with a flat, diamond-shaped structuring element of size  $S_o = 20$  pixels. We

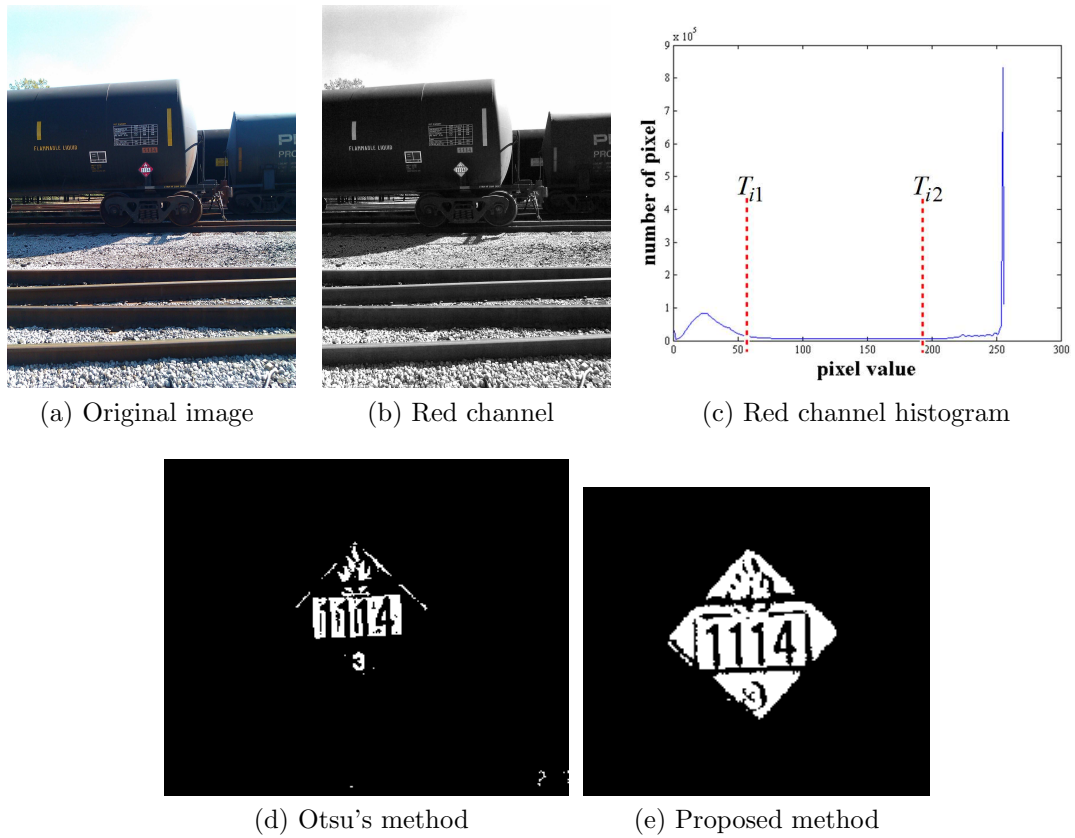


Fig. 4.16.: Example of image binarization using our proposed color channel thresholding method comparing with Otsu's method.

also remove objects containing less than  $T_c = 0.03\%$  of the total number of pixels in the image. We chose  $0.03\%$  because it is the minimum number of pixels contained in a hazmat sign in our image test set. Finally, we obtain closed contours by tracing the exterior boundaries of objects in the resulting binary image [299, 300]. Figure 4.17 shows some examples of extracted contours from input images. Note that the size of the structuring elements are empirically obtained from the ground-truth data in our dataset. They came from searching the best values that give the maximum number of signs before tracing the exterior boundaries of objects.

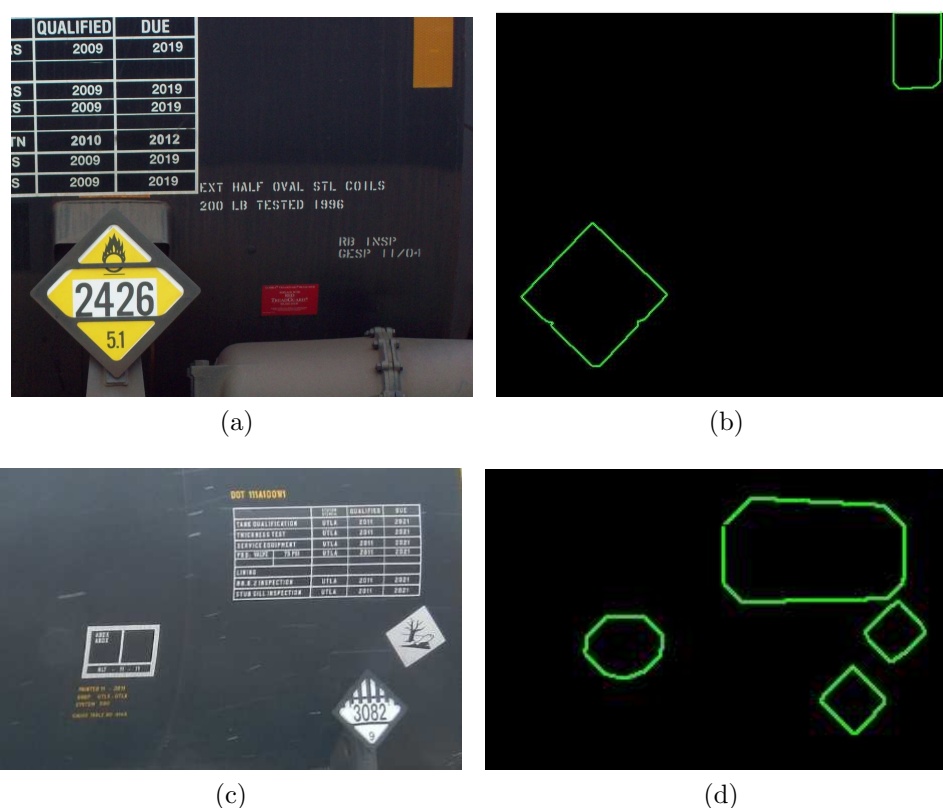


Fig. 4.17.: Examples of input images (left) and their contours (right).

Each contour found from the previous step is used to generate a Fourier Descriptor (FD). The FD describes the shape of an object through the use of the Fourier transform of the object's contour. Assuming the contour of a shape has  $N$  pixels,



numbered from 0 to  $N - 1$ , a set of coordinates describing the contour can be defined as

$$b(k) = (x(k), y(k)) = x(k) + iy(k), \quad (4.6)$$

where  $k = 0, 1, 2, \dots, N - 1$ . The Fourier transform of the contour function,  $A(v)$ , is the FD:

$$A(v) = F(b(k)) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} b(k) \exp^{-\frac{j2\pi vk}{N}}, \quad (4.7)$$

where  $v = 0, \dots, N - 1$ . To describe the shape of a boundary the Fourier coefficients have to be normalized to make them invariant to translation and scale [169, 257, 261, 262, 264].

If the 2D shape is translated by a distance  $z_0 = x_0 + jy_0$ :

$$b'(k) = b(k) + z_0 \quad (4.8)$$

its FD becomes

$$A'(v) = \frac{1}{N} \sum_{k=0}^{N-1} (b(k) + z_0) \exp^{-\frac{j2\pi vk}{N}} \quad (4.9)$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} b(k) \exp^{-\frac{j2\pi vk}{N}} + \frac{1}{N} \sum_{k=0}^{N-1} z_0 \exp^{-\frac{j2\pi vk}{N}} \quad (4.10)$$

$$= A(v) + z_0 \delta(v). \quad (4.11)$$

This means the translation only affects the DC component  $A(0)$  of the FD. Therefore, by setting the first coefficient,  $A(0)$ , to zero we make the FD invariant to translation.

If the 2D shape is scaled (with respect to origin) by a factor  $S$ :

$$b'(k) = Sb(k) \quad (4.12)$$

its FD is scaled by the same factor:

$$A'(v) = SA(v). \quad (4.13)$$

Therefore, by normalizing the energy of the remaining coefficients to 1 we make the FD invariant to scale. The normalized FD  $A'(v)$  then becomes:

$$A'(v) = \frac{A(v)}{\sqrt{\sum_{v=1}^{\infty} |A(v)|^2}}, \quad A'(0) = 0. \quad (4.14)$$

The low frequency components of  $A'(v)$  contain information about the general shape and the high frequency components contain finer details. Therefore, the first  $P$  Fourier descriptor coefficients can be used to create an approximate reconstruction of the contour  $b(k)$ ,

$$\widehat{b}(k) = \frac{1}{P} \sum_{v=0}^{P-1} A'(v) \exp\left(\frac{j2\pi vk}{N}\right), \quad k = 0, 1, 2, \dots, N - 1. \quad (4.15)$$

In order to determine if a contour obtained from an image belongs to a hazmat sign we need to compare its FD against the FD of a predefined shape template or shape contour in a process called contour matching. In this paper the shape template is a diamond shaped binary image resembling a hazmat sign (see Figure 4.18). Contour matching can be done in the spatial or frequency domain. We use matching in the frequency domain for two reasons. First, matching in the frequency domain is scale independent, as opposed to spatial domain matching. Second, matching in the spatial domain involves scanning an image multiple times modifying the scale and rotation of the shape template. Since the normalized FDs are invariant to scale and the correlation matching in frequency domain is invariant to rotation the matching is less computationally expensive. The frequency domain matching has also been shown to be more efficient [301, 302] and allows easy recognition for rotated and scaled noisy sign images [170].

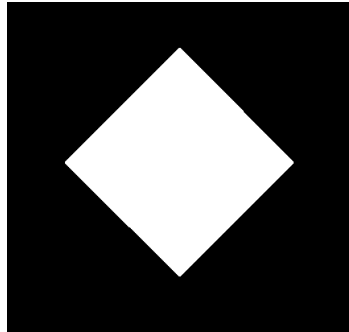


Fig. 4.18.: A diamond shaped binary image is used as a shape template.

FD matching is usually done by using only the magnitude and ignoring the phase information. By discarding the phase information we achieve rotation and starting point invariance [303]. This is because a rotation of the 2D shape by an angle  $\phi$  about the origin only introduces a phase shift in the FD:

$$b'(k) = b(k)e^{j\phi} \Rightarrow A'(v) = A(v)e^{j\phi}, \quad (4.16)$$

and a shift of the 2D shape from 0 to  $m_0$  only introduces a phase shift in the FD:

$$b'(k) = b(k - m_0) \Rightarrow A'(v) = A(v)e^{\frac{j2\pi m_0 v}{N}}. \quad (4.17)$$

However, different shapes can have similar magnitude but completely different phase information, thus making magnitude-based matching less accurate [257]. Therefore, we use a correlation-based matching cost function that uses both magnitude and phase information [257]. The cross-correlation between the shape template contour  $T$  and the image contour  $I$ ,  $r_{TI}(l)$  is

$$\begin{aligned} r_{TI}(l) &= (T * I)(l) = \int_0^K \overline{T(k)} I(l+k) dk \\ &= \sum_{v=0}^{\infty} \overline{A'_T(v)} A'_I(v) \exp^{-\frac{j2\pi vl}{K}} \\ &= F^{-1}\{\overline{A'_T} A'_I\}(v). \end{aligned} \quad (4.18)$$

$A'_T(v)$  and  $A'_I(v)$  are the normalized FDs of the template and the input contours, respectively.

By using normalized contours and complex FD matching we approximately compensate for scaling, rotation, translation and starting point. We say “approximately” because we are only using the first few Fourier coefficients to describe the shape of the contour. To find the appropriate number of Fourier coefficients needed for matching we examined the effect of varying the number of low-frequency coefficients we used

from our shape template. Figure 4.19 illustrates the effect of using the first 2, 5, 8, 16, 30, 50, 80 and 100 coefficients from our shape template. Using more Fourier coefficients than necessary leads to increasing computation time with no additional benefit. Adding too many coefficients does not significantly improve the matching performance [168]. Thus, only the first eight Fourier coefficients were used in our experiments.

To decide if a contour extracted from an image corresponds to a hazmat sign we need some way of matching the normalized FD of our shape template and the normalized FD of the extracted contour. Correlation-based matching estimates the cost between two normalized FDs. The cost is defined as

$$e = 2 - 2 \max_l |r_{TI}(l)|, \quad (4.19)$$

where  $|\cdot|$  denotes the complex modulus. Thus we check if the correlation-based matching cost  $e$  between the normalized FD of our shape template and the normalized FD of the extracted contour is below a threshold  $T_e$ . To obtain the value of  $T_e$  we calculate the correlation-based matching cost  $e$  between our shape contour (Figure 4.18) and each of the shape template contours shown in Figure 4.20. Since the cost of matching our shape template against a diamond shape (including rotation) is not greater than 1.75 we set  $T_e = 1.75$ . Note that the shape templates in Figure 4.20 are only used to decide the value of  $T_e$ .

Table 4.3 shows all the parameters/thresholds we used including empirically derived parameters.

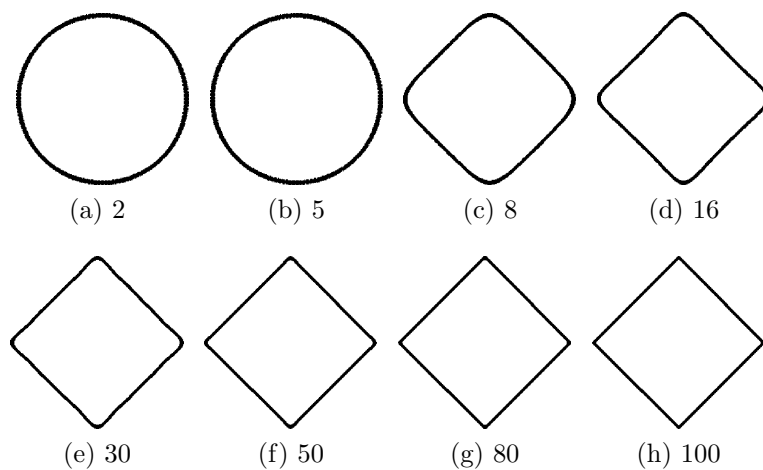


Fig. 4.19.: Reconstruction of the shape template using the first 2, 5, 8, 16, 30, 50, 80 and 100 Fourier coefficients.

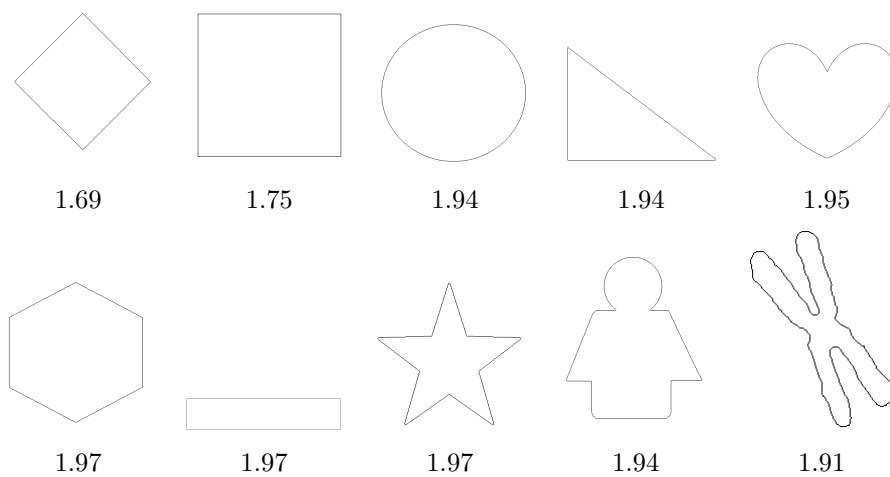


Fig. 4.20.: Comparison of our shape template contour against different shape templates and their matching costs  $e$ .

Table 4.3: Parameters and thresholds used in our proposed method. Automatically determined values are denoted by \*.  $W$  and  $H$  are the width and height of the image.

<b>Parameter</b>	<b>Description</b>	<b>Value</b>
$T_{i_1}$	Low threshold for channel thresholding	*
$T_{i_2}$	High threshold for channel thresholding	*
$T_{i_b}$	Otsu's threshold for binarization	*
$S_d$	Size of structuring element for dilation	5 px
$S_o$	Size of structuring element for opening	20 px
$T_c$	Connected components threshold	$0.03WH$
$T_e$	Correlation-based matching cost threshold	1.75

## 4.5 System Implementation<sup>4</sup>

### 4.5.1 System Overview

We implemented a prototype of the MERGE system as an application for Android and iOS devices and as a web-based interface accessible from any web browser. Figure 4.21 illustrates the MERGE system, which is divided in two groups:

1. **Client-side:** Browse an internal database on the Android device, consisting of the contents of the ERG 2012 Guidebook<sup>5</sup>. Figure 4.22 illustrates the client-side system.
2. **Server-side:** Use image analysis on the server and communicate the results back to the client. Figure 4.23 illustrates the server-side system.

The client-side includes the device and methods available to the users, operating without the use of a network connection. The offline services are only available from Android devices (Section 4.5.3). The online services are available from both Android devices or any web browser (e.g., Internet Explorer, Mozilla Firefox, Google Chrome). This includes desktop and laptop computers as well as Blackberry smartphones (Section 4.5.4). The server-side includes the image analysis process to detect and interpret the hazmat signs.

---

<sup>4</sup>The work presented in this section was done by the author jointly with Andrew W. Haddad.

<sup>5</sup>The internal database was initially created by Andrew W. Haddad and later updated by the author.



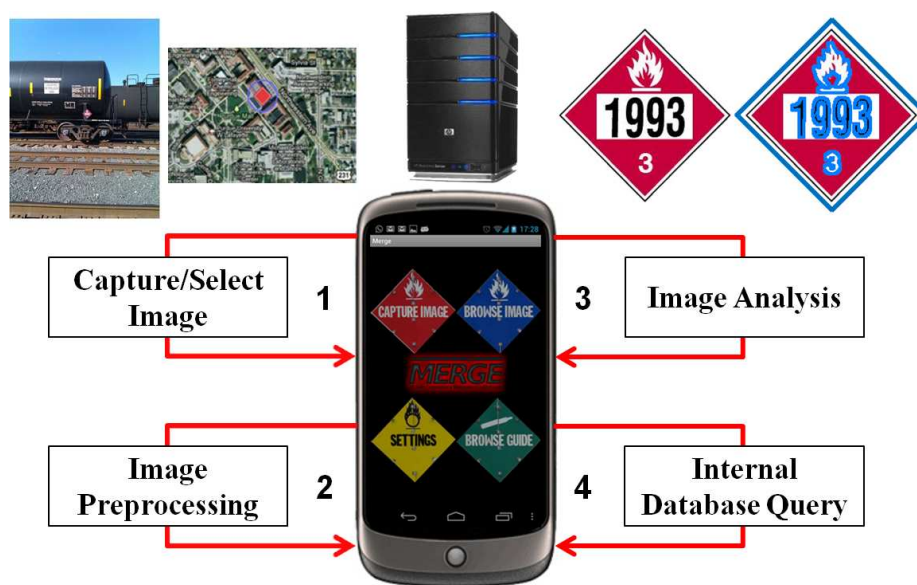


Fig. 4.21.: Mobile-Based Hazmat Sign Detection and Recognition.

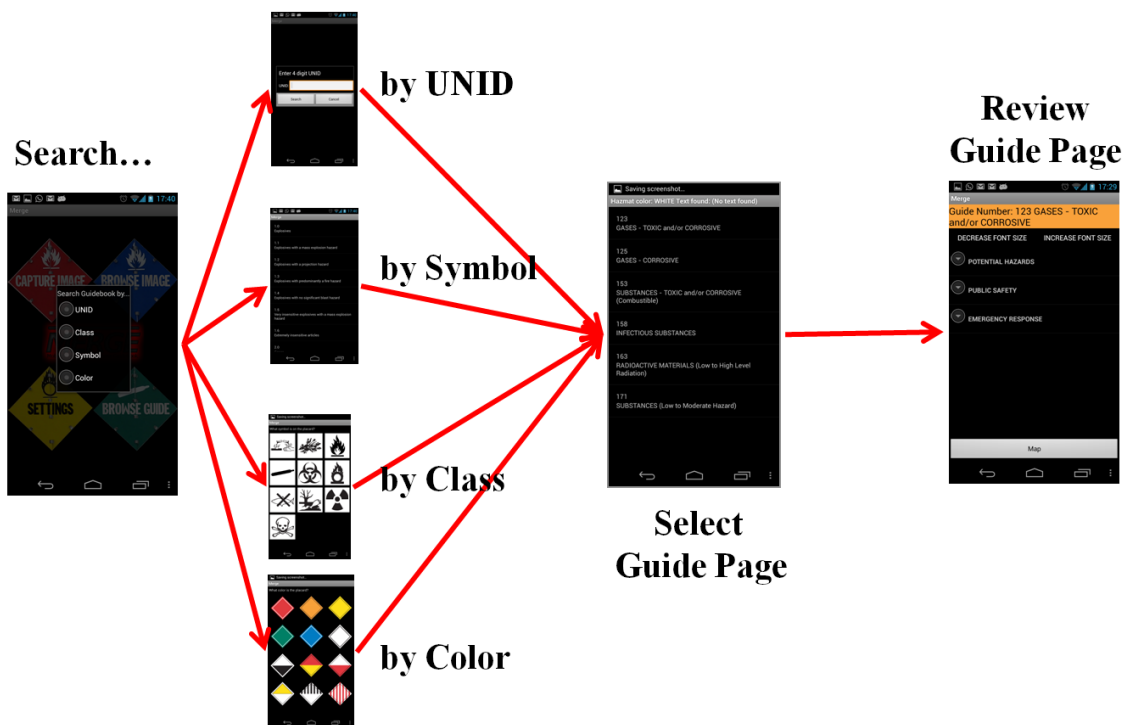


Fig. 4.22.: Overview of the MERGE Client-Side Components.

#### 4.5.2 MERGE Databases

In this section we describe how the image database is organized. We will first describe the database schema and then show by an example how the information GARI acquires is added to the database. The database of hazmat signs was deployed for three reasons:

1. To collect and organize images acquired by first responders. This includes images of hazmat signs, images of scenes for forensic analysis, and metadata.
2. To store the results of the image analysis.
3. To manage first responders' credentials, allowing them to access the services available through the Android/iOS applications and the web based interface.

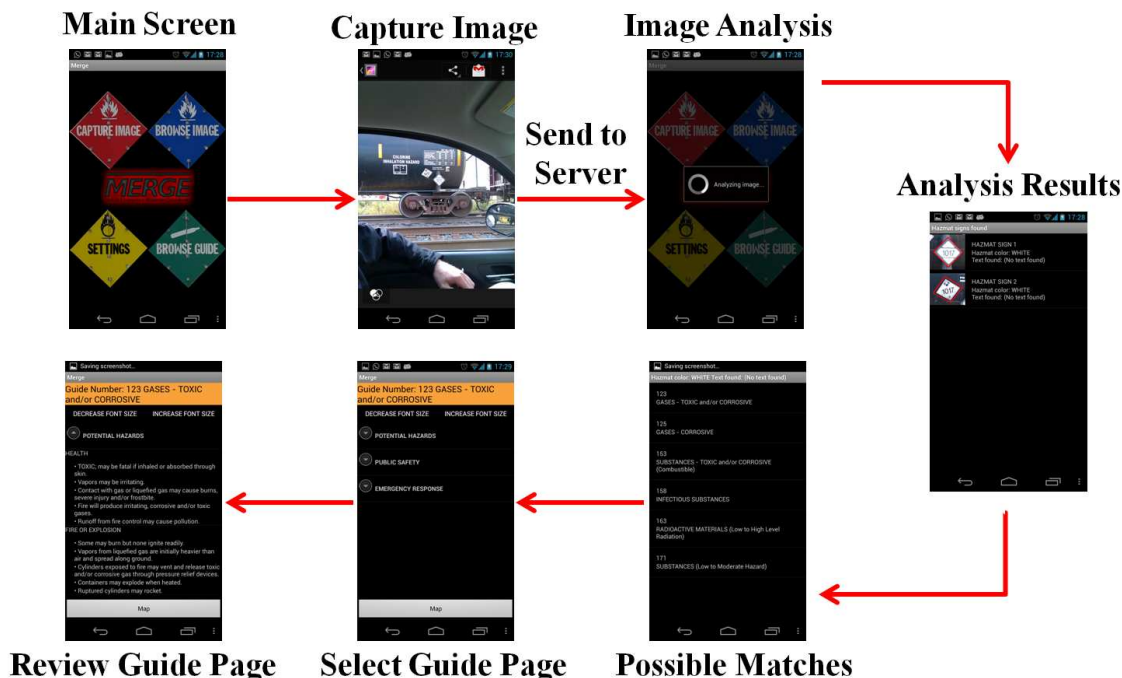


Fig. 4.23.: Overview of the MERGE Server-Side Components.

The MERGE database is implemented in PostgreSQL on a Linux server. It consists of 15 tables, all of them detailed in Appendix E. Figure 3.48 illustrates the structure of the 11 main tables. Note that the schema does not show all the fields in all the tables but just the relevant fields to indicate the association between the tables.

1. **images**: Stores EXIF data from the images along with image location and general image information and the results from the image analysis. The fields related to this table are shown in Tables E.1, E.2 and E.3 in Appendix E.
2. **vw\_01\_orange\_page**: Stores the relationships between guide page numbers, guide pages, categories and details.
3. **vw\_03\_yellow\_page**: Stores the relationships between guide page numbers and UNIDs.

4. **vw\_05\_water\_reactive\_materials**: Stores relationships between UNIDs, dangerous goods and guide page numbers.
5. **placard**: Stores the relationships between UNIDs, placards, symbols and classes.
6. **unids**: Stores the relationships between guide pages, UNIDs and hazardous materials.
7. **class**: Stores information about classes.
8. **colorPages**: Stores the relationships between guide pages and placard colors.
9. **textPages**: Stores information about the text contained in the guide pages.
10. **symbols**: Stores information about the symbols that can appear in hazmat signs.
11. **users**: Stores users' credentials to access to the system services as well as information concerning administrative privileges, email addresses, and registration and login status. Table E.4 in Appendix E describes the fields of this table.

Note that currently we only populate the tables **images** and **users**.

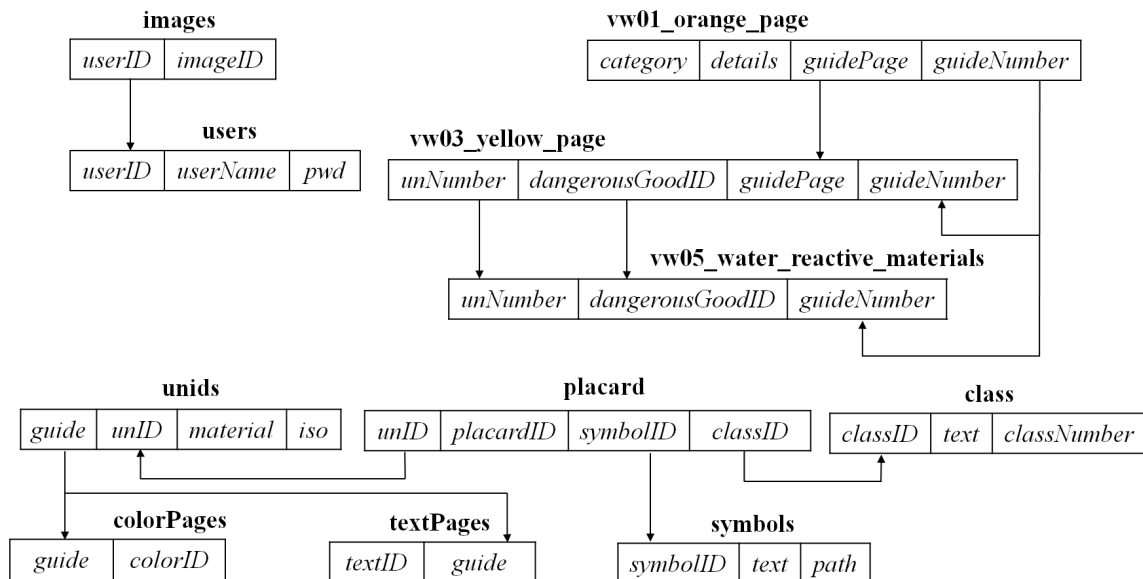


Fig. 4.24.: Database Schema Showing The Associations Between the Tables in the Database.

### 4.5.3 Android/iOS Implementation

We implemented the MERGE system on Android and iOS devices as summarized in Figures 4.22 and 4.23. We called this application Mobile MERGE. In this section we describe how the application works and describe its user interface.

#### Overview

A user takes an image of the scene containing one or multiple hazmat signs using the embedded camera on the device via the Graphical User Interface (GUI). The EXIF data of the image, including GPS location and date and time of capture, is automatically added to the image header. The image is then automatically sent to the server for analysis. The results are sent back to the user, and links to an internal database are provided. The internal database is a digitized version of the 2012 ERG.

Another option is to browse the internal database. The user can search for information about hazmat signs by UNID, symbol, class, or color. Each of the options provide links to the guide pages containing information to determine what specialty equipment, procedures and precautions should be taken in the event of an emergency.

We implemented the system on different smartphones makes and models, but always targeting version 3.0 of the Android operating system (OS).

## **User Interface<sup>6</sup>**

Our Android application does not require the use of a network connection. However it is mandatory if the user wants to update the application or analyze an image. The application automatically checks for updates when launched, notifying the user if a new version is available (Figure 4.25). A user must be assigned a User ID and a unique password in order to use MERGE. Once the User ID and password has been entered, the main screen is shown (Figure 4.26). The main screen includes the following options, which are described below:

- Browse Image
- Browse Guide Pages
- Capture Image
- Settings
- About

Note that the “About” option appears when the user presses the menu button.

### **Browse Image**

The user has the option to browse images stored on the Android device, instead of taking an image using the “Capture Image” option. Note that the entire phone image gallery is browsed, including images that have not been taken using the Mobile

---

<sup>6</sup>The user interface was initially created by Andrew W. Haddad, and later updated by the author.

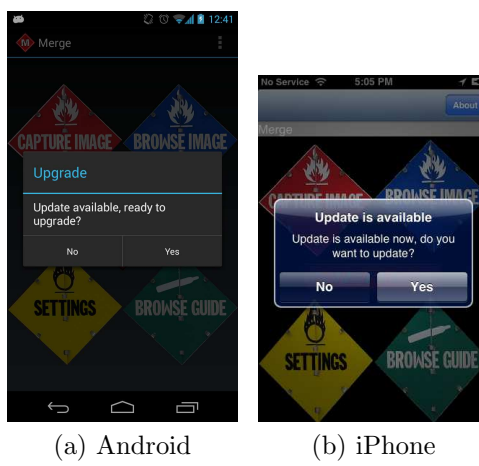


Fig. 4.25.: Automatic updates.

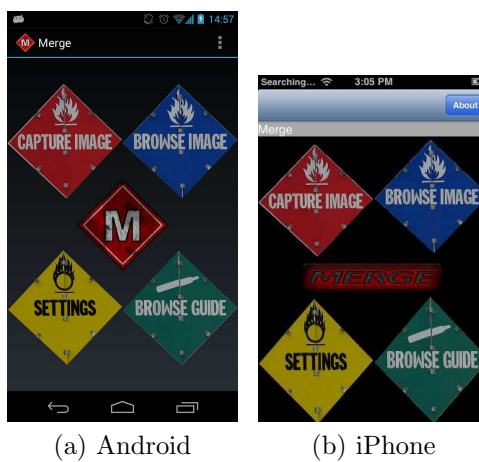


Fig. 4.26.: Main Screen.

MERGE application. When the option “Browse Image” is tapped, a directory browsing window is opened, and the user can search and select the desired image. Figure 4.27 shows an example of browsing. Once the image is selected, it is automatically sent to the server for analysis.

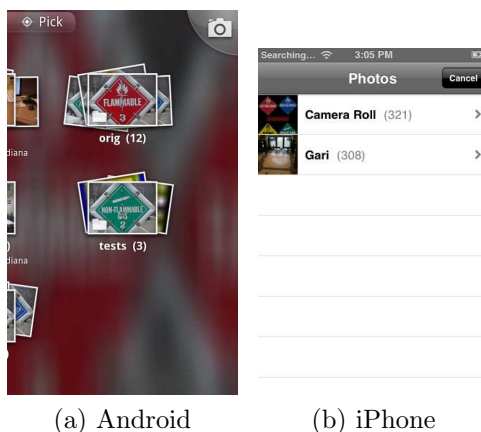


Fig. 4.27.: Screens for browsing images.

### Browse Guide Pages

When the user chooses to search for a guide page, they are presented with a dialog containing four different ways to search the database, depending on what information is available to the user. The four options are shown in Figure 4.28:

#### 1. UNID

The four-digit UNID number should be one of the UNIDs found in the 2012 emergency response guidebook. The valid range for guide pages is 1001-9279. Numbers outside this range will produce an error indicating the proper range.

#### 2. Class

Each class produces a list of pages or a single guide page pertaining to the particular class selected. In many cases, the list cannot be narrowed automatically and the decision is left to the user. The possible classes are: Explosives, Gases, Flammable Liquids, Flammable Solids, Oxidizing Substances, Toxic Substances, Corrosive Substances, and Miscellaneous Hazardous Materials.



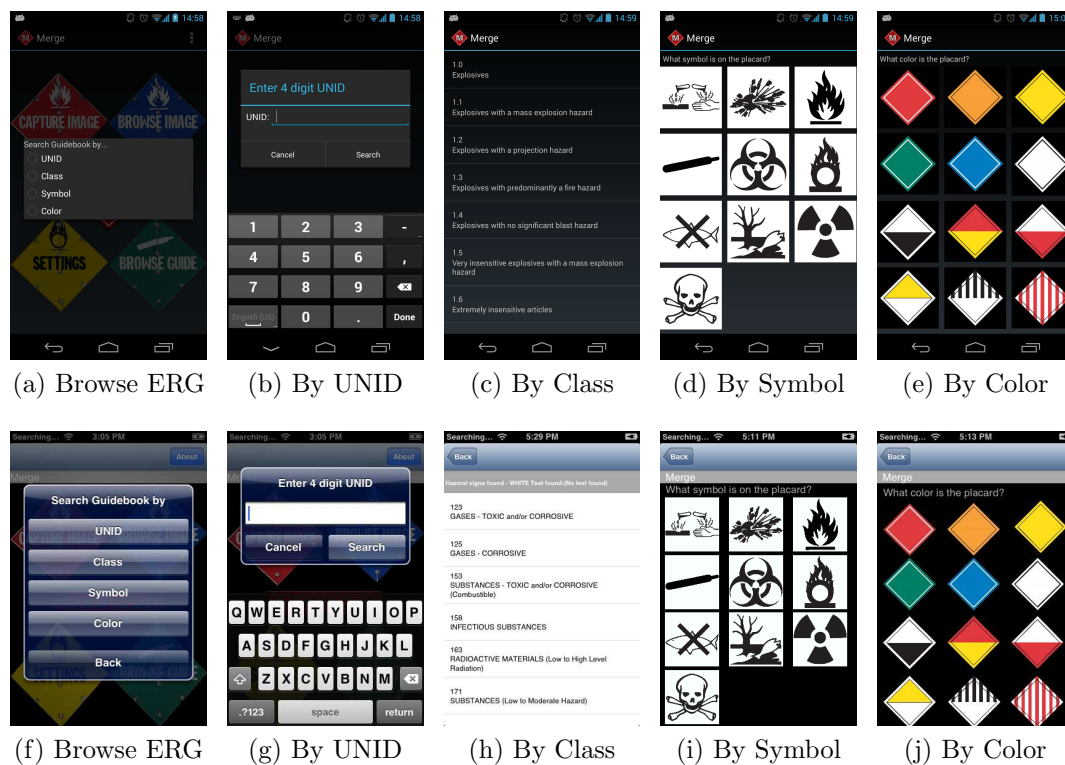


Fig. 4.28.: Methods for browsing. Android (top) and iPhone (bottom).

### 3. Symbol

Though symbols are often related to the guide pages similarly to the classes, they do not always match. Often we have multiple symbols per class and multiple classes per symbol. Similarly to classes, each symbol produces a list of pages or a single guide page pertaining to the particular symbol selected. In many cases, the list cannot be narrowed automatically and the decision is left to the user. The possible symbols are: Corrosive, Explosive, Flammable, Gases, Infectious, Oxidizing, Pollutant, Radioactive, and Toxic.

### 4. Color

Each color or combination of colors represents a number of guide pages. Similarly to classes and symbols, each color produces a list of pages or a single guide page pertaining to the particular color or combination of colors selected. In many cases, the list cannot be narrowed automatically and the decision is left to the user. The possible colors and combinations of colors are: Red, Orange, Yellow, Green, Blue, White, White and Black, Red and Yellow, White and Red, Yellow and White, White and Black Stripes, White and Red Stripes.

### 5. Guide Page

The ERG contains a section where the general hazards of the dangerous goods are covered (orange-bordered pages, also known as guides). Each guide is divided into three main sections: potential hazards, public safety, and emergency response (Figure 4.29a). The guides in Mobile MERGE are organized in the same fashion as in the ERG, but using expandable lists. That is, the user can search for a specific guide page and tap on any of the three sections to read all the information available (Figure 4.29c).

#### (a) Page Number

The first thing the user will notice, at the top most of the orange header, is the Guide Page number. This is made available so the user can cross-

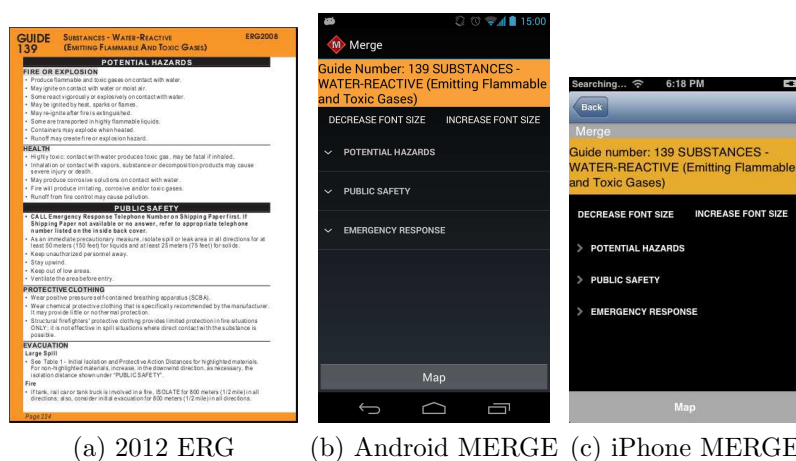


Fig. 4.29.: Guide page in the ERG 2012 and corresponding guide page in Mobile MERGE for Android (middle) and iPhone (right).

reference the information provided by MERGE with the Emergency Response Guidebook (ERG) 2012 if necessary.

(b) Substance

Next, also in the header, the user will see the substance name/category. E.g. Oxidizers.

(c) Categories

As previously stated, the page is separated into categories, subcategories, and details. The headers for possible categories are: Potential Hazards, Public Safety, Emergency Response, Supplemental Information.

(d) Map

If a green table entry is available for a given guide page, the user will be presented with the option of displaying a map with a recommended evacuation region defined according to the current location of the user and the chemical chosen. Figure 4.30) shows the steps followed to obtain the evacuation region. The user will be asked up to three questions to better define the evacuation region: “Large of Small Spill?”, “Initial Isolation or Protective Action?” and “Is it Day or Night?”. After the user answers these questions, a map is displayed. The map will always contain a circle shape indicating the evacuation region, and for some available chemicals it will also contain a plume model, as seen in Figure 4.30h. The plume shape is obtained by querying database of real-time weather information, which provides more accurate evacuation information using wind speed and direction at the current location.

### **Capture Image**

If the user taps the “Capture Image” button from the main screen an image can be acquired. The camera interface, shown in Figure 4.31, allows the user to take an image of a hazmat sign to be analyzed (“SIGN”) or an image of the scene for future forensic analysis (“SCENE”).



Fig. 4.30.: Evacuation region for Android (top) and iPhone (bottom). From left to right, questions asked to refine evacuation region, and general evacuation circle and weather-based plume model.

The image is automatically sent to the server, where it is be stored and analyzed. The user is notified through a dialog that the image upload and analysis is taking place. The analysis is done only when the image is captured using the “SIGN” option. After the analysis, the user will be presented with options to determine correctness and the closest matching guide page associated with the captured placard.

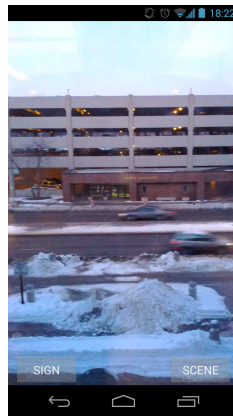


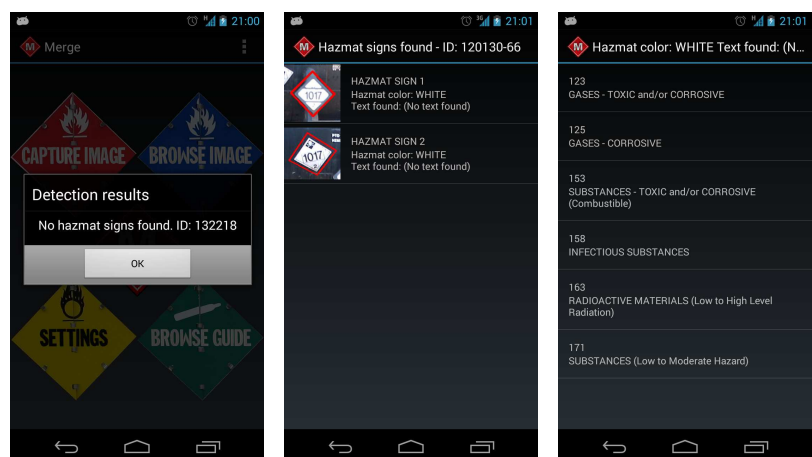
Fig. 4.31.: Camera Interface with “SIGN” and “SCENE” options.

When the image analysis is completed, the results are shown to the user. There are two possible scenarios.

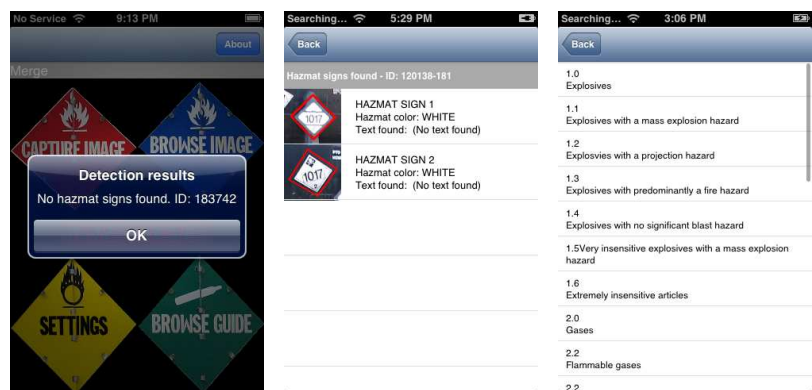
1. If no placard has been found a dialog informs the user (Figure 4.32d).
2. If the system has successfully determined which placard the image corresponds to, it will show a list with the results (Figure 4.32e). If more than one result is possible for a particular placard (e.g., if the placard color is found but not the text) a list of all the associated guide pages are shown (Figure 4.32f); otherwise, a single guide page is shown (Figure 4.29c).

## Security

Our Android application is used by first responders from multiple agencies. Therefore, it is mandatory to ensure that only authorized users can access and use the



(a) No Placard Found (b) Results of Analysis (c) Possible Guide Pages



(d) No Placard Found (e) Results of Analysis (f) Possible Guide Pages

Fig. 4.32.: Results of the Image Analysis Process. Android (top) and iPhone (bottom)

application. The connections to the server must be secure and all the information transmitted to and from the server must be encrypted (using the SSL/TLS protocol). The user credentials are sent every time the application contacts the server to make sure the connection is made by an authorized user. In the Android version we use ProGuard [246], a code optimizer and obfuscator for the Android SDK. It reduces the application size up to 70% and makes the source code more difficult to reverse engineer. It also improves the battery life by removing verbose logging code in a background service. An additional level of security includes the creation of two types of users:

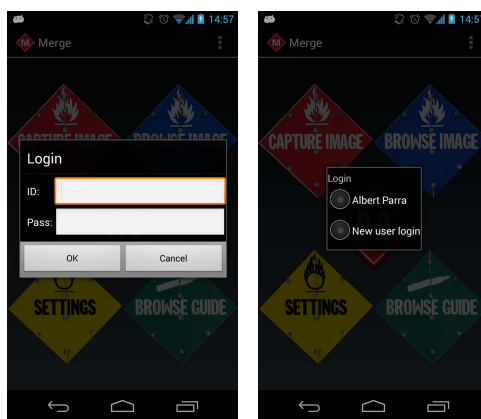
- Regular users: Can switch between users, change their password, delete specific images only taken by themselves, and send crashlogs to the server.
- Administrative users: Can modify the server domain name/IP address, change user IDs, change passwords, delete specific images from any user, delete all images of any specific user, and send crashlogs to the server.

When launching the MERGE application, a dialog box prompts the user (Figure 4.33a). The user ID and a password is entered. If this is the first time the user logs in, a new dialog box prompts the user to change the default password (provided by the MERGE staff by email). For successive logins, the user will appear on a list of previously logged users, and no password is necessary (Figure 4.33b).

All authorized users can access the “Settings” option from the main screen of the application. Figure 3.73 shows the various options.

- Server Location: Administrative users can change the server IP address (Figure 4.34f).
- Change Login: The change user dialog is exactly the same as the login dialog, with the exception that if someone chooses to change the user for the application, they can cancel the change before submitting (Figure 4.34g).

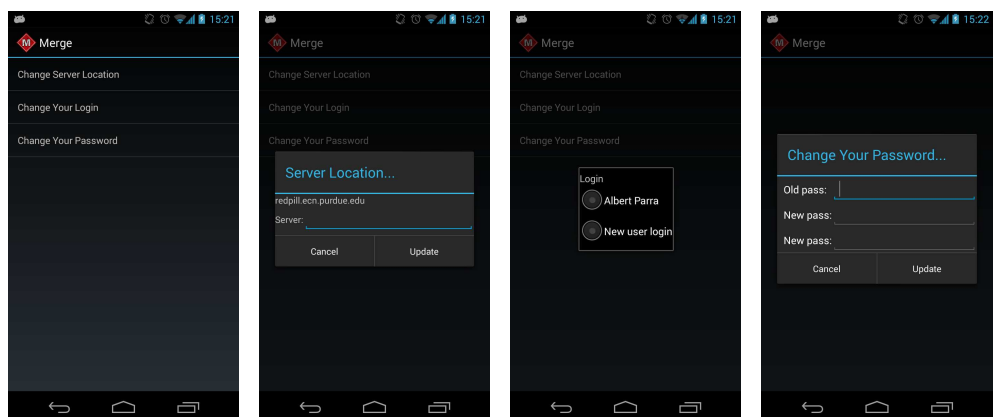




(a) First time login (b) Returning user

Fig. 4.33.: User ID Screen.

- **Change Password:** The change password dialog is exactly the same as the change password dialog, which appears immediately after the first login - on either the website or the mobile app - with the exception that if someone chooses to change the password for the current user of the application, they can cancel the change before submitting (Figure 4.34h).



(a) Settings Menu (b) Change Server (c) Change User (d) Change Password



(e) Settings Menu (f) Change Server (g) Change User (h) Change Password

Fig. 4.34.: Settings Menu Options. Android (top) and iPhone (bottom).

#### 4.5.4 Web Interface<sup>7</sup>

##### System Overview

We also implemented our system as a web interface that gives the user access to the hazmat database, and provides the ability to upload and browse images, and browse the official guidebook. We called this application Desktop MERGE. The web interface is available from any device with a web browser. This includes all desktop and laptop machines and all mobile telephones capable of browsing the web (e.g., iPhone, Blackberry, Android devices).

##### User Interface

As of March 2014 the MERGE website is located at [www.hazmat-signs.org](http://www.hazmat-signs.org). The main page contains information about the MERGE project, its principal investigators, and the graduate students involved. The “Internal” page (Figure 4.35) displays the options the user has to interact with the graffiti database, including Browse Guidebook and Browse Images.

##### Browse Guidebook

Users can browse the guidebook using four different methods (Figure 4.36). The intersection of the sets created by the Color, Symbol and Classes chosen will be returned as a list of guide pages. Given more information, users can combine colors with symbols and classes. This will produce a smaller list of placards, containing all of the characteristics added. When a user searches by UNID, it takes preference over the other fields. That is, if UNID is searched, the Color, Symbol and Class fields are ignored.

The list of results contains images representing the color and symbol and shows the class searched. Each entry in the list of results contains the Guide Page number and Guide Page name (Figure 4.37).

---

<sup>7</sup>The work presented in this section was done by Andrew W. Haddad.

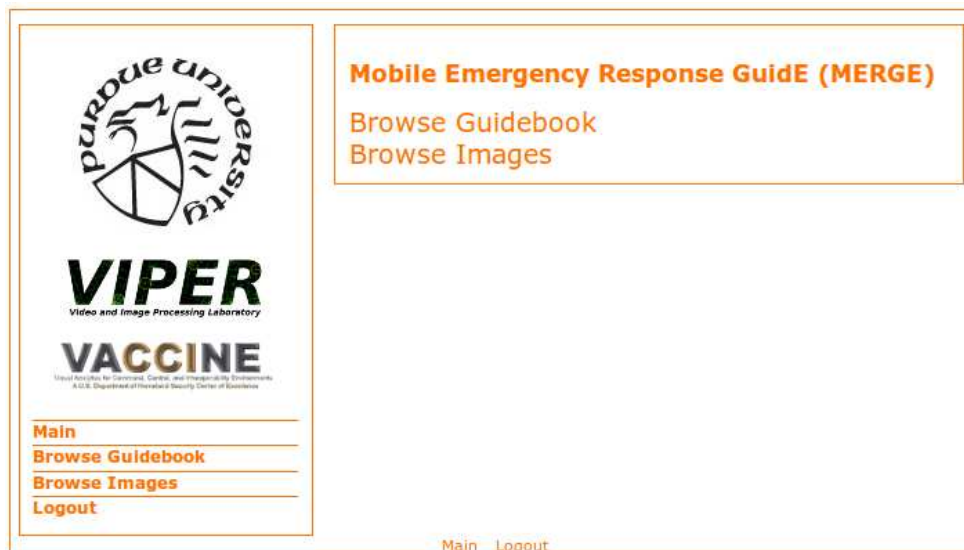


Fig. 4.35.: “Internal” Section of Desktop MERGE.



Fig. 4.36.: Search Guidebook Pages by Color, Symbol, Class, or UNID



**VIPER**  
Video and Image Processing Laboratory

**VACCINE**  
Virtual Analysis for Terrorism, Crime, and Investigative Environments  
A U.S. Department of Homeland Security Center of Excellence

---

[Main](#)

---

[Browse Guidebook](#)

---

[Browse Images](#)

---

[Logout](#)

### Mobile Emergency Response Guide (MERGE)

Back

**Color:**

**Symbol:**

**Class:**

**UNID:**  (takes preference over other fields)




**Class: 1**

Page Number: 112

Page Name: EXPLOSIVES\* - DIVISION 1.1, 1.2, 1.3, 1.5 OR 1.6; CLASS A OR B

---

[Main](#) [Logout](#)

Fig. 4.37.: Browse Guidebook Page Results

The Guide Page in MERGE is very similar to the Guide Page in the Emergency Response Guidebook. It contains the Guide Page Number, Guide Page Name, Categories (Potential Hazards, Public Safety, Emergency Response, and Supplemental Information), Sub-categories (Fire or Explosion, Health, Protective Clothing, Evacuation, Spill or Leak, First Aid, etc) and each sub-category contains a bulleted list of details (Figure 4.38).

Guide 112	EXPLOSIVES* - DIVISION 1.1, 1.2, 1.3, 1.5 OR 1.6; CLASS A OR B	ERG 2008
<b>POTENTIAL HAZARDS</b>		
<b>FIRE OR EXPLOSION</b>		
<ul style="list-style-type: none"> <li>• MAY EXPLODE AND THROW FRAGMENTS 1600 meters (1 MILE) OR MORE IF FIRE REACHES CARGO.</li> <li>• For information on "Compatibility Group" letters, refer to Glossary section.</li> </ul>		
<b>HEALTH</b>		
<ul style="list-style-type: none"> <li>• Fire may produce irritating, corrosive and/or toxic gases.</li> </ul>		
<b>PUBLIC SAFETY</b>		
<ul style="list-style-type: none"> <li>• CALL Emergency Response Telephone Number on Shipping Paper first. If Shipping Paper not available or no answer, refer to appropriate telephone number listed on the inside back cover.</li> <li>• Isolate spill or leak area immediately for at least 500 meters (1/3 mile) in all directions.</li> <li>• Move people out of line of sight of the scene and away from windows.</li> <li>• Keep unauthorized personnel away.</li> <li>• Stay upwind.</li> <li>• Ventilate closed spaces before entering.</li> </ul>		
<b>PROTECTIVE CLOTHING</b>		
<ul style="list-style-type: none"> <li>• Wear positive pressure self-contained breathing apparatus (SCBA).</li> <li>• Structural firefighters' protective clothing will only provide limited protection.</li> </ul>		
<b>EVACUATION</b>		
<ul style="list-style-type: none"> <li>• Consider initial evacuation for 800 meters (1/2 mile) in all directions.</li> </ul>		

Fig. 4.38.: View Guidebook Page

## Browse Images

Administrative users can browse images that have been uploaded (Figure 4.39). The images are listed along with the user who uploaded the image and the date and time they were taken. The user can choose between browsing the images containing signs (Signs) or the scene images uploaded for forensic analysis (Scene).



**VIPER**  
Video and Image Processing Laboratory

**VACCINE**  
Virtual Analysis for Chemical, Combat, and Homeland Security  
A U.S. Department of Homeland Security Center of Excellence

---

[Main](#)

---

[Browse Guidebook](#)

---

[Browse Images](#)

---

[Logout](#)

**Mobile Emergency Response Guide (MERGE)**

Signs | Scene

TOTAL IMAGES: 216



Uploaded By: Edward Hertelendy  
Date/Time: 2013-08-30 13:19:06

---



Fig. 4.39.: Browse Images



## 5. EXPERIMENTAL RESULTS

All the experiments in this section were done using a Samsung Galaxy Nexus mobile device with a dual-core 1.2GHz CPU and 1GB RAM for the client tasks, and a desktop computer with a quad-core 3.2GHz CPU and 32GB RAM for the server tasks.

### 5.1 GARI

#### 5.1.1 RGB to Y'CH Conversion

In Section 3.4 and Appendix A we describe two approaches to transform the RGB color space to our Y'CH color space. The first, which we called *arithmetic approach*, converts RGB to Y'CH by only doing arithmetic operations. The second, which we called *trigonometric approach*, converts RGB to YIQ color space as an intermediate step, and then to Y'CH, using arithmetic and trigonometric operations. As a reminder, Equation 5.1 shows the mathematical definition of the arithmetic approach and Equation 5.2 shows the mathematical definition of the trigonometric approach. Note that Equation 5.2 does not define the transformation RGB to YIQ, since it is a linear transformation, it will not have an influence on the execution time of the overall transformation RGB to Y'CH.

$$\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B \\
C &= \max(R, G, B) - \min(R, G, B) \\
&= M - n \\
H &= \begin{cases} 60\left(\frac{G-B}{C}\right) & \text{if } M=R \\ 60\left(\frac{B-R}{C} + 2\right) & \text{if } M=G \\ 60\left(\frac{R-G}{C} + 4\right) & \text{if } M=B \\ \text{undefined} & \text{if } C=0 \end{cases} \quad (5.1)
\end{aligned}$$

$$\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B \\
C &= \sqrt{I^2 + Q^2} \\
H &= \begin{cases} \arctan\left(\frac{Q}{I}\right) & I > 0 \\ \pi + \arctan\left(\frac{Q}{I}\right) & Q \geq 0, I < 0 \\ -\pi + \arctan\left(\frac{Q}{I}\right) & Q < 0, I < 0 \\ \frac{\pi}{2} & Q > 0, I = 0 \\ -\frac{\pi}{2} & Q < 0, I = 0 \\ \text{undefined} & Q = 0, I = 0 \end{cases} \quad (5.2)
\end{aligned}$$

Given that trigonometric operations are computationally more complex than arithmetic operations [304], we could assume that the arithmetic approach is always computationally faster than the trigonometric approach. However, we conducted tests to verify this. Table 5.1 and Figure 5.1 show the results of both transformations using various number of data points on the HTC Desire. Note that each data point corresponds to a pixel operation. Also note that the functions used to compute the time differential both on the hand-held device are accurate to the nearest millisecond. One can see how the execution time of the trigonometric approach grows exponentially faster than the arithmetic approach when the number of data points is greater than

approximately one million. For example, for a five megapixel image (i.e., five million data points) the difference between the arithmetic approach and the trigonometric approach can be linearly interpolated to 3.36 seconds. Since the RGB to Y'CH conversion is done not only along a traced path during the color recognition process, but also on entire images during the image segmentation process, it is worth considering the arithmetic approach as a lightweight and fast approach if we plan on doing color image segmentation on the device in the future.

Table 5.1: Execution Time (seconds) of the Arithmetic and the Trigonometric Approaches For Color Conversion.

Data Points	Execution Time	
	Arithmetic	Trigonometric
100	0	0
1,000	0.002	0.004
10,000	0.010	0.010
100,000	0.02	0.10
1 million	0.20	0.96
10 million	1.91	9.39
100 million	18.37	91.85
1 billion	183	922

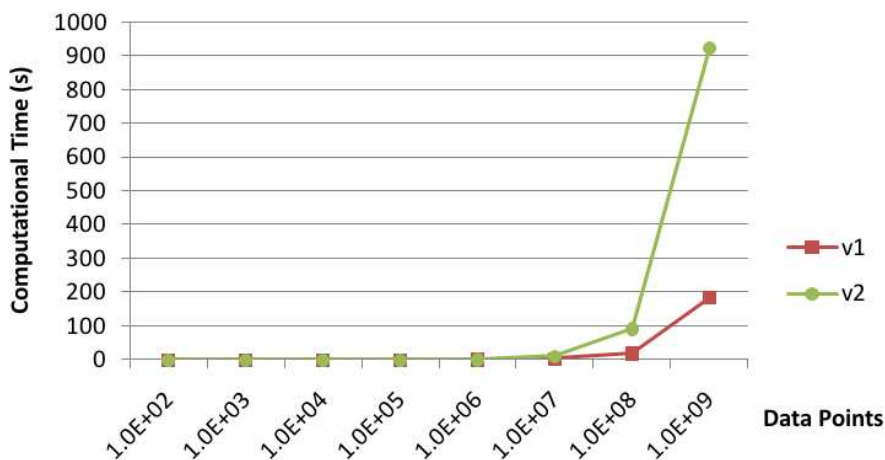


Fig. 5.1.: Execution Time with Respect to the Number of Data Points for the Arithmetic and the Trigonometric Approaches For Color Conversion.

### 5.1.2 Color Correction Based on Mobile Light Sensor<sup>1</sup>

To evaluate the performance of our proposed Color Correction Based on Mobile Light Sensor we did an experiment in 3 different scenarios: 1) using a fiducial marker in every image (M1), 2) using a fiducial marker every week (M2), 3) using the mobile light sensor values (M3). Using a fiducial marker every week means taking an image of the fiducial marker under daylight conditions to create a color correction matrix, and using this matrix on every image taken in the following week. For scenario M3 4,916 images were acquired during a period of three weeks during August of 2013, using a 5Mpx camera on a Samsung Galaxy Nexus mobile device, to obtain 612 unique lux values. Figure 5.2 illustrates the distribution of lux values for each lightning step.

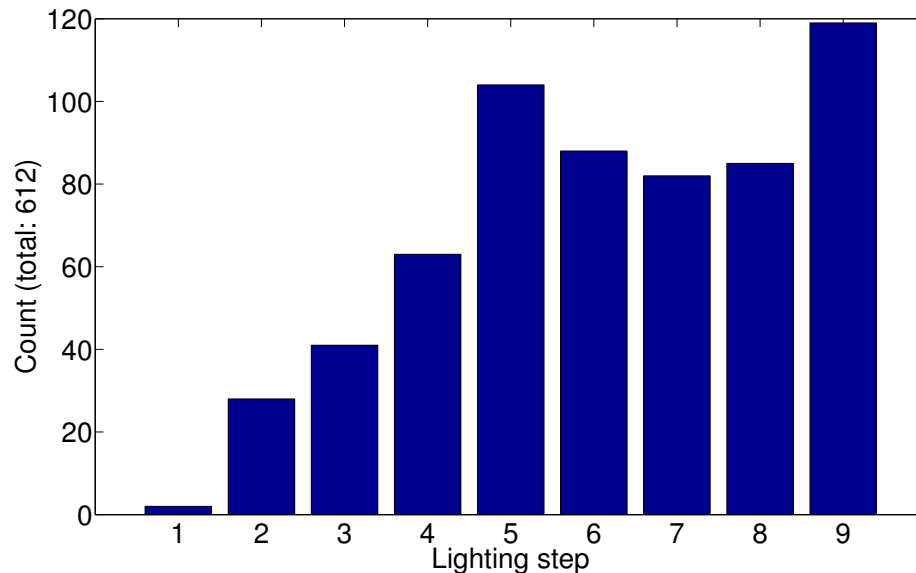


Fig. 5.2.: Distribution of Lux Values for Each Lightning Step.

For each scenario we computed 3 different color correction matrices to map colors under an unknown lighting condition and a D65 reference lighting condition: 1) CIELab based mapping ( $M_{GT \rightarrow D65}^{Lab}$ ), 2) Linear-RGB mapping ( $M_{GT \rightarrow D65}^{RGB}$ ), and 3) Polynomial-RGB mapping ( $M_{GT \rightarrow D65}^{RGBPOL}$ ). The mapping  $M_{GT \rightarrow D65}^{Lab}$  is described in

<sup>1</sup>The work presented in this section is partly based on the work by Chang Xu on color correction.

Section 3.3. The mappings  $M_{GT \rightarrow D65}^{RGB}$  and  $M_{GT \rightarrow D65}^{RGBPOL}$  are obtained following the description from [61]:

$$M_{GT \rightarrow D65}^{Lab} = \operatorname{argmin}_{M_{3 \times 3}} \sum_{i=1}^{11} \left\| (Lab_i)_{D65}^T - M_{3 \times 3} (Lab_i)_{GT}^T \right\| \quad (5.3)$$

$$M_{GT \rightarrow D65}^{RGB} = \operatorname{argmin}_{M_{3 \times 3}} \sum_{i=1}^{11} \left\| (RGB_i)_{D65}^T - M_{3 \times 3} (RGB_i)_{GT}^T \right\| \quad (5.4)$$

$$M_{GT \rightarrow D65}^{RGBPOL} = \operatorname{argmin}_{M_{3 \times 10}} \sum_{i=1}^{11} \left\| (RGB_i)_{D65}^T - M_{3 \times 10} P_{10 \times 11} \right\|, \quad (5.5)$$

where

$$P_{10 \times 11} = [R_{GT} \ G_{GT} \ B_{GT} \ R_{GT}^2 \ G_{GT}^2 \ B_{GT}^2 \ R_{GT}B_{GT} \ R_{GT}G_{GT} \ G_{GT}B_{GT} \ 1]^T. \quad (5.6)$$

For this experiment we acquired 200 images during a period of 3 days during March of 2014 using a 8Mpx camera on a LG Nexus 5 mobile device. Each image contained the fiducial marker already introduced in Section 3.3 and a GrentagMacbeth Colorchecker [305], which is a calibrated color reference chart. Figure 5.3 shows both markers. The fiducial marker was used to obtain the color correction matrices in M1 and M2, and the GrentagMacbeth Colorchecker was used to compute the differences between the original image and the corrected images.

Each image was color corrected using the 3 mappings under each of the 3 scenarios for a total of 9 different color corrections. Figure 5.4 shows an example of color correction for each mapping. For each color corrected image we obtained the mean RGB channel errors  $\Delta$  by calculating the Euclidean distances of the average color of each color patch in the GrentagMacbeth Colorchecker between the color corrected marker ( $RGB_{corr}$ ) and the known reference marker under D65 illumination ( $RGB_{D65}$ ). That is,

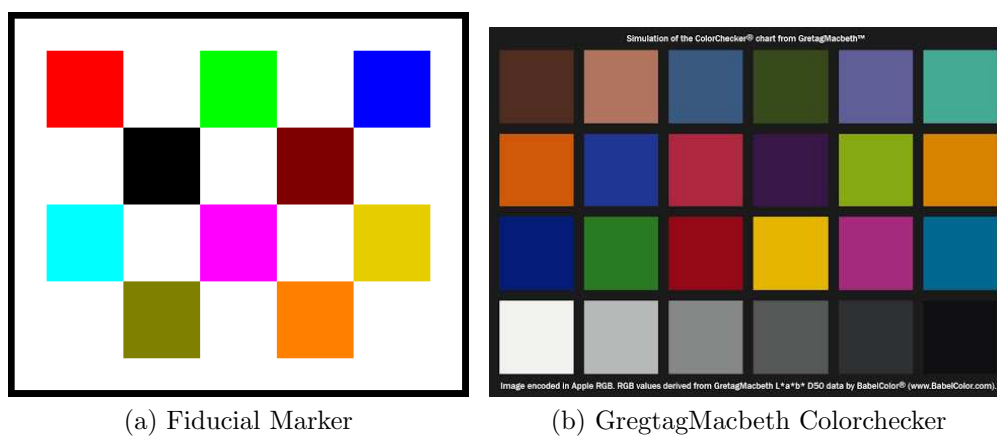


Fig. 5.3.: Fiducial Marker (left) and GretagMacbeth Colorchecker (right).

$$\Delta = \frac{1}{24} \sum_{i=1}^{24} \left\| (RGB_i)_{corr}^T - (RGB_i)_{D65}^T \right\|. \quad (5.7)$$



(a) Original Image. Lux: 2219



(b) M1 Lab



(c) M1 RGB



(d) M1 RGB POL



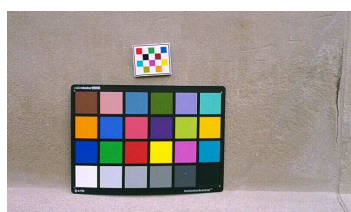
(e) M2 Lab



(f) M2 RGB



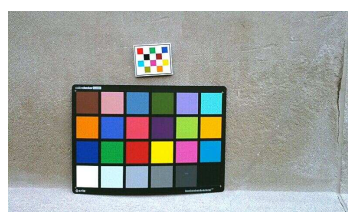
(g) M2 RGB POL



(h) M3 Lab



(i) M3 RGB



(j) M3 RGB POL

Fig. 5.4.: Color Correction Example Under Each Scenario and Each Mapping. M1: using a fiducial marker in every image, M2: using a fiducial marker every week, M3: using the mobile light sensor value.

Table 5.2 shows the mean RGB channel errors ( $\Delta$ ) and running times for each scenario (M1, M2, M3) and each mapping (Lab, RGB, RGB POL), including individual

errors in the R, G, and B color channels. We also include the  $\Delta$  of image before correction for comparison. Figures 5.5 and 5.6 illustrate the RGB results in bar graphs. Note that since the errors are computed in the RGB color space, the Lab corrected images are transformed back to RGB. The time spent on this transformation is not taken into account in the running time.

Table 5.2: Mean Channel Errors ( $\Delta$ ) and Average Running Times (seconds) For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL).

	<b>RGB</b>	<b>R</b>	<b>G</b>	<b>B</b>	<b>Time</b>
<b>Before</b>	14.06	6.90	6.10	8.35	-
<b>M1 Lab</b>	8.55	2.92	5.06	5.11	1.81
<b>M1 RGB</b>	11.99	4.49	4.97	8.07	1.11
<b>M1 RGB POL</b>	8.73	3.44	4.07	5.26	1.33
<b>M2 Lab</b>	12.72	6.02	5.97	7.18	1.80
<b>M2 RGB</b>	13.96	6.03	5.65	9.04	1.07
<b>M2 RGB POL</b>	12.18	5.84	5.04	6.82	1.31
<b>M3 Lab</b>	10.88	4.99	5.62	6.00	1.76
<b>M3 RGB</b>	13.27	5.63	5.33	8.59	1.05
<b>M3 RGB POL</b>	10.88	5.17	4.75	6.30	1.27

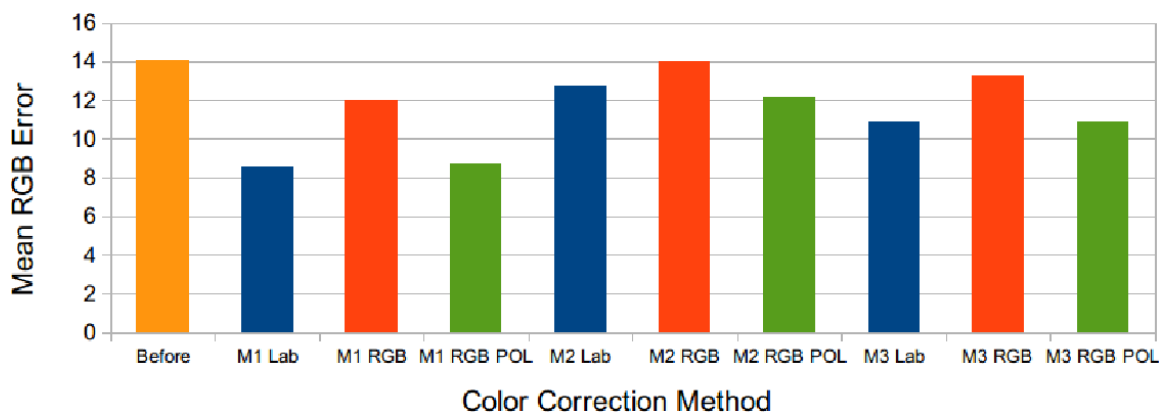


Fig. 5.5.: Mean Channel Errors ( $\Delta$ ) For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL).

The Lab color correction method always gives the best results, at the expense of a small increase on the computational time. As expected, color correcting an image



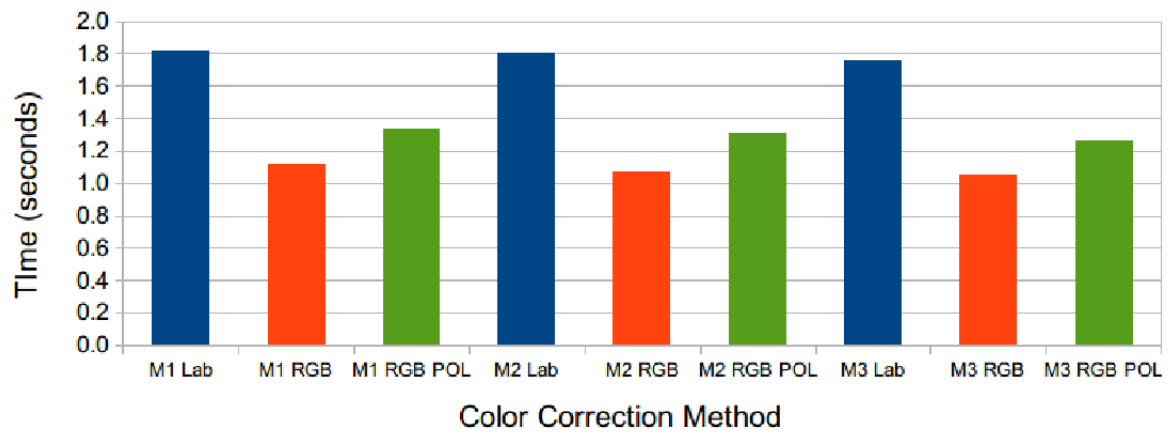


Fig. 5.6.: Average Running Times For Each Scenario (M1, M2, M3) and Mapping (Lab, RGB, RGB POL).

using always a fiducial marker produces the best results ( $\Delta = 8.55$ ). However, the color correction based on the mobile light sensor produces better results than using a fiducial marker once a week ( $\Delta = 10.88$  and  $\Delta = 12.72$  respectively). Also, when using light sensor values we do not have to compute the color correction matrix for each image, thus being the fastest of the three scenarios.

### 5.1.3 Content Based Image Retrieval<sup>2</sup>

We did two experiments to determine the accuracy and the speed of our image retrieval approach.

The goal of the first experiment was to match query images to images in our database based on the scene. We call this process “Gang Graffiti Scene Recognition”. That is, by finding features not only from the graffiti in the image, but also of the background. We trained 1,329 images from our database to extract a total of 633,764 SIFT descriptors (an average of 477 descriptor per image), and used hierarchical  $k$ -means to create a vocabulary tree. Figure 5.7 shows some samples from the training dataset. A separate set of 156 images was used for testing. Both training and testing images were acquired using multiple cameras with different resolutions, at different distances, and lighting conditions over a period of 3 years.

Each of the test images corresponded to one of the scenes in our database, but under different viewpoint, rotation, and illumination, and using different camera makes and models. Figure 5.8 shows some samples from the testing dataset. For each test image we retrieved its 5 closest matches from the training set and we gave it a score from 5 to 0, 5 meaning that the matching image was ranked in first position and 0 meaning that there was no match in the top 5 results. We called this scoring method “weighted top-5 accuracy”.

Table 5.3 summarizes the results of the first experiment using different combinations of  $k$  and  $n_w$  in the range  $k \in [2 \dots 1,000]$  and  $n_w \in [100 \dots 1,000,000]$ . Table

---

<sup>2</sup>The work presented in this section was done by the author in cooperation with Bin Zhao and Joonsoo Kim.



Fig. 5.7.: Samples from Training Dataset.



Fig. 5.8.: Samples Image Matches. Left: Training Images (Samsung Galaxy Nexus). Right: Matching Testing Images (Casio PowerShot S95).

5.4 shows the Top-1 accuracies for the same ranges of  $k$  and  $n_w$ . Tables 5.5 and 5.6 show the average training and query times. Figures 5.9 to 5.11 illustrate the same information using color maps. Even though the retrieval accuracy increases with the number of leaves, the query time is directly related to the number of nodes and levels in the vocabulary tree, as shown in Figures 5.12 and 5.13. A wise choice for  $k$  and  $n_w$  would then take into account both the accuracy and the query time (not the training time, since it does not affect the real time retrieval). For  $k = 3$  and  $n_w = 10,000$  we obtain a retrieval accuracy of 99.10% with a Top-1 accuracy of 96.15% and an average query time of 70 ms. As a comparison, using basic L2-norm matching of SIFT features between two images in the same computer takes 0.18 seconds on average. Therefore, a query against the 1,329 training images takes 4 minutes on average.

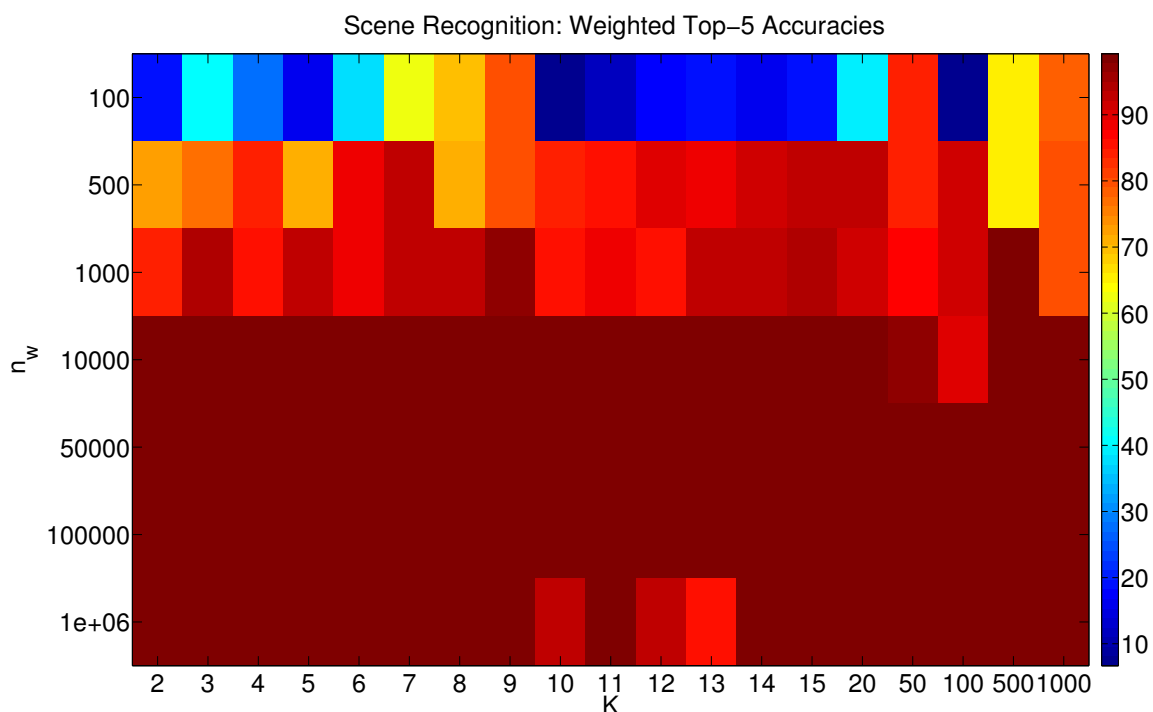


Fig. 5.9.: Color Map of Weighted Top-5 Accuracies of Scene Recognition Using Different Values of  $k$  and  $n_w$ .

Table 5.3: Weighted Top-5 Accuracies of Scene Recognition for Different Values of  $k$  and  $n_w$  (percentage).

$n_w \backslash k$	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>100</b>	19.62	40.64	28.21	16.15	38.33	62.82	68.97	79.23	7.56	11.54
<b>500</b>	72.31	76.41	84.74	70.77	88.72	92.05	71.15	79.62	83.85	85.51
<b>1000</b>	84.74	94.23	84.87	93.21	88.59	92.18	92.95	96.54	85.00	87.69
<b>10000</b>	99.10	99.10	98.21	98.21	98.46	98.21	98.72	98.97	98.21	97.95
<b>50000</b>	98.85	99.10	98.85	98.85	99.23	99.10	99.10	98.85	99.10	99.10
<b>100000</b>	98.85	98.85	98.97	99.10	99.23	98.97	99.23	99.23	99.10	99.10
<b>1000000</b>	99.10	98.85	98.85	98.33	99.23	98.97	97.95	99.23	92.18	99.23

$n_w \backslash k$	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>20</b>	<b>50</b>	<b>100</b>	<b>500</b>	<b>1000</b>
<b>100</b>	17.69	19.10	15.64	18.46	39.23	83.33	6.67	65.64	78.46
<b>500</b>	89.74	87.69	90.64	92.69	92.95	83.59	91.80	65.64	80.26
<b>1000</b>	85.51	92.31	92.31	94.49	91.28	87.31	91.28	98.46	80.00
<b>10000</b>	98.33	98.72	98.46	98.59	98.72	96.54	90.26	98.33	98.72
<b>50000</b>	99.23	99.10	99.23	98.97	98.97	98.21	98.72	98.72	98.33
<b>100000</b>	99.10	99.10	99.10	99.23	98.46	98.46	98.33	97.95	99.10
<b>1000000</b>	92.31	85.00	99.10	98.97	98.72	99.23	98.21	99.10	99.10

Table 5.4: Top-1 Accuracies of Scene Recognition for Different Values of  $k$  and  $n_w$  (percentage).

$n_w \backslash k$	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>100</b>	10.90	17.31	10.90	6.41	14.74	37.82	49.36	58.97	3.21	4.49
<b>500</b>	52.56	55.13	69.23	41.03	76.28	82.05	51.28	58.97	71.80	75.64
<b>1000</b>	73.08	83.33	72.44	83.97	73.72	82.69	83.33	89.10	77.56	73.08
<b>10000</b>	73.08	96.15	93.59	94.23	94.23	93.59	94.23	95.51	91.67	93.59
<b>50000</b>	96.15	96.15	95.51	95.51	96.80	96.15	96.15	94.87	96.15	96.15
<b>100000</b>	93.59	93.59	95.51	96.15	96.80	95.51	96.80	96.80	96.15	96.15
<b>1000000</b>	96.15	96.15	95.51	95.51	94.23	96.80	95.51	95.51	96.80	96.80

$n_w \backslash k$	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>20</b>	<b>50</b>	<b>100</b>	<b>500</b>	<b>1000</b>
<b>100</b>	9.62	10.26	8.33	9.62	19.23	63.46	1.28	42.31	62.18
<b>500</b>	80.13	76.92	82.05	82.69	80.77	66.67	76.92	39.74	62.18
<b>1000</b>	71.80	81.41	80.77	83.97	71.80	69.23	77.56	95.51	61.54
<b>10000</b>	92.31	94.23	95.51	94.23	94.23	92.31	78.21	95.51	96.15
<b>50000</b>	96.80	96.15	96.80	95.51	95.51	94.23	95.51	96.15	96.15
<b>100000</b>	96.15	96.15	96.15	96.80	94.87	94.87	95.51	95.51	96.15
<b>1000000</b>	96.80	94.23	94.23	92.31	94.87	96.80	96.15	96.15	96.15

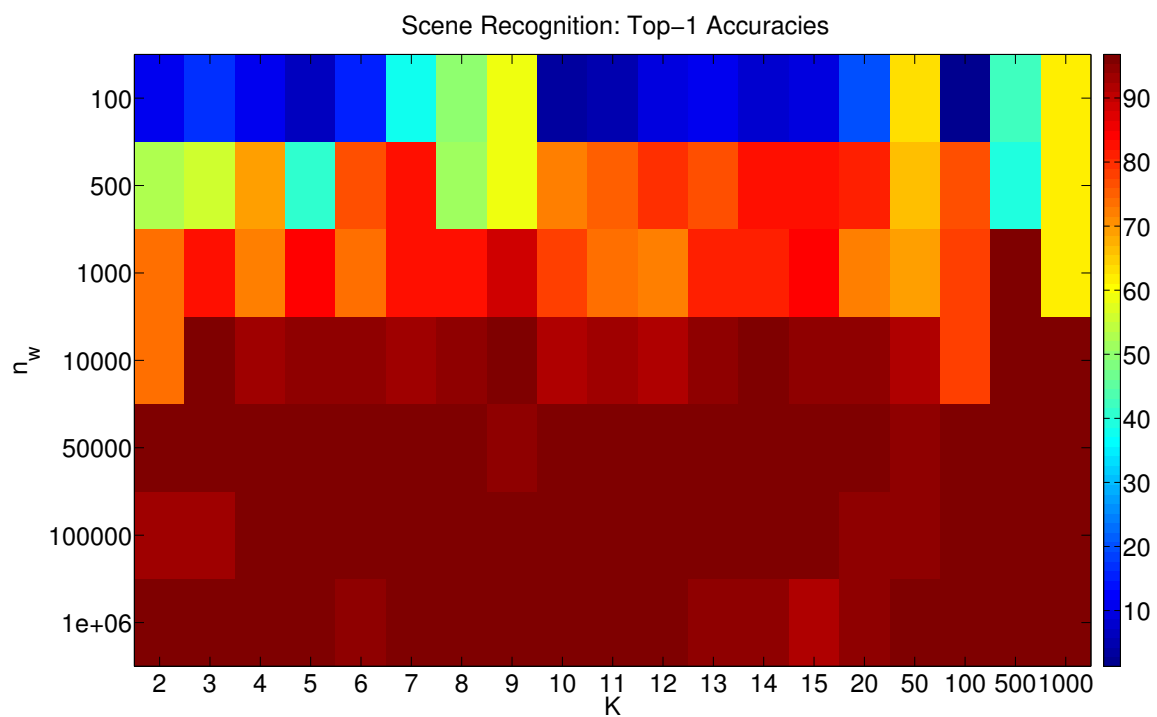


Fig. 5.10.: Color Map of Top-1 Accuracies of Scene Recognition Using Different Values of  $k$  and  $n_w$ .



Table 5.5: Training Times of Scene Recognition for Different Values of  $k$  and  $n_w$  (minutes).

$n_w \backslash k$	2	3	4	5	6	7	8	9	10	11
<b>100</b>	69	94	110	116	107	142	219	399	169	241
<b>500</b>	92	106	112	141	173	189	288	220	243	272
<b>1000</b>	87	121	122	179	186	169	234	256	250	343
<b>10000</b>	146	133	152	179	221	220	220	257	261	416
<b>50000</b>	152	134	152	170	202	208	218	256	302	293
<b>100000</b>	175	154	143	189	205	219	287	270	338	391
<b>1000000</b>	723	429	229	292	328	367	350	323	325	396

$n_w \backslash k$	12	13	14	15	20	50	100	500	1000
<b>100</b>	216	365	278	353	309	327	340	293	386
<b>500</b>	359	337	478	519	307	321	342	292	385
<b>1000</b>	448	373	367	452	308	328	346	300	385
<b>10000</b>	380	369	460	497	309	323	346	299	390
<b>50000</b>	327	371	401	374	309	328	346	300	392
<b>100000</b>	452	370	412	415	457	308	379	320	234
<b>1000000</b>	424	492	530	552	785	311	327	306	205

Table 5.6: Query Times of Scene Recognition for Different Values of  $k$  and  $n_w$  (seconds).

$n_w \backslash k$	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
<b>100</b>	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
<b>500</b>	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
<b>1000</b>	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
<b>10000</b>	3.45	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
<b>50000</b>	52.27	0.07	0.08	0.08	0.12	0.09	0.11	0.07	0.08	0.09
<b>100000</b>	199.09	54.46	35.68	0.15	0.12	0.09	0.11	0.14	0.08	0.09
<b>1000000</b>	6381.98	3444.07	3408.00	3378.00	3349.00	3325.00	3291.00	3278.00	3215.00	3211.00

$n_w \backslash k$	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>20</b>	<b>50</b>	<b>100</b>	<b>500</b>	<b>1000</b>
<b>100</b>	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.09	0.12
<b>500</b>	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.09	0.12
<b>1000</b>	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.14	0.12
<b>10000</b>	0.07	0.07	0.07	0.07	0.08	0.08	0.07	0.14	0.21
<b>50000</b>	0.10	0.11	0.15	0.07	0.08	0.08	0.18	0.14	0.20
<b>100000</b>	0.10	0.11	0.12	0.14	0.08	0.08	0.13	0.14	0.20
<b>1000000</b>	3207.00	3185.00	3182.00	3171.00	3122.00	3081.00	3051.00	3036.00	2997.00

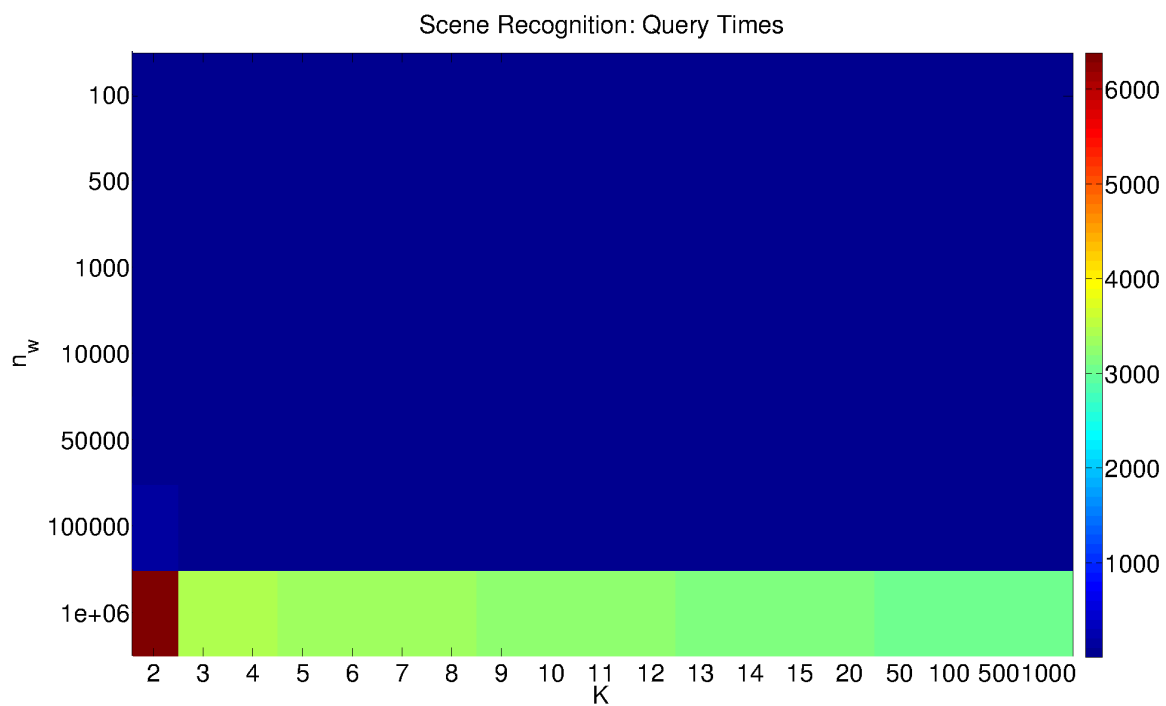


Fig. 5.11.: Color Map of Query Times of Scene Recognition Using Different Values of  $k$  and  $n_w$ .

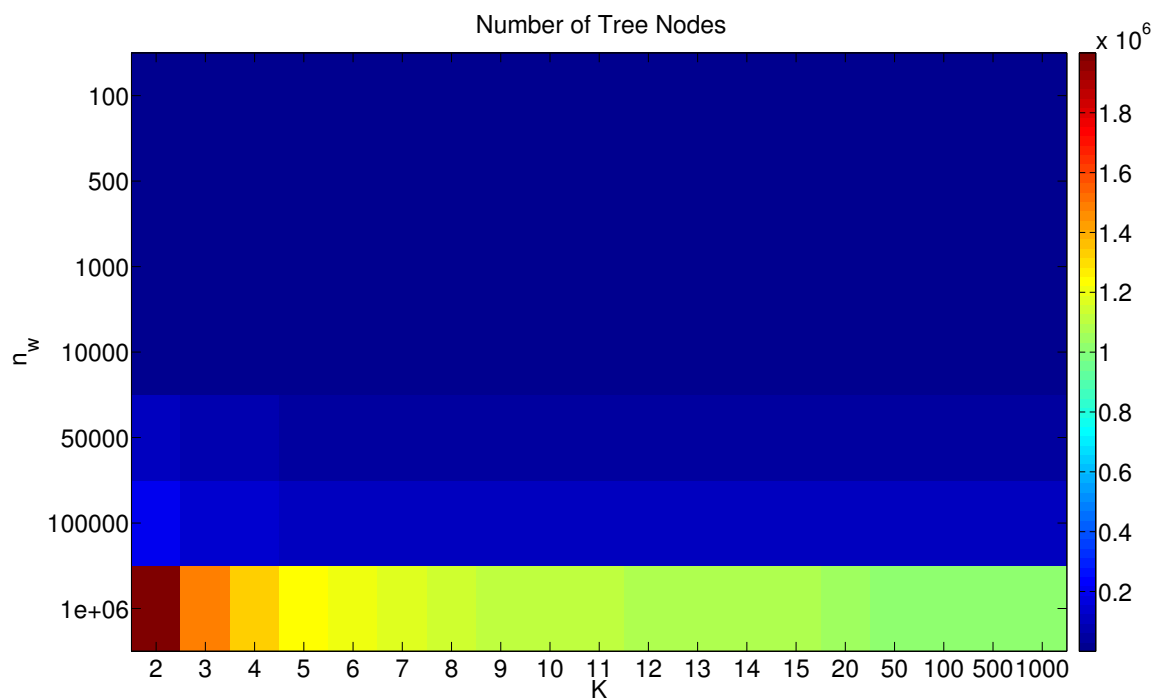


Fig. 5.12.: Number of Vocabulary Tree Nodes As a Function of  $k$  and  $n_w$ .

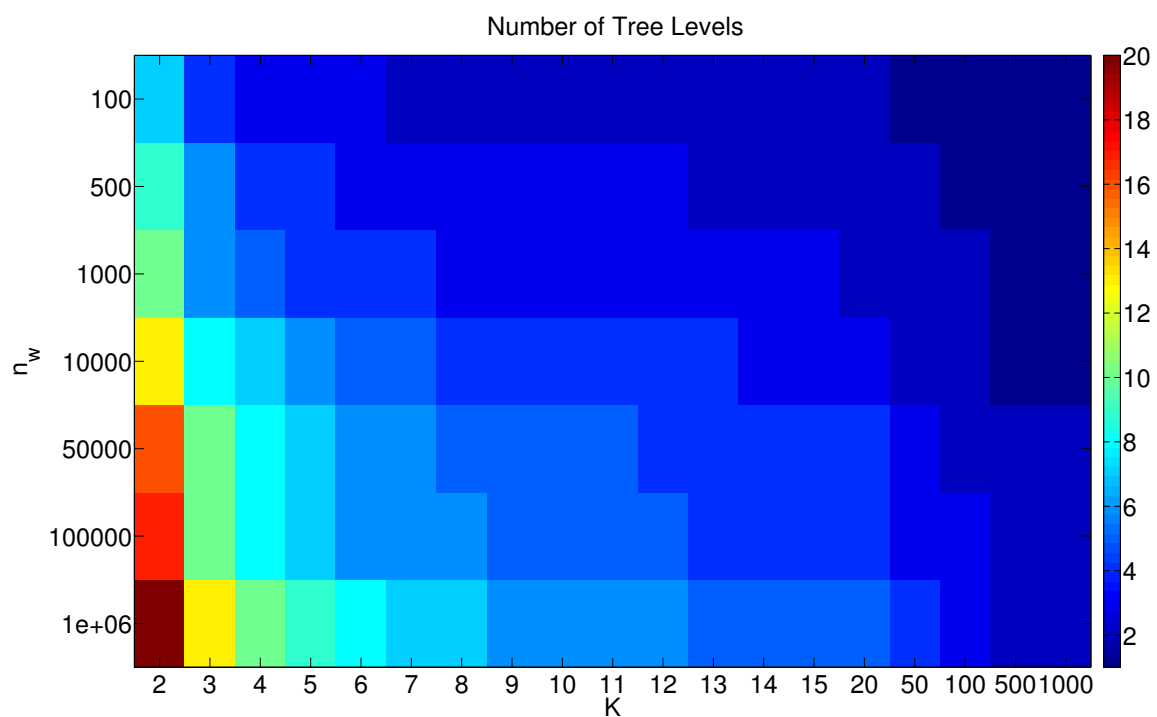


Fig. 5.13.: Number of Vocabulary Tree Levels As a Function of  $k$  and  $n_w$ .

It is worth noting that although this experiment only accounted for scene recognition, we found that sometimes the results returned included scenes of nearby graffiti or even graffiti that have been removed. Figure 5.14 illustrates an example.



Fig. 5.14.: Query Images (Left) And Similar Retrieved Scenes (Right).

The goal of the second experiment was to classify query images into categories based on a set of gang graffiti symbols. We call this process “Gang Graffiti Component Classification”. We created 14 classes for training, where each class corresponds to a distinct graffiti component, including: *0*, *1*, *2*, *3*, *4*, *5-point star*, *G*, *6-point star*, *8*, *arrow*, *E*, *pitchfork*, *S*, and *X*. For each class we trained 17 images, making a total of 238 images for training. Each training image consists of one graffiti component in black with white background. A separate set of 56 images, 4 images per class, was used for testing. Each of the test images also consisted of one graffiti component in black with white background. Figure 5.15 shows some sample images. Note the inter-class variance as well as the intra-class similarity.

Since in this experiment we used our proposed SIFT-based Local Shape Context (LSC) descriptors to generate the vocabulary tree we need to set two additional parameters:  $n_r$  for the number of concentric circumferences representing log-radial distance bins and  $n_\theta$  for the number of angular bins. Given the results of the first experiment we chose  $k = 3$  and  $n_w = 10,000$  to create the vocabulary tree.

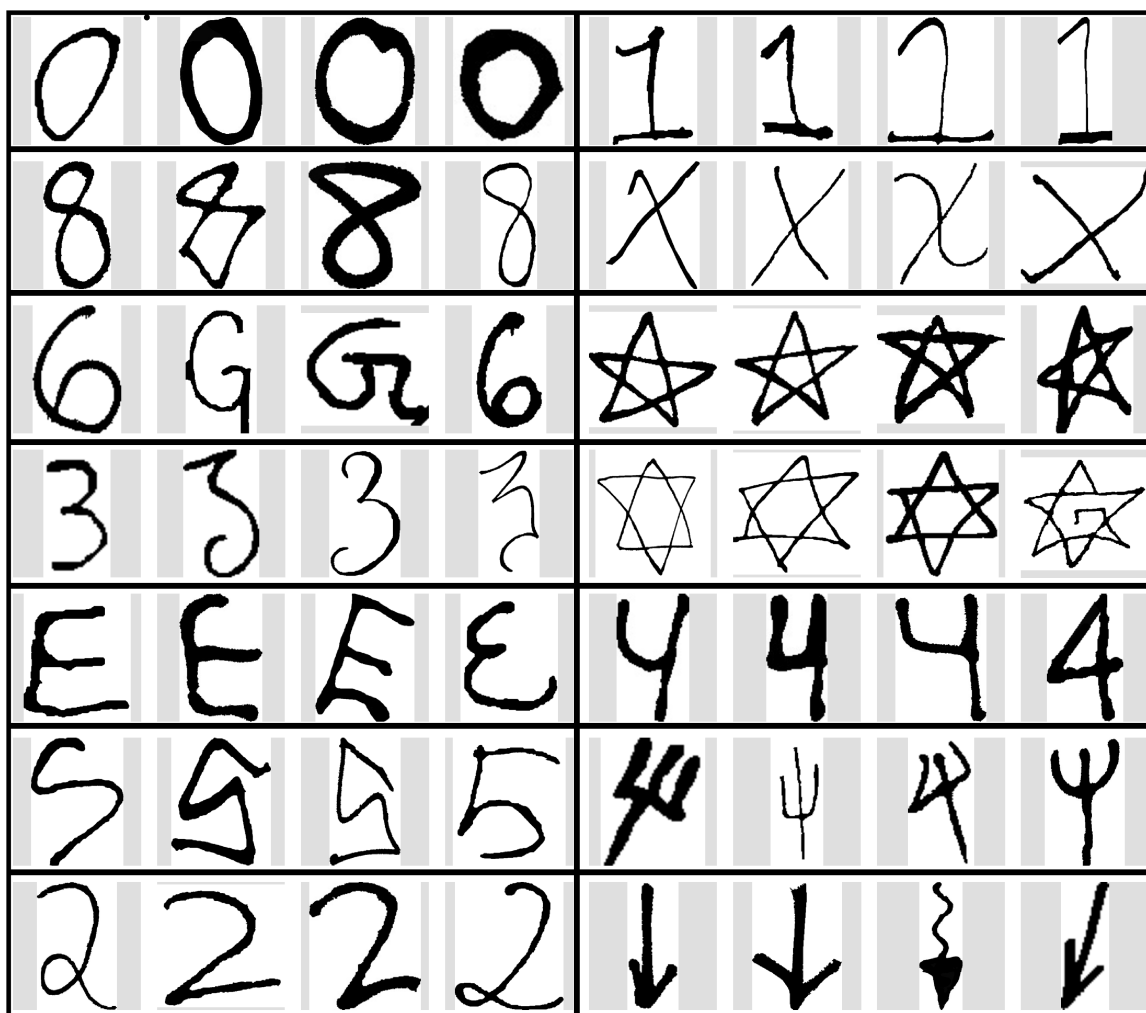


Fig. 5.15.: Sample Images for Each Class. From left to right, top to bottom, in groups of 4 images: 0, 1, 8, X, G, 5-point star, 3, 6-point star, E, 4, S, pitchfork, 2, and arrow. Note the inter-class variance as well as the intra-class similarity.

For each query image we retrieved its 10 closest matches from the training set and we assigned a class based on the following scoring method. Given the scores (votes) of the 10 closest matches  $p = \{p_1, \dots, p_{10}\}$  in ascending order, we manually group them into  $N$  classes,  $N \in \{1, \dots, 14\}$ . We add up the new scores associated to each class, and we assign the class  $C$  with the highest score to the query image, such that  $C = \operatorname{argmax}_n \{\sum_k p_k^{(n)}\}$ , where  $k$  is the set of indices of  $p$  belonging to the  $n$ -th class,  $n \in \{1, \dots, N\}$ .

Tables 5.7 and 5.8 summarize the results of the second experiment using different combinations of  $n_r$  and  $n_\theta$  in the range  $n_r \in [1 \dots 20]$  and  $n_\theta \in [4 \dots 30]$ . Tables 5.9 and 5.10 show the Top-10 accuracies, and Tables 5.11 and 5.12 show the Top-5 accuracies for the same ranges of  $n_r$  and  $n_\theta$ .

Figures 5.16 to 5.18 illustrate the same information using color maps. Low values of  $n_\theta$  cause low classification accuracy, because we do not have enough discrimination between feature locations. High values of both  $n_r$  and  $n_\theta$  also cause low classification accuracy, because we do not account for the elasticity of the graffiti components.

Since we use fixed values of  $k$  and  $n_w$  on this experiment,  $n_r$  and  $n_\theta$  do not have a strong impact in the query time. Therefore we can choose our values from the results of Tables 5.7 and 5.8. For  $n_r = 3$  and  $n_\theta = 16$  we achieve a classification accuracy of 89.29% with a Top-10 accuracy of 94.64% and a Top-5 accuracy of 92.86%. The average query time is 71 ms, from which 6 ms are spent on average to compute the LSC descriptor.

Figure 5.19 illustrates the confusion matrix [306] for each of the 14 classes when  $n_r = 3$  and  $n_\theta = 16$ . Each column of the matrix represents the instances in a predicted class, and each row represents the instances in the ground-truth (i.e. expected) class. High counts on the diagonal indicate high classification accuracy for a specific class. Table 5.13 summarizes the classification results for each class, including precision, recall and  $F_1$  score for each class [307]. Given a confusion matrix  $M$  where the  $x$ -axis

Table 5.7: Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [4 \dots 17]$  (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	55.36	69.64	71.43	75.00	75.00	73.21	76.79	71.43	82.14	78.57	80.36	71.43	89.29	80.36
2	60.71	67.86	76.79	69.64	73.21	82.14	73.21	80.36	78.57	78.57	82.14	85.71	87.50	87.50
3	64.29	69.64	80.36	76.79	75.00	80.36	78.57	82.14	76.79	76.79	85.71	78.57	89.29	82.14
4	71.43	71.43	73.21	76.79	82.14	78.57	80.36	80.36	76.79	78.57	78.57	78.57	78.57	83.93
5	71.43	76.79	76.79	73.21	78.57	83.93	76.79	82.14	80.36	80.36	80.36	80.36	83.93	82.14
6	75.00	75.00	75.00	76.79	76.79	69.64	75.00	75.00	76.79	73.21	76.79	78.57	82.14	85.71
7	67.86	73.21	71.43	73.21	67.86	80.36	78.57	73.21	76.79	73.21	71.43	80.36	83.93	80.36
8	62.50	75.00	71.43	76.79	76.79	80.36	73.21	80.36	82.14	76.79	80.36	80.36	80.36	80.36
9	66.07	73.21	73.21	76.79	73.21	82.14	76.79	80.36	75.00	83.93	75.00	80.36	82.14	78.57
10	69.64	75.00	71.43	75.00	76.79	78.57	76.79	76.79	78.57	82.14	73.21	78.57	78.57	83.93
11	78.57	73.21	71.43	75.00	75.00	76.79	76.79	75.00	83.93	83.93	78.57	80.36	83.93	76.79
12	76.79	75.00	71.43	78.57	76.79	78.57	76.79	80.36	83.93	82.14	75.00	82.14	82.14	80.36
13	69.64	71.43	78.57	78.57	80.36	76.79	69.64	80.36	78.57	82.14	76.79	78.57	76.79	80.36
14	69.64	69.64	76.79	76.79	76.79	80.36	78.57	76.79	80.36	78.57	82.14	78.57	82.14	82.14
15	67.86	71.43	76.79	71.43	80.36	75.00	80.36	76.79	82.14	78.57	78.57	85.71	78.57	76.79
16	71.43	69.64	71.43	75.00	73.21	73.21	75.00	80.36	80.36	82.14	75.00	80.36	75.00	83.93
17	66.07	69.64	75.00	73.21	73.21	75.00	78.57	78.57	80.36	78.57	75.00	80.36	75.00	80.36
18	67.86	75.00	73.21	69.64	78.57	80.36	78.57	78.57	78.57	78.57	82.14	78.57	82.14	78.57
19	67.86	69.64	71.43	78.57	78.57	76.79	75.00	76.79	76.79	80.36	76.79	76.79	78.57	75.00
20	64.29	75.00	73.21	80.36	80.36	78.57	67.86	80.36	73.21	76.79	76.79	78.57	85.71	80.36



Table 5.8: Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [18 \dots 30]$  (percentage).

$n_r \backslash n_\theta$	18	19	20	21	22	23	24	25	26	27	28	29	30
<b>1</b>	82.14	80.36	82.14	85.71	82.14	87.50	80.36	82.14	80.36	78.57	78.57	80.36	78.57
<b>2</b>	83.93	83.93	83.93	83.93	83.93	80.36	83.93	85.71	83.93	80.36	82.14	78.57	80.36
<b>3</b>	82.14	82.14	82.14	87.50	78.57	82.14	82.14	83.93	78.57	76.79	82.14	78.57	78.57
<b>4</b>	87.50	76.79	82.14	82.14	85.71	80.36	85.71	78.57	82.14	75.00	76.79	82.14	78.57
<b>5</b>	85.71	83.93	85.71	82.14	87.50	82.14	80.36	80.36	80.36	80.36	85.71	75.00	78.57
<b>6</b>	85.71	78.57	85.71	82.14	85.71	85.71	82.14	80.36	80.36	78.57	80.36	82.14	76.79
<b>7</b>	82.14	78.57	78.57	82.14	85.71	87.50	76.79	78.57	80.36	85.71	76.79	80.36	76.79
<b>8</b>	80.36	76.79	85.71	78.57	76.79	80.36	78.57	80.36	85.71	83.93	78.57	82.14	82.14
<b>9</b>	76.79	76.79	82.14	78.57	76.79	78.57	82.14	75.00	78.57	71.43	80.36	82.14	82.14
<b>10</b>	82.14	80.36	83.93	80.36	78.57	76.79	78.57	78.57	75.00	80.36	75.00	75.00	78.57
<b>11</b>	80.36	78.57	83.93	82.14	78.57	75.00	76.79	78.57	78.57	75.00	76.79	78.57	80.36
<b>12</b>	80.36	78.57	83.93	82.14	80.36	78.57	80.36	82.14	73.21	78.57	80.36	78.57	80.36
<b>13</b>	78.57	78.57	80.36	76.79	82.14	78.57	78.57	82.14	75.00	80.36	76.79	78.57	82.14
<b>14</b>	75.00	80.36	82.14	80.36	73.21	75.00	82.14	80.36	73.21	82.14	76.79	69.64	82.14
<b>15</b>	80.36	75.00	82.14	80.36	82.14	78.57	80.36	83.93	80.36	78.57	76.79	80.36	69.64
<b>16</b>	80.36	80.36	78.57	76.79	75.00	80.36	76.79	78.57	76.79	80.36	80.36	76.79	78.57
<b>17</b>	76.79	78.57	80.36	76.79	80.36	76.79	75.00	83.93	82.14	73.21	71.43	76.79	78.57
<b>18</b>	73.21	82.14	80.36	67.86	78.57	76.79	78.57	75.00	73.21	78.57	82.14	76.79	71.43
<b>19</b>	80.36	80.36	82.14	82.14	78.57	83.93	73.21	78.57	80.36	76.79	75.00	78.57	76.79
<b>20</b>	78.57	75.00	78.57	76.79	69.64	78.57	76.79	76.79	67.86	82.14	78.57	82.14	76.79

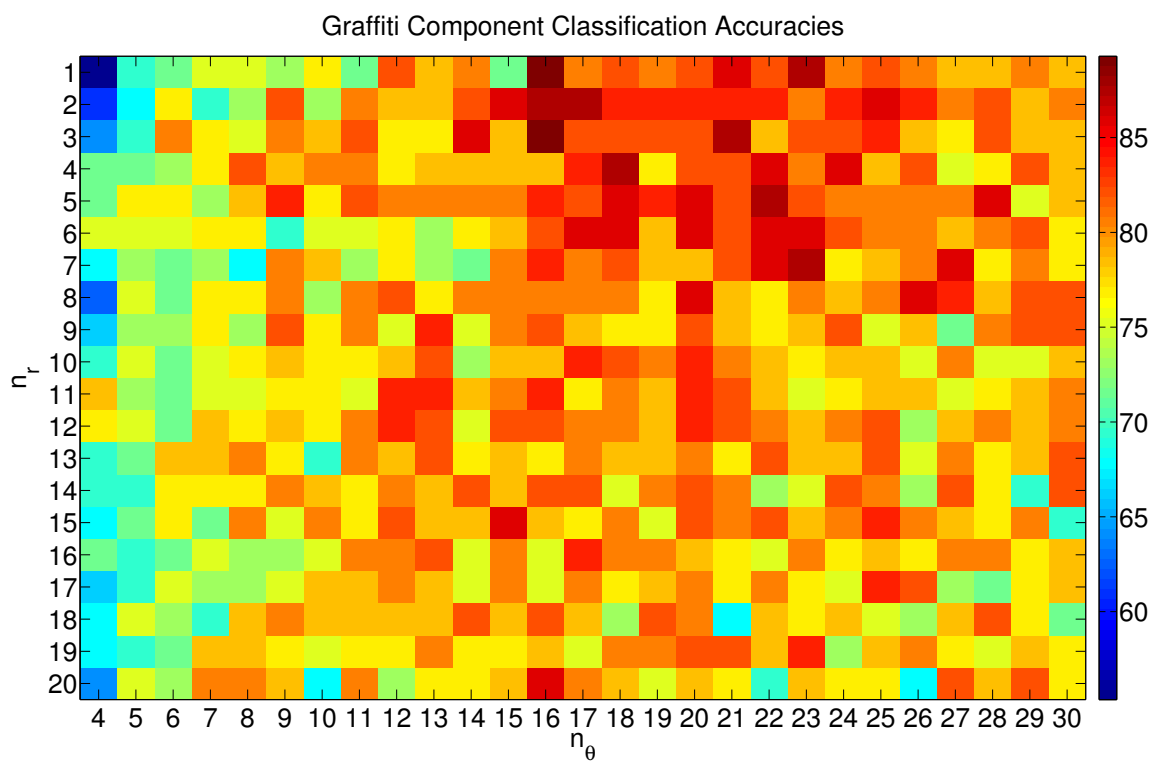


Fig. 5.16.: Color Map of Classification Accuracies of Gang Graffiti Component Classification Using Different Values of  $n_r$  and  $n_\theta$ .

Table 5.9: Top-10 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [4 \dots 17]$  (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	91.07	92.86	96.43	98.21	96.43	98.21	100.00	100.00	96.43	98.21	96.43	98.21	98.21	96.43
2	87.50	92.86	100.00	98.21	98.21	94.64	98.21	94.64	94.64	94.64	94.64	96.43	96.43	98.21
3	96.43	98.21	96.43	98.21	94.64	96.43	96.43	96.43	94.64	96.43	92.86	98.21	94.64	98.21
4	92.86	96.43	98.21	94.64	94.64	98.21	94.64	94.64	98.21	98.21	96.43	94.64	94.64	100.00
5	92.86	96.43	96.43	96.43	94.64	94.64	92.86	94.64	94.64	92.86	92.86	96.43	96.43	96.43
6	91.07	91.07	96.43	98.21	98.21	96.43	96.43	96.43	98.21	94.64	94.64	100.00	100.00	96.43
7	92.86	92.86	96.43	92.86	96.43	94.64	89.29	91.07	96.43	96.43	98.21	94.64	94.64	96.43
8	87.50	92.86	94.64	92.86	94.64	96.43	94.64	94.64	96.43	96.43	94.64	92.86	96.43	92.86
9	92.86	92.86	98.21	92.86	94.64	94.64	94.64	92.86	92.86	96.43	96.43	98.21	94.64	96.43
10	96.43	92.86	91.07	91.07	96.43	98.21	92.86	92.86	94.64	92.86	96.43	91.07	98.21	94.64
11	96.43	89.29	92.86	92.86	94.64	96.43	96.43	94.64	96.43	96.43	94.64	96.43	96.43	92.86
12	92.86	91.07	94.64	92.86	92.86	94.64	94.64	98.21	96.43	96.43	96.43	94.64	96.43	98.21
13	91.07	92.86	94.64	96.43	96.43	92.86	91.07	96.43	96.43	94.64	96.43	96.43	94.64	94.64
14	89.29	91.07	94.64	98.21	96.43	92.86	92.86	92.86	91.07	96.43	94.64	94.64	94.64	94.64
15	92.86	87.50	96.43	94.64	92.86	94.64	91.07	92.86	92.86	96.43	96.43	96.43	92.86	92.86
16	92.86	91.07	96.43	94.64	94.64	91.07	91.07	96.43	94.64	96.43	94.64	94.64	96.43	94.64
17	91.07	87.50	92.86	96.43	94.64	96.43	89.29	96.43	91.07	98.21	92.86	98.21	98.21	94.64
18	96.43	87.50	96.43	91.07	94.64	91.07	92.86	94.64	96.43	92.86	96.43	96.43	94.64	96.43
19	91.07	91.07	92.86	98.21	98.21	91.07	92.86	92.86	94.64	91.07	96.43	94.64	94.64	96.43
20	92.86	91.07	91.07	94.64	92.86	94.64	94.64	96.43	96.43	94.64	96.43	92.86	96.43	100.00

Table 5.10: Top-10 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [18 \dots 30]$  (percentage).

$n_r \backslash n_\theta$	18	19	20	21	22	23	24	25	26	27	28	29	30
1	96.43	98.21	98.21	100.00	98.21	98.21	96.43	98.21	100.00	96.43	100.00	98.21	100.00
2	98.21	96.43	98.21	100.00	100.00	98.21	98.21	98.21	98.21	96.43	98.21	100.00	96.43
3	96.43	94.64	96.43	98.21	100.00	100.00	98.21	100.00	98.21	98.21	100.00	100.00	98.21
4	100.00	98.21	96.43	98.21	96.43	94.64	96.43	98.21	96.43	96.43	94.64	96.43	94.64
5	100.00	96.43	94.64	94.64	96.43	98.21	98.21	96.43	94.64	96.43	98.21	94.64	96.43
6	98.21	91.07	94.64	96.43	100.00	96.43	94.64	96.43	94.64	96.43	96.43	98.21	92.86
7	98.21	96.43	96.43	96.43	96.43	94.64	94.64	96.43	98.21	96.43	92.86	98.21	96.43
8	98.21	96.43	96.43	96.43	96.43	96.43	96.43	98.21	94.64	96.43	96.43	98.21	98.21
9	98.21	96.43	94.64	94.64	96.43	98.21	94.64	98.21	98.21	96.43	98.21	92.86	94.64
10	96.43	96.43	98.21	96.43	98.21	96.43	96.43	92.86	96.43	98.21	94.64	94.64	94.64
11	98.21	98.21	94.64	96.43	98.21	98.21	96.43	94.64	92.86	96.43	96.43	94.64	96.43
12	100.00	98.21	94.64	96.43	91.07	96.43	96.43	96.43	96.43	96.43	98.21	98.21	96.43
13	94.64	96.43	92.86	96.43	98.21	96.43	96.43	96.43	96.43	94.64	91.07	96.43	98.21
14	94.64	100.00	94.64	100.00	96.43	98.21	96.43	94.64	96.43	94.64	92.86	92.86	98.21
15	96.43	96.43	98.21	100.00	98.21	94.64	96.43	96.43	96.43	98.21	94.64	96.43	96.43
16	96.43	96.43	100.00	92.86	96.43	96.43	98.21	94.64	98.21	96.43	94.64	96.43	98.21
17	96.43	92.86	94.64	92.86	98.21	94.64	94.64	96.43	98.21	96.43	96.43	96.43	96.43
18	94.64	98.21	100.00	94.64	100.00	96.43	96.43	96.43	96.43	94.64	94.64	94.64	92.86
19	100.00	94.64	96.43	96.43	96.43	91.07	96.43	96.43	98.21	96.43	94.64	96.43	94.64
20	96.43	98.21	98.21	98.21	98.21	96.43	96.43	92.86	96.43	94.64	98.21	92.86	94.64

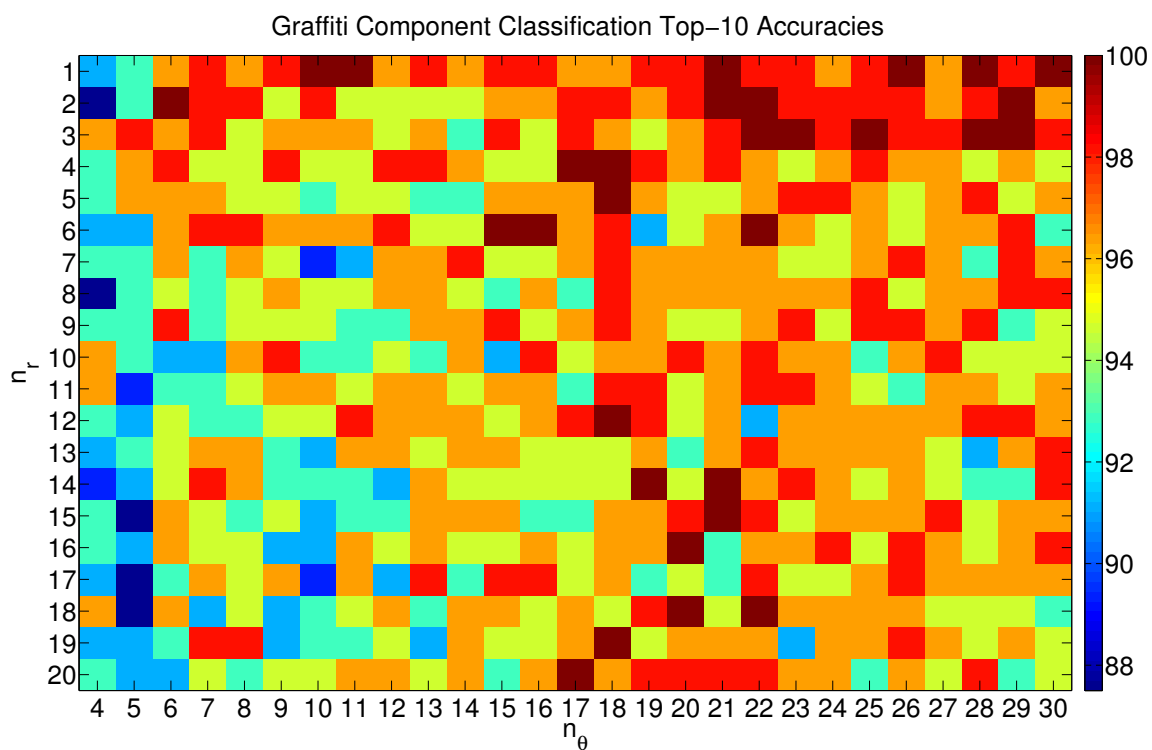


Fig. 5.17.: Color Map of Top-10 Classification Accuracies of Gang Graffiti Component Classification Using Different Values of  $n_r$  and  $n_\theta$ .

Table 5.11: Top-5 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [4 \dots 17]$  (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	76.79	87.50	89.29	92.86	96.43	92.86	96.43	92.86	92.86	91.07	89.29	92.86	96.43	92.86
2	78.57	91.07	94.64	91.07	94.64	92.86	94.64	89.29	89.29	92.86	92.86	94.64	96.43	96.43
3	83.93	89.29	94.64	89.29	87.50	91.07	94.64	92.86	94.64	91.07	91.07	96.43	92.86	94.64
4	91.07	87.50	96.43	92.86	91.07	89.29	92.86	92.86	92.86	92.86	92.86	92.86	94.64	96.43
5	89.29	89.29	92.86	91.07	91.07	91.07	89.29	94.64	87.50	91.07	92.86	92.86	91.07	92.86
6	85.71	89.29	92.86	98.21	92.86	92.86	89.29	89.29	91.07	92.86	92.86	96.43	96.43	91.07
7	83.93	85.71	89.29	89.29	91.07	92.86	89.29	89.29	91.07	92.86	91.07	92.86	91.07	89.29
8	82.14	87.50	92.86	89.29	89.29	92.86	91.07	94.64	89.29	91.07	87.50	92.86	92.86	91.07
9	87.50	89.29	94.64	87.50	89.29	89.29	85.71	92.86	91.07	92.86	91.07	91.07	91.07	92.86
10	92.86	89.29	91.07	85.71	85.71	94.64	89.29	92.86	89.29	89.29	94.64	87.50	91.07	91.07
11	89.29	85.71	87.50	91.07	91.07	92.86	89.29	87.50	94.64	92.86	92.86	92.86	96.43	87.50
12	87.50	83.93	92.86	87.50	92.86	89.29	91.07	89.29	92.86	94.64	89.29	92.86	87.50	94.64
13	83.93	85.71	91.07	89.29	92.86	85.71	89.29	92.86	91.07	89.29	91.07	89.29	94.64	89.29
14	85.71	83.93	87.50	92.86	92.86	89.29	92.86	91.07	91.07	91.07	92.86	91.07	94.64	92.86
15	89.29	82.14	87.50	91.07	89.29	91.07	87.50	91.07	89.29	89.29	91.07	94.64	87.50	87.50
16	89.29	83.93	91.07	94.64	91.07	89.29	87.50	92.86	89.29	94.64	92.86	89.29	94.64	91.07
17	85.71	83.93	87.50	91.07	92.86	87.50	85.71	87.50	89.29	91.07	92.86	94.64	96.43	91.07
18	89.29	80.36	94.64	91.07	92.86	89.29	89.29	91.07	89.29	92.86	91.07	91.07	92.86	92.86
19	85.71	89.29	89.29	89.29	92.86	89.29	85.71	89.29	85.71	89.29	94.64	91.07	89.29	91.07
20	83.93	85.71	83.93	91.07	91.07	87.50	89.29	89.29	92.86	92.86	92.86	89.29	92.86	91.07

Table 5.12: Top-5 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [18 \dots 30]$  (percentage).

$n_r \backslash n_\theta$	18	19	20	21	22	23	24	25	26	27	28	29	30
1	96.43	98.21	98.21	100.00	98.21	98.21	96.43	98.21	100.00	96.43	100.00	98.21	100.00
2	98.21	96.43	98.21	100.00	100.00	98.21	98.21	98.21	98.21	96.43	98.21	100.00	96.43
3	96.43	94.64	96.43	98.21	100.00	100.00	98.21	100.00	98.21	98.21	100.00	100.00	98.21
4	100.00	98.21	96.43	98.21	96.43	94.64	96.43	98.21	96.43	96.43	94.64	96.43	94.64
5	100.00	96.43	94.64	94.64	96.43	98.21	98.21	96.43	94.64	96.43	98.21	94.64	96.43
6	98.21	91.07	94.64	96.43	100.00	96.43	94.64	96.43	94.64	96.43	96.43	98.21	92.86
7	98.21	96.43	96.43	96.43	96.43	94.64	94.64	96.43	98.21	96.43	92.86	98.21	96.43
8	98.21	96.43	96.43	96.43	96.43	96.43	96.43	98.21	94.64	96.43	96.43	98.21	98.21
9	98.21	96.43	94.64	94.64	96.43	98.21	94.64	98.21	98.21	96.43	98.21	92.86	94.64
10	96.43	96.43	98.21	96.43	98.21	96.43	96.43	92.86	96.43	98.21	94.64	94.64	94.64
11	98.21	98.21	94.64	96.43	98.21	98.21	96.43	94.64	92.86	96.43	96.43	94.64	96.43
12	100.00	98.21	94.64	96.43	91.07	96.43	96.43	96.43	96.43	96.43	98.21	98.21	96.43
13	94.64	96.43	92.86	96.43	98.21	96.43	96.43	96.43	96.43	94.64	91.07	96.43	98.21
14	94.64	100.00	94.64	100.00	96.43	98.21	96.43	94.64	96.43	94.64	92.86	92.86	98.21
15	96.43	96.43	98.21	100.00	98.21	94.64	96.43	96.43	96.43	98.21	94.64	96.43	96.43
16	96.43	96.43	100.00	92.86	96.43	96.43	98.21	94.64	98.21	96.43	94.64	96.43	98.21
17	96.43	92.86	94.64	92.86	98.21	94.64	94.64	96.43	98.21	96.43	96.43	96.43	96.43
18	94.64	98.21	100.00	94.64	100.00	96.43	96.43	96.43	96.43	94.64	94.64	94.64	92.86
19	100.00	94.64	96.43	96.43	96.43	91.07	96.43	96.43	98.21	96.43	94.64	96.43	94.64
20	96.43	98.21	98.21	98.21	98.21	96.43	96.43	92.86	96.43	94.64	98.21	92.86	94.64

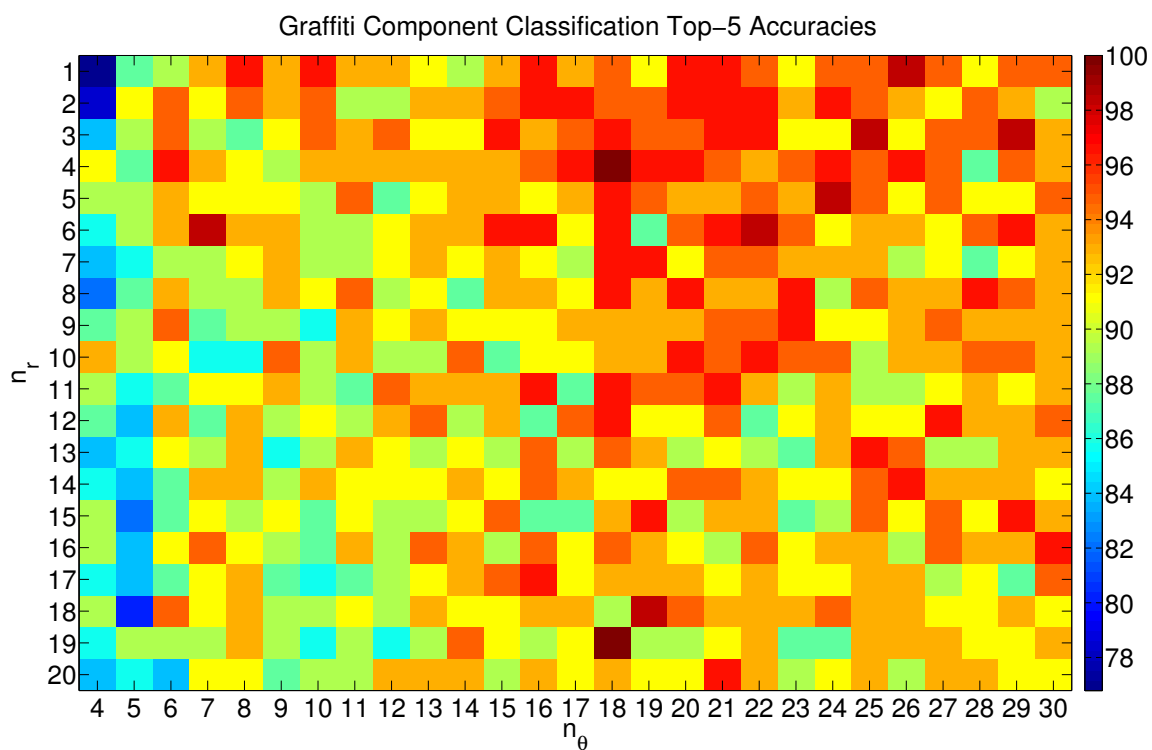


Fig. 5.18.: Color Map of Top-5 Classification Accuracies of Gang Graffiti Component Classification Using Different Values of  $n_r$  and  $n_\theta$ .



Table 5.13: Classification Accuracy, Precision, Recall and  $F_1$  Score for Each Class.

Class	Accuracy	Precision	Recall	$F_1$ Score
0	100%	100%	100%	1
8	100%	66.67%	100%	0.80
G	75%	100%	75%	0.86
3	100%	100%	100%	1
E	75%	100%	75%	0.86
s	50%	100%	50%	0.67
2	100%	80%	100%	0.89
1	100%	80%	100%	0.89
x	100%	100%	100%	1
5-point star	100%	80%	100%	0.89
6-point star	100%	100%	100%	1
4	75%	100%	75%	0.86
pitchfork	100%	80%	100%	0.89
arrow	75%	100%	75%	0.86

corresponds to predicted outputs and the  $y$ -axis corresponds to expected outputs, precision  $P_i$  and recall  $R_i$  for class  $i$  are defined as

$$P_i = \frac{M_{ii}}{\sum_j M_{ji}} \quad (5.8)$$

$$R_i = \frac{M_{ii}}{\sum_j M_{ij}}. \quad (5.9)$$

Given precision and recall values, the  $F_{1i}$  score is given by

$$F_{1i} = 2 \frac{P_i R_i}{P_i + R_i} \quad (5.10)$$

As a comparison, Tables 5.14 to 5.16 show the classification accuracies, Top-10 accuracies and Top-5 accuracies when using SIFT descriptors instead of LSC descriptors. The maximum classification accuracy achieved is 41.07% with  $n_r = 6$  and  $n_\theta = 13$ , with a Top-10 accuracy of 75.00% and a Top-5 accuracy of 55.36%. The

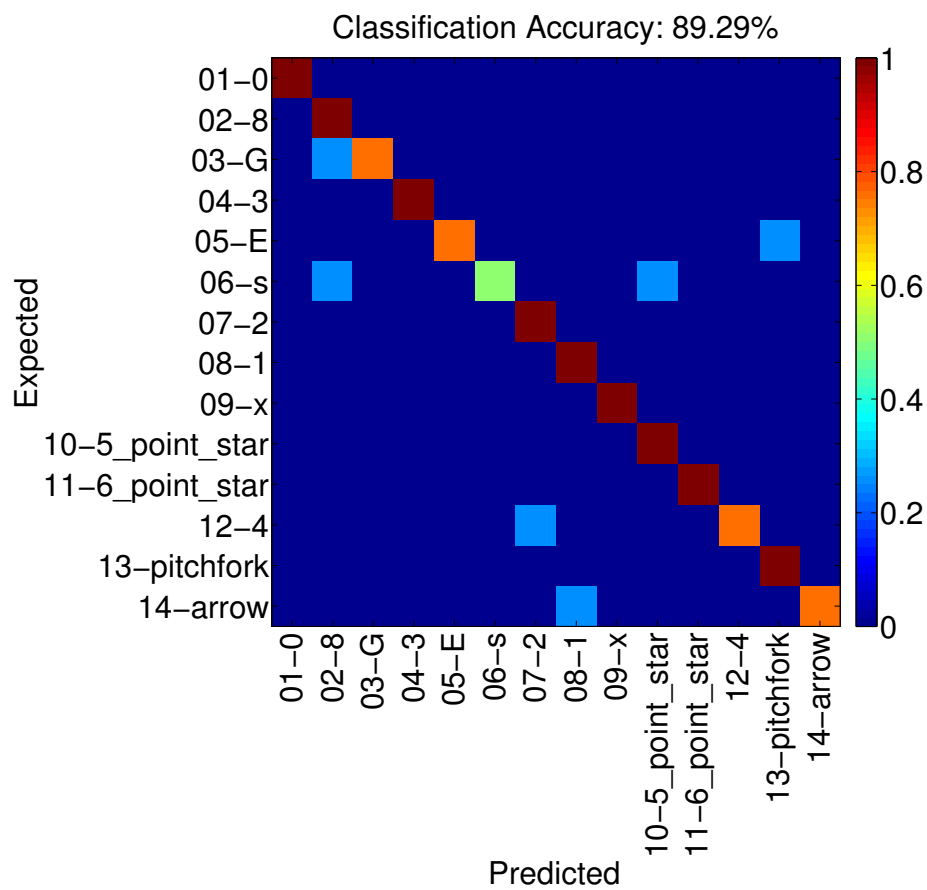


Fig. 5.19.: Confusion Matrix for the 14 Graffiti Component Classes.

average query time is the same as when using LSC descriptors, because most of the time is spent pushing the descriptors down the vocabulary tree.

In order to evaluate the overall performance of our “Gang Graffiti Component Classification” system we also used the Mean Average Precision (*MAP*) measure, which provides a single-figure measure of quality across recall levels and has been shown to have especially good discrimination and stability [308–310].

The *MAP* is defined as

$$MAP = \frac{1}{Q} \sum_{j=1}^Q \frac{1}{N} \sum_{k=1}^N P_{jk}, \quad (5.11)$$

where  $Q$  is the total number of query images and  $N$  is the number of database images retrieved for each query. Equation 5.11 can be redefined as the average precision scores for the set of queries:

$$MAP = \frac{\sum_{j=1}^Q AveP(j)}{Q}, \quad (5.12)$$

where  $AveP(j)$  is average precision of the  $j$ -th query image, defined as

$$AveP(j) = \frac{\sum_{k=1}^N P_{jk}}{N}, \quad (5.13)$$

being  $P_{jk}$  is the precision of the  $j$ -th query image at rank  $k$ :

$$P_{jk} = \frac{\sum_{i=1}^k I_{ji}}{k}. \quad (5.14)$$

Table 5.14: Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 10]$  and  $n_\theta \in [4 \dots 17]$  using SIFT Descriptors (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>1</b>	33.93	35.71	32.14	37.50	33.93	37.50	32.14	30.36	30.36	32.14	39.29	35.71	30.36	33.93
<b>2</b>	33.93	28.57	32.14	32.14	30.36	32.14	33.93	37.50	28.57	30.36	30.36	33.93	30.36	30.36
<b>3</b>	32.14	30.36	30.36	28.57	33.93	35.71	28.57	35.71	33.93	32.14	33.93	30.36	33.93	32.14
<b>4</b>	26.79	33.93	32.14	37.50	41.07	35.71	30.36	33.93	33.93	33.93	33.93	30.36	35.71	30.36
<b>5</b>	30.36	32.14	33.93	33.93	35.71	30.36	28.57	33.93	30.36	30.36	35.71	32.14	32.14	28.57
<b>6</b>	28.57	30.36	30.36	33.93	33.93	33.93	30.36	32.14	37.50	41.07	35.71	28.57	33.93	33.93
<b>7</b>	32.14	32.14	32.14	35.71	35.71	32.14	30.36	39.29	33.93	32.14	35.71	30.36	25.00	32.14
<b>8</b>	32.14	28.57	33.93	30.36	32.14	33.93	26.79	32.14	33.93	30.36	35.71	35.71	30.36	33.93
<b>9</b>	32.14	35.71	32.14	32.14	28.57	30.36	35.71	30.36	33.93	30.36	35.71	35.71	33.93	33.93
<b>10</b>	30.36	32.14	35.71	33.93	32.14	33.93	33.93	33.93	30.36	33.93	35.71	30.36	32.14	30.36

Table 5.15: Top-10 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 10]$  and  $n_\theta \in [4 \dots 17]$  using SIFT Descriptors (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>1</b>	73.21	76.79	75.00	69.64	64.29	67.86	67.86	69.64	71.43	67.86	73.21	67.86	66.07	69.64
<b>2</b>	73.21	69.64	75.00	71.43	67.86	73.21	71.43	71.43	67.86	67.86	66.07	64.29	73.21	73.21
<b>3</b>	69.64	75.00	78.57	71.43	67.86	69.64	69.64	76.79	64.29	67.86	67.86	76.79	66.07	71.43
<b>4</b>	69.64	71.43	66.07	69.64	66.07	75.00	66.07	67.86	64.29	67.86	76.79	69.64	78.57	73.21
<b>5</b>	73.21	71.43	73.21	71.43	67.86	67.86	75.00	69.64	78.57	71.43	73.21	69.64	69.64	66.07
<b>6</b>	73.21	75.00	69.64	69.64	73.21	62.50	73.21	75.00	66.07	75.00	69.64	69.64	69.64	67.86
<b>7</b>	73.21	75.00	71.43	67.86	75.00	75.00	62.50	75.00	67.86	69.64	69.64	69.64	71.43	69.64
<b>8</b>	71.43	78.57	69.64	66.07	75.00	64.29	75.00	71.43	71.43	69.64	71.43	69.64	76.79	69.64
<b>9</b>	62.50	66.07	67.86	66.07	67.86	80.36	69.64	60.71	78.57	69.64	71.43	71.43	75.00	67.86
<b>10</b>	71.43	64.29	67.86	73.21	62.50	71.43	71.43	69.64	66.07	76.79	66.07	69.64	67.86	67.86

Table 5.16: Top-5 Classification Accuracies of Gang Graffiti Component Classification for  $n_r \in [1 \dots 10]$  and  $n_\theta \in [4 \dots 17]$  using SIFT Descriptors (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>1</b>	51.79	50.00	57.14	53.57	48.21	53.57	55.36	53.57	53.57	55.36	64.29	53.57	53.57	48.21
<b>2</b>	60.71	50.00	58.93	50.00	53.57	53.57	55.36	55.36	51.79	53.57	51.79	50.00	53.57	50.00
<b>3</b>	55.36	48.21	58.93	55.36	57.14	55.36	51.79	58.93	48.21	53.57	55.36	62.50	53.57	58.93
<b>4</b>	50.00	58.93	48.21	55.36	55.36	58.93	50.00	58.93	48.21	50.00	55.36	55.36	67.86	53.57
<b>5</b>	55.36	57.14	62.50	57.14	53.57	57.14	53.57	53.57	55.36	57.14	57.14	51.79	58.93	53.57
<b>6</b>	58.93	55.36	53.57	55.36	53.57	53.57	57.14	53.57	55.36	55.36	55.36	50.00	62.50	48.21
<b>7</b>	58.93	51.79	53.57	57.14	66.07	57.14	51.79	62.50	57.14	53.57	58.93	58.93	50.00	53.57
<b>8</b>	51.79	55.36	55.36	53.57	51.79	50.00	51.79	55.36	46.43	53.57	55.36	50.00	55.36	55.36
<b>9</b>	51.79	55.36	53.57	57.14	50.00	55.36	58.93	51.79	53.57	55.36	51.79	51.79	60.71	53.57
<b>10</b>	53.57	50.00	58.93	55.36	50.00	57.14	57.14	60.71	51.79	57.14	51.79	57.14	57.14	55.36

Table 5.17: Example of *MAP* score calculation for a set of two queries. The total *MAP* score is  $\frac{0.22+0.41}{2} = 0.31$ .

Prediction	Correctness	Precision
1	wrong	none
2	right	1/2
3	right	2/3
4	wrong	none
5	right	3/5
6	wrong	none
7	wrong	none
8	wrong	none
9	right	4/9
10	wrong	none

$$(a) \text{Ave}P = \frac{1/2+2/3+3/5+4/9}{10} = 0.22$$

Prediction	Correctness	Precision
1	right	1/1
2	right	2/2
3	right	3/3
4	wrong	none
5	wrong	none
6	wrong	none
7	wrong	none
8	right	4/8
9	right	5/9
10	wrong	none

$$(b) \text{Ave}P = \frac{1/1+2/2+3/3+4/8+5/9}{10} = 0.41$$

$I_{ji}$  is an indicator function equaling 1 if the  $j$ -th query image at rank  $k$  is a match, and zero otherwise. Table 5.17 shows an example of how to calculate the *MAP* score with  $Q = 2$  and  $N = 10$ . In our experiments  $Q = 56$  and  $N = 10$ .

Tables 5.18 and 5.19 show the *MAP* scores for a range of  $n_r$  and  $n_\theta$  using LSC descriptors. Figure 5.20 illustrates the same information using a color map. This results confirm that not using enough bins for radius and angles, or using too many, will cause the classification accuracy to drop. Values of  $n_r \in [3 \dots 18]$  and  $n_\theta \in$

[12...28] provide enough discrimination between feature locations and robustness against shape elasticity.

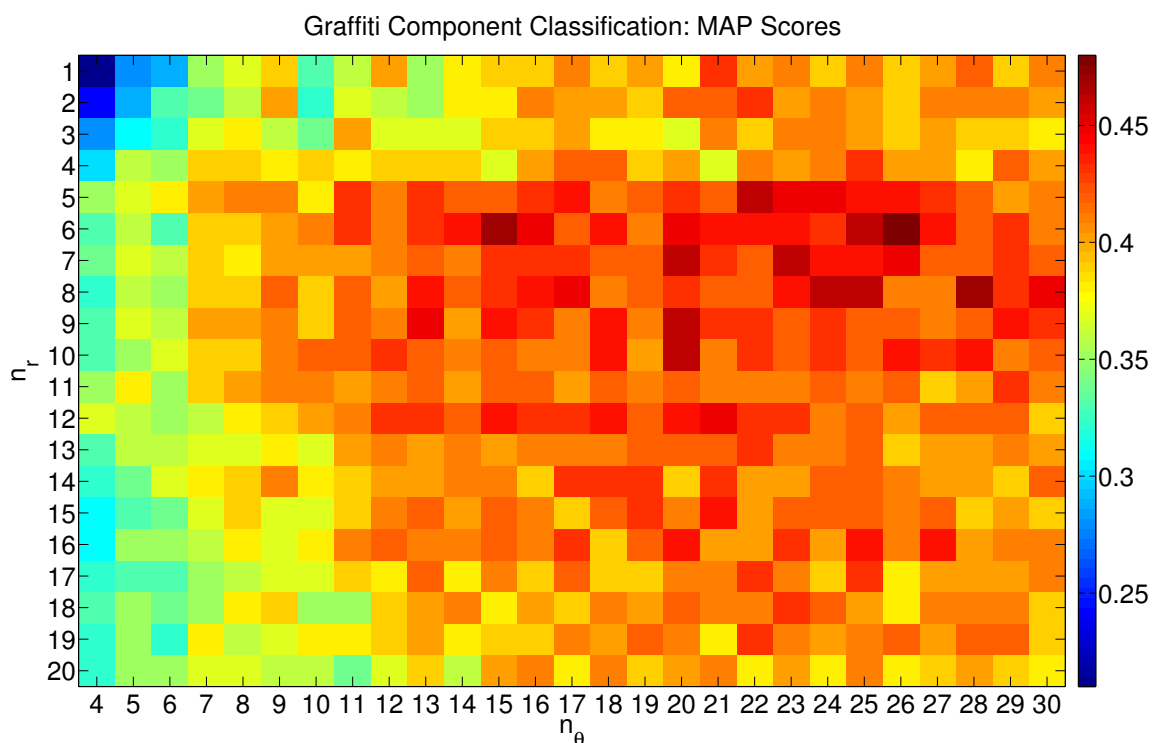


Fig. 5.20.: Color Map of MAP Scores of Gang Graffiti Component Classification Using Different Values of  $n_r$  and  $n_\theta$ .

#### 5.1.4 End-To-End System

In this experiment we tested the entire GARI system, including all the steps shown in Figure 5.21. The system is composed of seven blocks: Color Recognition Based on Touchscreen Tracing, Color Correction Based on Mobile Light Sensor, Color Image Segmentation Based on Gaussian Thresholding, Block-Wise Gaussian Segmentation Enhancement, Background Stripe Removal, Graffiti Component Reconnection, and Graffiti Component Classification. Note that the Color Recognition Based on Touch-



Table 5.18: MAP Scores of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [4 \dots 17]$  (percentage).

$n_r \backslash n_\theta$	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>1</b>	0.23	0.28	0.32	0.37	0.40	0.37	0.35	0.37	0.39	0.39	0.39	0.39	0.41	0.42
<b>2</b>	0.23	0.27	0.33	0.35	0.37	0.39	0.34	0.35	0.38	0.36	0.39	0.42	0.41	0.40
<b>3</b>	0.32	0.34	0.34	0.39	0.38	0.39	0.37	0.37	0.38	0.35	0.37	0.38	0.38	0.39
<b>4</b>	0.34	0.39	0.39	0.41	0.40	0.39	0.41	0.41	0.39	0.40	0.41	0.42	0.40	0.42
<b>5</b>	0.35	0.39	0.40	0.42	0.40	0.41	0.41	0.44	0.43	0.41	0.42	0.45	0.44	0.44
<b>6</b>	0.34	0.39	0.38	0.41	0.41	0.41	0.39	0.41	0.42	0.44	0.43	0.43	0.43	0.46
<b>7</b>	0.36	0.37	0.36	0.40	0.40	0.40	0.39	0.42	0.41	0.41	0.43	0.44	0.44	0.42
<b>8</b>	0.36	0.37	0.37	0.39	0.39	0.42	0.41	0.41	0.41	0.44	0.43	0.42	0.41	0.43
<b>9</b>	0.34	0.39	0.37	0.41	0.41	0.42	0.39	0.42	0.39	0.42	0.43	0.44	0.44	0.43
<b>10</b>	0.34	0.39	0.39	0.41	0.40	0.43	0.40	0.42	0.41	0.43	0.43	0.42	0.43	0.42
<b>11</b>	0.36	0.39	0.39	0.40	0.42	0.42	0.43	0.42	0.42	0.43	0.44	0.46	0.44	0.44
<b>12</b>	0.35	0.36	0.38	0.41	0.43	0.41	0.41	0.39	0.42	0.43	0.42	0.42	0.42	0.42
<b>13</b>	0.34	0.36	0.38	0.43	0.39	0.38	0.38	0.42	0.40	0.42	0.41	0.43	0.41	0.42
<b>14</b>	0.36	0.37	0.36	0.40	0.41	0.40	0.41	0.41	0.40	0.40	0.41	0.39	0.42	0.41
<b>15</b>	0.34	0.36	0.36	0.41	0.42	0.42	0.39	0.41	0.42	0.44	0.43	0.43	0.41	0.41
<b>16</b>	0.34	0.37	0.37	0.40	0.40	0.41	0.42	0.41	0.43	0.41	0.42	0.42	0.42	0.43
<b>17</b>	0.34	0.34	0.37	0.42	0.40	0.38	0.41	0.41	0.40	0.43	0.41	0.41	0.42	0.42
<b>18</b>	0.35	0.36	0.37	0.39	0.39	0.39	0.41	0.42	0.42	0.42	0.39	0.44	0.42	0.43
<b>19</b>	0.34	0.36	0.40	0.40	0.38	0.40	0.38	0.39	0.42	0.41	0.42	0.39	0.42	0.41
<b>20</b>	0.35	0.36	0.37	0.39	0.39	0.41	0.39	0.40	0.41	0.41	0.39	0.40	0.42	0.42

Table 5.19: MAP Scores of Gang Graffiti Component Classification for  $n_r \in [1 \dots 20]$  and  $n_\theta \in [18 \dots 30]$  (percentage).

$n_r \backslash n_\theta$	18	19	20	21	22	23	24	25	26	27	28	29	30
<b>1</b>	0.40	0.42	0.39	0.41	0.43	0.42	0.41	0.41	0.42	0.42	0.39	0.40	0.42
<b>2</b>	0.40	0.42	0.39	0.42	0.41	0.41	0.41	0.40	0.40	0.41	0.43	0.41	0.43
<b>3</b>	0.40	0.42	0.39	0.40	0.41	0.39	0.40	0.40	0.41	0.40	0.40	0.41	0.41
<b>4</b>	0.41	0.38	0.42	0.43	0.41	0.42	0.41	0.41	0.41	0.42	0.40	0.42	0.42
<b>5</b>	0.43	0.44	0.43	0.44	0.44	0.45	0.43	0.45	0.44	0.44	0.44	0.45	0.45
<b>6</b>	0.44	0.44	0.46	0.42	0.46	0.46	0.47	0.45	0.45	0.45	0.46	0.44	0.43
<b>7</b>	0.44	0.44	0.43	0.46	0.44	0.45	0.45	0.45	0.46	0.44	0.44	0.44	0.43
<b>8</b>	0.43	0.43	0.43	0.43	0.43	0.43	0.45	0.43	0.44	0.44	0.43	0.45	0.44
<b>9</b>	0.43	0.44	0.47	0.43	0.44	0.42	0.44	0.44	0.43	0.43	0.44	0.45	0.45
<b>10</b>	0.44	0.43	0.44	0.43	0.43	0.43	0.43	0.42	0.44	0.42	0.42	0.45	0.43
<b>11</b>	0.42	0.43	0.44	0.44	0.45	0.44	0.42	0.43	0.44	0.42	0.41	0.45	0.43
<b>12</b>	0.43	0.44	0.44	0.42	0.44	0.44	0.42	0.43	0.43	0.43	0.42	0.47	0.43
<b>13</b>	0.43	0.41	0.44	0.44	0.43	0.44	0.42	0.43	0.42	0.42	0.41	0.42	0.41
<b>14</b>	0.42	0.42	0.44	0.43	0.43	0.42	0.43	0.43	0.42	0.40	0.40	0.42	0.42
<b>15</b>	0.42	0.42	0.44	0.44	0.42	0.44	0.42	0.44	0.42	0.43	0.43	0.41	0.44
<b>16</b>	0.43	0.43	0.42	0.43	0.40	0.45	0.44	0.43	0.41	0.43	0.43	0.41	0.39
<b>17</b>	0.43	0.43	0.45	0.41	0.43	0.43	0.42	0.44	0.41	0.44	0.41	0.43	0.40
<b>18</b>	0.42	0.41	0.43	0.43	0.45	0.43	0.43	0.42	0.42	0.43	0.42	0.42	0.39
<b>19</b>	0.43	0.42	0.43	0.42	0.43	0.44	0.44	0.42	0.41	0.43	0.42	0.42	0.40
<b>20</b>	0.43	0.40	0.42	0.43	0.43	0.41	0.42	0.41	0.42	0.41	0.41	0.42	0.41

screen Tracing is the only step that is done on the mobile device. The rest of the process is done in the server.

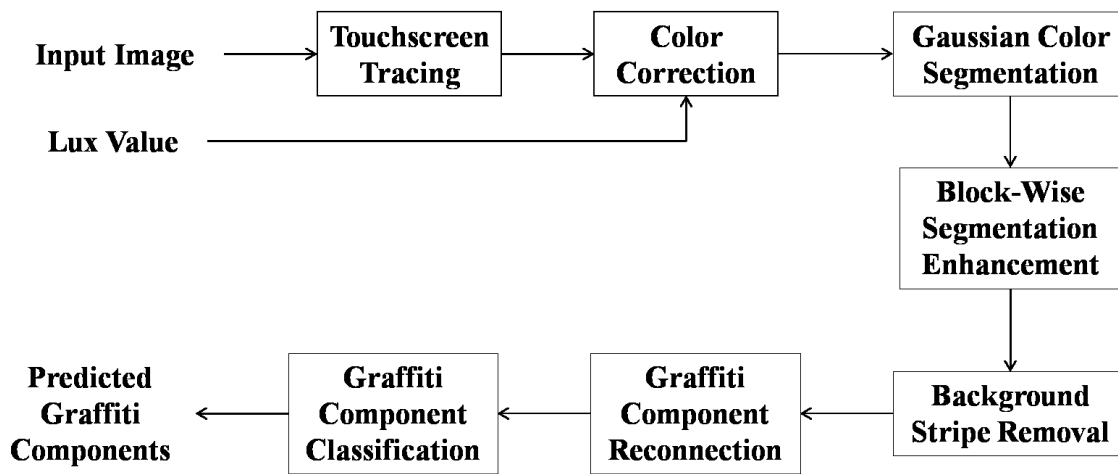


Fig. 5.21.: GARI End-To-End System.

We use the touchscreen tracing method to obtain the color median (either luma or hue) of a graffiti component, and we send this information to the server along with the image and the lux value automatically obtained from the device’s light sensor. Once on the server we color correct the image by mapping the lux value to a color correction matrix. We then use the color median to automatically segment the image using our proposed Gaussian thresholding method. The segmented image is locally enhanced, the existing background stripes are removed and the disjoint connected components are reconnected. The extracted components are gang graffiti component candidates that are classified and the predicted results are returned the mobile device.

We tested the entire process in 20 images with different colors, shapes, backgrounds, lighting conditions, and taken in different seasons (Summer and Winter). Figure 5.22 illustrates the 20 images.

Table 5.20 shows the running times of each step for all the test images. The processing times vary from 3.15 to 10.39 seconds, with a median of 4.69 seconds. Images 1016 and 1019 have two versions each because we segmented them using



Fig. 5.22.: Test Images for Automatic Gang Graffiti Segmentation.

different colors (i.e. two different touchscreen tracings). Figure 5.23 illustrates these cases. Figure 5.21 shows the running times of the three main blocks: color correction, image segmentation, and component classification. The high standard deviations of some steps indicate their dependence of the complexity of the input image. For example, the Graffiti Component Reconnection step evaluates each end-point of the image skeleton. The more complex the graffiti is, the more end-points it will have, and the more time it will take to process. Also, depending on the graffiti the number of segmented components vary from 4 to 31. This affects the running time of the Content Based Image Retrieval method following the segmentation process.

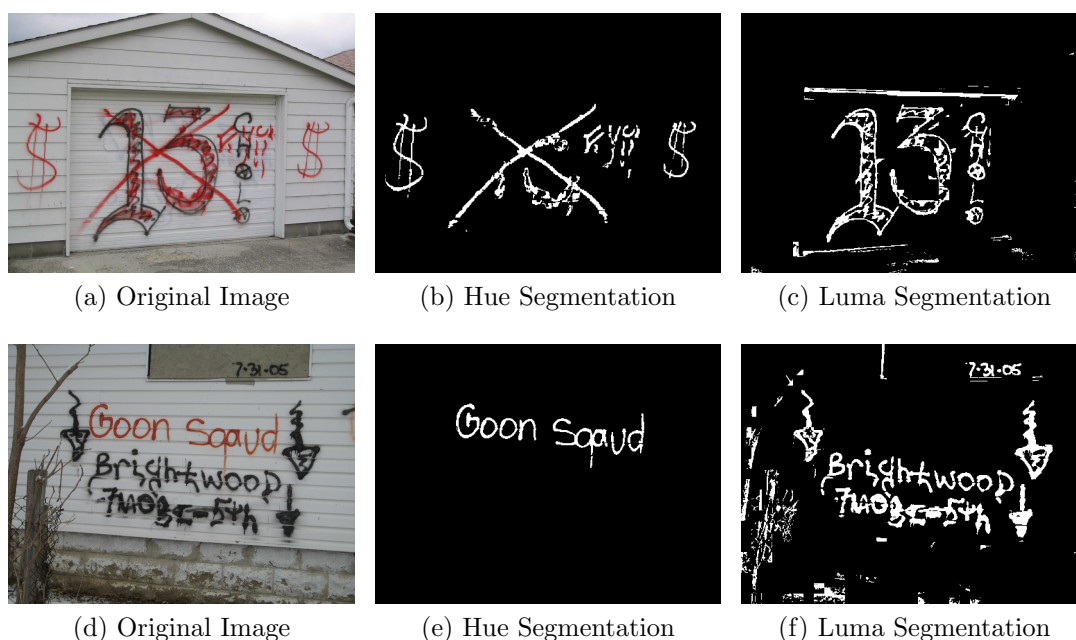


Fig. 5.23.: Images Segmented Separately From Two Different TouchScreen Tracings.

Figure 5.24 shows some examples of the proposed Color Image Segmentation Based on Gaussian Thresholding followed by Block-Wise Gaussian Segmentation Enhancement. Note that the enhancement contributes to both noise removal and graffiti component reconstruction. Figure C.22 shows some examples of our color image segmentation compared against other thresholding methods, including Niblack [20] (local thresholding) and Otsu [294] (global thresholding). For Niblack we set the

Table 5.20: Running Times (seconds) of Each Step in The GARI End-To-End System. 1: Color Correction Based on Mobile Light Sensor, 2: Color Image Segmentation Based on Gaussian Thresholding, 3: Block-Wise Gaussian Segmentation Enhancement, 4: Background Stripe Removal, 5: Graffiti Component Reconnection, 6: Graffiti Component Classification.

<b>Image Number</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>Total</b>
<b>1001</b>	1.72	0.24	0.88	0.05	2.05	0.64	5.57
<b>1002</b>	1.91	0.13	0.52	0.24	3.26	1.07	7.12
<b>1003</b>	1.85	0.28	0.94	0.05	0.91	0.99	5.04
<b>1004</b>	1.69	0.60	1.28	0.04	0.35	0.64	4.60
<b>1005</b>	2.27	0.17	0.79	0.03	0.28	0.64	4.18
<b>1006</b>	2.05	0.12	0.47	0.04	0.76	0.64	4.08
<b>1007</b>	1.71	0.13	0.65	0.04	0.34	0.28	3.15
<b>1008</b>	1.69	0.12	0.46	0.04	0.74	0.71	3.76
<b>1009</b>	1.73	0.25	0.62	0.04	0.75	0.99	4.39
<b>1010</b>	1.75	0.61	1.03	0.19	3.01	2.20	8.79
<b>1011</b>	1.87	0.19	0.62	0.07	5.86	1.78	10.39
<b>1012</b>	1.92	0.78	1.23	0.07	2.89	0.92	7.81
<b>1013</b>	1.70	0.20	0.85	0.04	0.75	0.50	4.04
<b>1014</b>	1.73	0.73	1.21	0.04	0.57	0.50	4.77
<b>1015</b>	1.67	0.76	1.19	0.05	2.00	1.07	6.73
<b>1016_1</b>	1.84	0.20	0.89	0.05	0.97	0.57	4.51
<b>1016_2</b>	1.80	0.19	0.61	0.05	1.10	0.43	4.17
<b>1017</b>	2.30	0.15	0.66	0.04	1.05	0.85	5.05
<b>1018</b>	1.86	0.14	0.73	0.04	0.39	0.43	3.58
<b>1019_1</b>	1.92	0.56	1.24	0.03	0.15	1.78	5.68
<b>1019_2</b>	1.71	0.55	1.05	0.09	2.51	0.36	6.27
<b>1020</b>	1.76	0.13	0.56	0.05	0.55	0.50	3.54
<b>Median</b>	<b>1.78</b>	<b>0.20</b>	<b>0.82</b>	<b>0.05</b>	<b>0.84</b>	<b>0.64</b>	<b>4.69</b>
<b>Std Dev</b>	0.17	0.24	0.27	0.05	1.38	0.50	1.85

Table 5.21: Running Times (seconds) of The Three Main Blocks in The GARI End-To-End System. 1: Color Correction, 2: Automatic Graffiti Component Segmentation, 3: Graffiti Component Classification. CCs: Number of Connected Components.

<b>Image Number</b>	<b>1</b>	<b>2</b>	<b>CCs</b>	<b>3</b>	<b>Total</b>
<b>1001</b>	1.72	3.22	9	0.64	5.57
<b>1002</b>	1.91	4.15	15	1.07	7.12
<b>1003</b>	1.85	2.19	14	0.99	5.04
<b>1004</b>	1.69	2.27	9	0.64	4.60
<b>1005</b>	2.27	1.27	9	0.64	4.18
<b>1006</b>	2.05	1.39	9	0.64	4.08
<b>1007</b>	1.71	1.16	4	0.28	3.15
<b>1008</b>	1.69	1.36	10	0.71	3.76
<b>1009</b>	1.73	1.66	14	0.99	4.39
<b>1010</b>	1.75	4.84	31	2.20	8.79
<b>1011</b>	1.87	6.74	25	1.78	10.39
<b>1012</b>	1.92	4.97	13	0.92	7.81
<b>1013</b>	1.70	1.84	7	0.50	4.04
<b>1014</b>	1.73	2.54	7	0.50	4.77
<b>1015</b>	1.67	4.00	15	1.07	6.73
<b>1016_1</b>	1.84	2.10	8	0.57	4.51
<b>1016_2</b>	1.80	1.95	6	0.43	4.17
<b>1017</b>	2.30	1.89	12	0.85	5.05
<b>1018</b>	1.86	1.29	6	0.43	3.58
<b>1019_1</b>	1.92	1.98	25	1.78	5.68
<b>1019_2</b>	1.71	4.20	5	0.36	6.27
<b>1020</b>	1.76	1.28	7	0.50	3.54
<b>Median</b>	<b>1.78</b>	<b>2.04</b>	<b>9</b>	<b>0.64</b>	<b>4.69</b>
<b>Std Dev</b>	0.17	1.52	7.04	0.50	1.85

filter radius to 25 pixels and standard deviation threshold to -0.2. When the graffiti surface has uniform texture and color all the methods produce good results. However, for complex surfaces and non-uniform illumination scenes Niblack and Otsu fail to segment the graffiti from the background. The only disadvantage of our proposed method is the running time. The average running times of Niblack and Otsu are 0.5 seconds and 0.01 seconds respectively, while our proposed method runs in 1 second on average. The comparison of the three methods for all 20 test images can be found on Appendix C. We also considered a stroke-width based image operator proposed in [311] to detect text in natural scenes, but it is not robust against non-alphanumeric symbols.

The Background Strip Removal process is the fastest of the four segmentation steps on average. This is because even though 18 of the 20 test images contain background strips only two of them still contain strips after the enhancement step. Figure 5.27 shows some examples of background strips removed during previous steps. Figure 5.26 shows the strip removal process in the two remaining images.

The Graffiti Component Reconnection process is the slowest of the four segmentation steps. This is because it conducts an exhaustive search among all the end-points on the image skeleton to find connection point candidates. Figure 5.28 shows an example of a test image where 252 end-points are checked in 5.86 seconds. Large amount of end-points are usually the results of skeletonization of background noise, such as trash on the ground or vegetation. Figure 5.29 shows some examples of successful component reconnection. Note that reconnection is not necessary when two end-points already belong to the same 8-neighbor connected component. Sometimes the distribution of the connected components is such that false connections are created, as shown in Figure 5.29b between the *1* in *2-1* and the *l* in *Almighty*.

To illustrate the effectiveness of the automatic gang graffiti segmentation Figure 5.30 shows examples of the number of 8-neighbor connected components after Color Image Segmentation Based on Gaussian Thresholding, and after Graffiti Component Reconnection. An additional step can be added to merge connected components





Fig. 5.24.: Examples of our proposed Color Image Segmentation Based on Gaussian Thresholding followed by Block-Wise Gaussian Segmentation Enhancement.

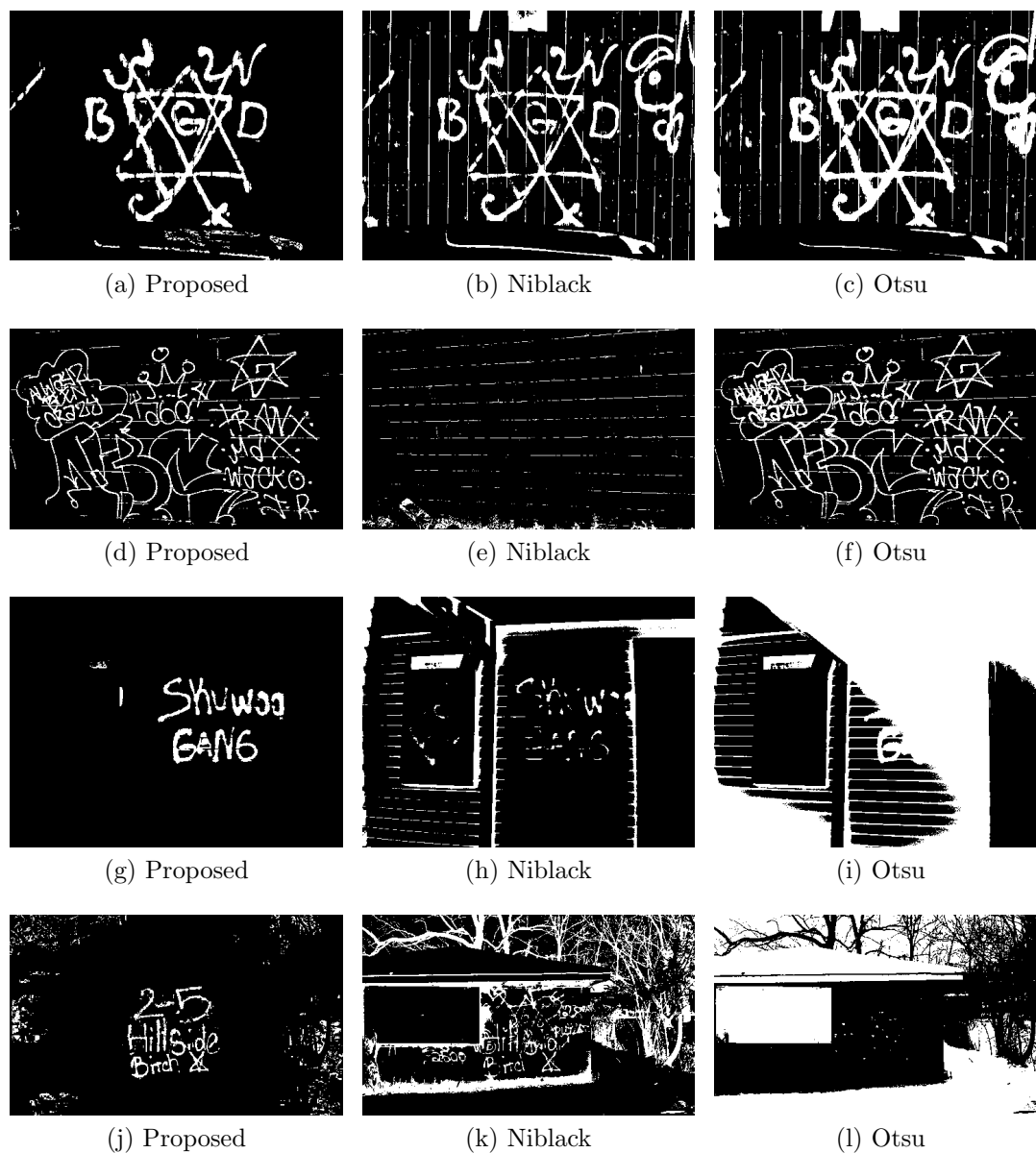
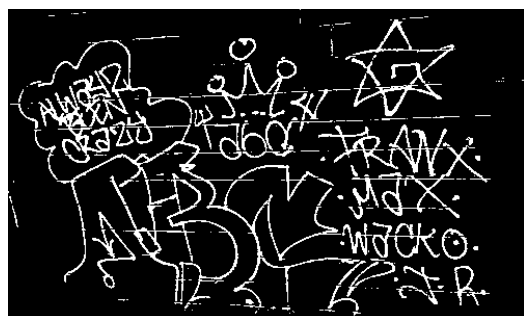
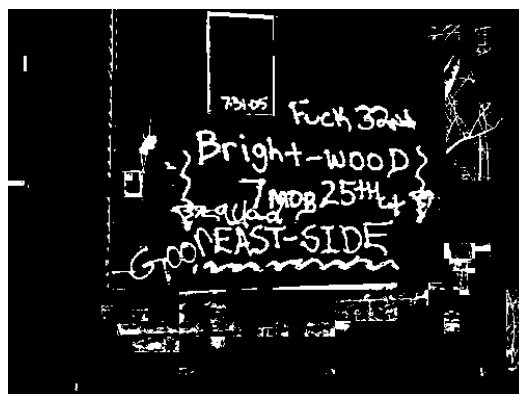


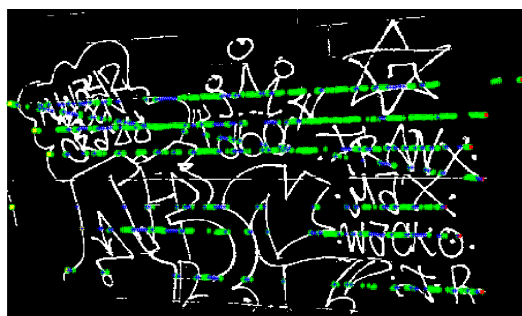
Fig. 5.25.: Comparison of our proposed color image segmentation method against Niblack and Otsu thresholding. From top to bottom: 1001, 1002, 1004, 1017.



(a) Enhanced



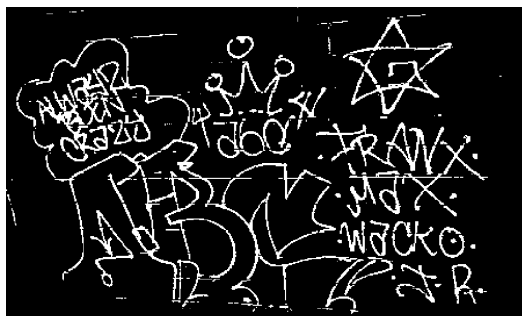
(b) Enhanced



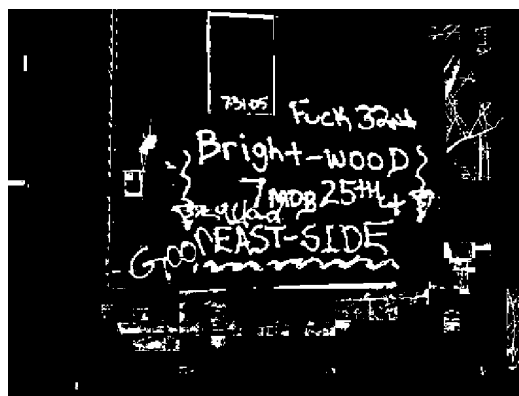
(c) Detected Strips



(d) Detected Strips



(e) Removed Strips



(f) Removed Strips

Fig. 5.26.: Examples of Background Strip Removal.



Fig. 5.27.: Examples of Background Strips Automatically Removed in Previous Steps.

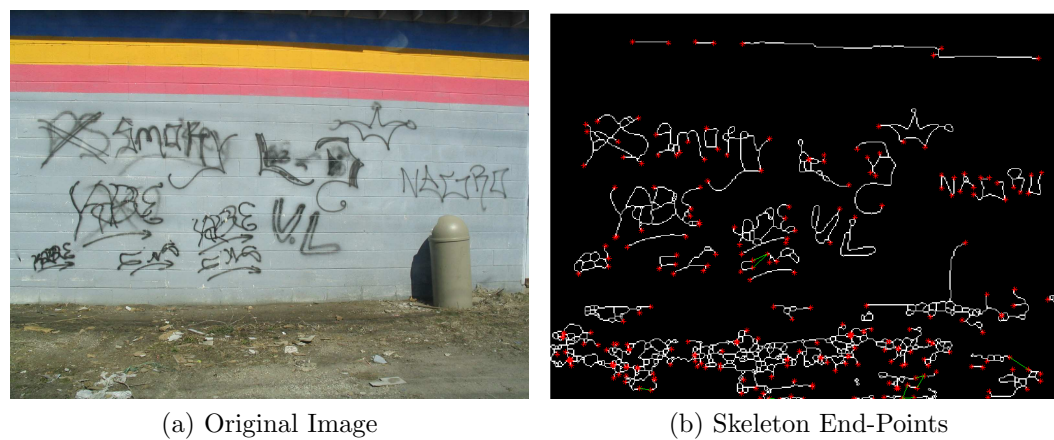


Fig. 5.28.: End-Points in Skeleton of Image 1011.

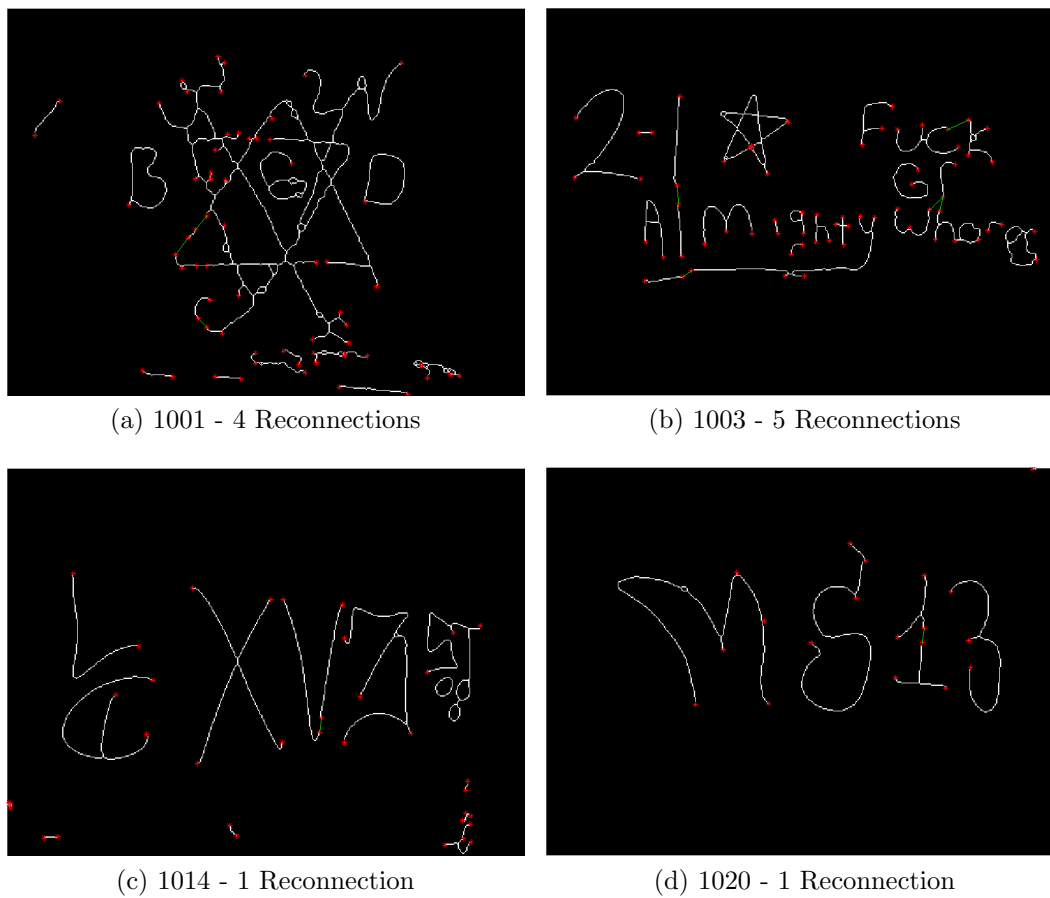


Fig. 5.29.: Examples of Graffiti Component Reconnection.

that may belong together forming words, as shown in Figure 5.31. Note how graffiti components are successfully segmented and can be now be treated separately for classification.

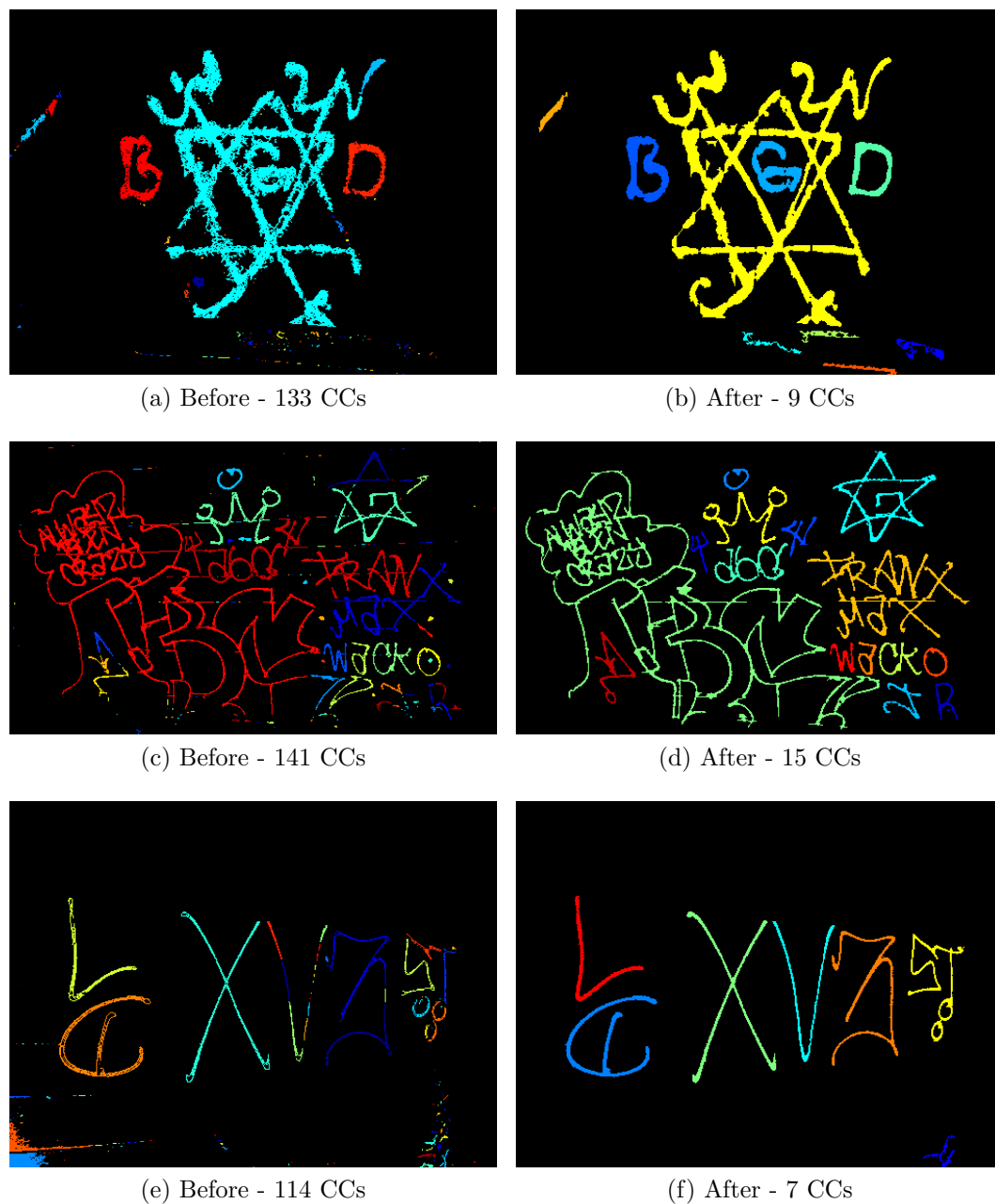
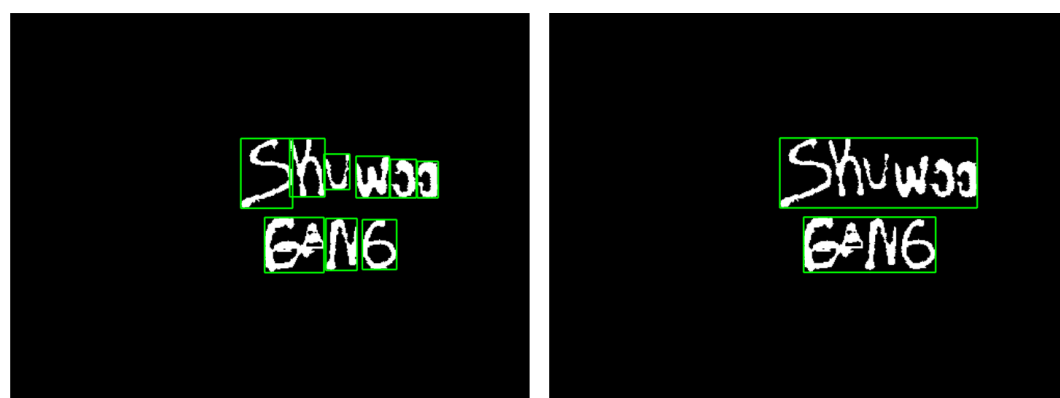


Fig. 5.30.: Number of Connected Components (CCs) Before and After Automatic Gang Graffiti Segmentation.



(a) Segmented Components

(b) Merged Components

Fig. 5.31.: Merged Connected Components Forming Words.

Each of the graffiti component candidates are independently classified to return a predicted class and a confidence score. The prediction class corresponds to one of the 14 trained classes, and the confidence score is the score given to the predicted class according to the equations presented in Section 5.1.3, in the range  $[0, 1]$ . Figures 5.32 to 5.34 show the classification results of one of the test images for each of its components, including component color, predicted class and confidence. Figure 5.35 shows a test image where gang graffiti components are found from two different colors (i.e. two different touchscreen traces). Note how even though one component is sprayed on top of the other we are able to recover the one on the back and successfully classify it. Further automatic interpretation can be done to understand that the component in the back has been crossed-out as a thread from a rival gang. Note that although some graffiti components have been successfully segmented they do not belong to any of the 14 classes we have trained. They are currently assigned to the closest class and given a low confidence score. For symbols that belong to the trained set we usually obtain a confidence higher than 0.60. Therefore, we can discard results if we do not achieve a minimum confidence score.

In the 20 test images there are a total of 98 gang graffiti components; 82 of them can be found in our set of trained classes. We are able to segment and isolate 75 of the 98 gang graffiti components, corresponding to 66 of the 82 recognizable components. We can then successfully classify 59 of them. The segmentation fails when either graffiti components are discarded or multiple graffiti components are merged into one. In all cases we are able to correctly identify the color of the graffiti component based on the median value of the color corrected touchscreen trace. That is, we have an end-to-end gang graffiti accuracy of 71.95%. The accuracy of each of the blocks is as follows: 100% color recognition accuracy, 76.56% automatic segmentation accuracy on color corrected images (80.49% for recognizable components), and 89.39% gang graffiti component classification accuracy on successfully segmented components. Table 5.22 show the accuracies of the automatic segmentation and graffiti component classification steps.





(a) Original Image

(b) Segmented Components



(c) Graffiti Component Candidates

Fig. 5.32.: Automatically Segmented Candidate Graffiti Components.

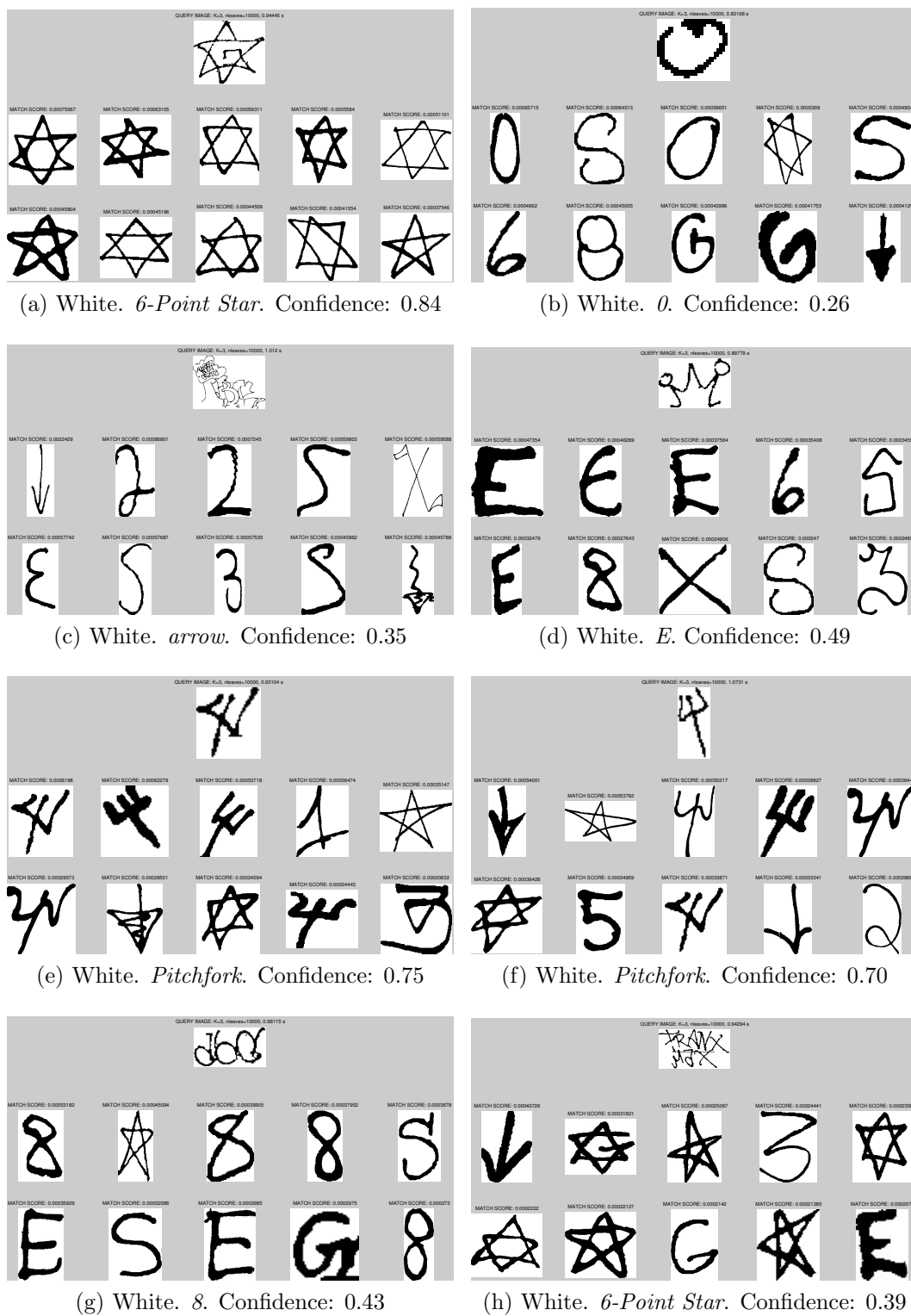


Fig. 5.33.: Classification Results and Top-10 Matches for Candidates 1 to 8.

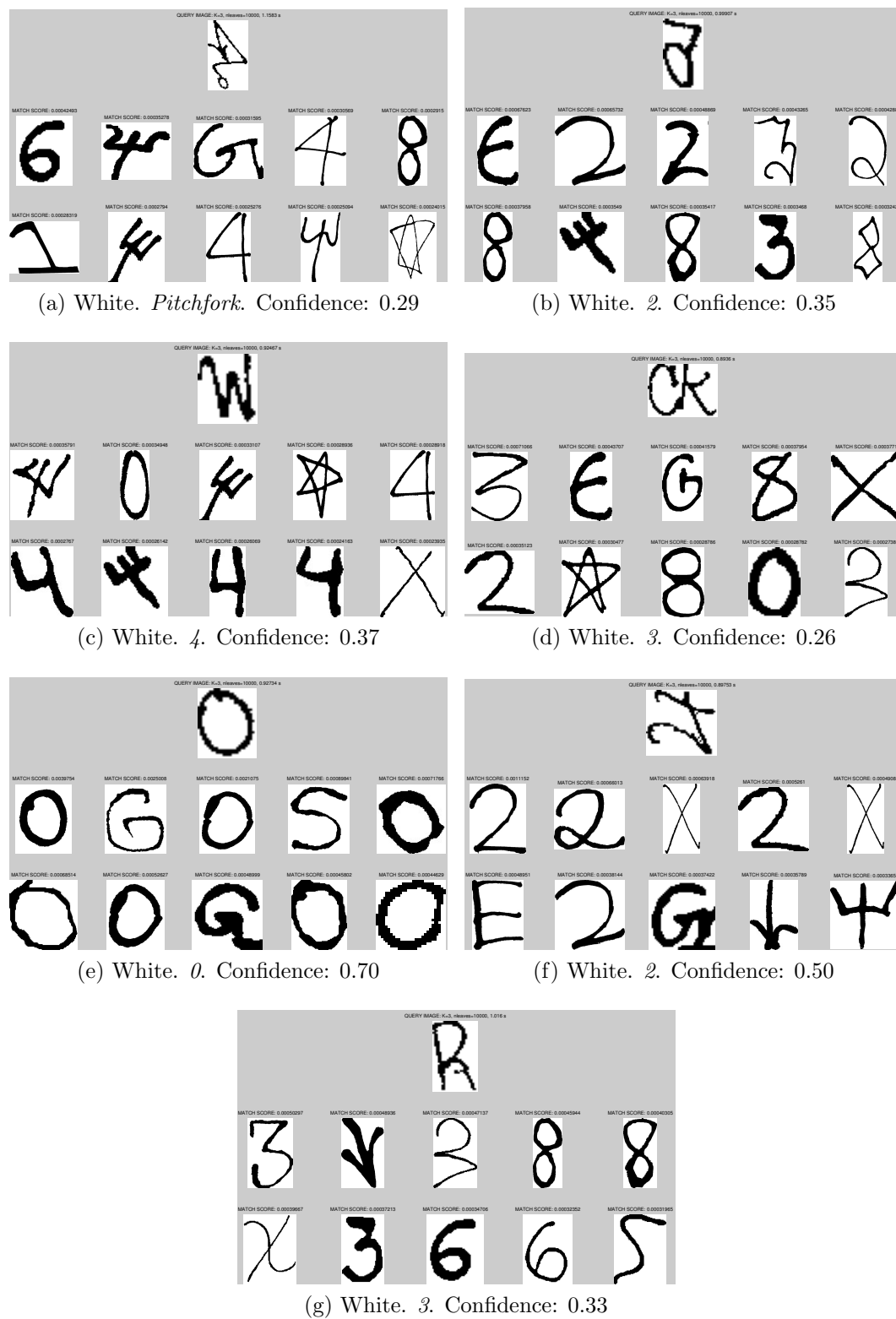
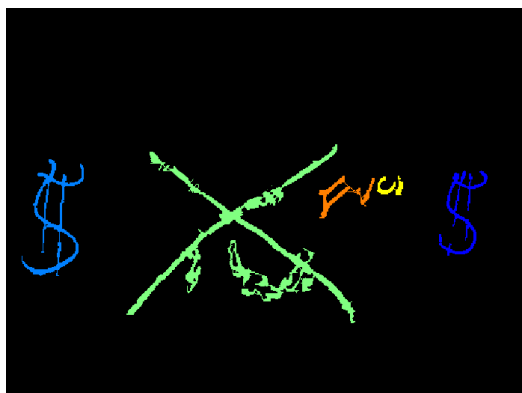


Fig. 5.34.: Classification Results and Top-10 Matches for Candidates 9 to 15.



(a) Original Image



(b) Segmented Components in Hue



(c) Segmented Components in Luma

(d) Red. *X*. Confidence: 0.72(e) Black. *1*. Confidence: 0.71(f) Black. *3*. Confidence: 0.67

Fig. 5.35.: Automatic Segmentation and Classification from Multiple Colors.

Table 5.22: Automatic Segmentation and Graffiti Component Classification Accuracies. N GC: Number of gang graffiti components. N GC Rec: Number of recognizable gang graffiti components.

Image Number	N GC	Segmented	N GC Rec	Segmented Rec	Classified
1001	6	4	4	2	2
1002	7	7	3	3	3
1003	5	5	4	4	3
1004	2	2	7	6	3
1005	3	3	8	7	6
1006	2	2	1	1	1
1007	2	2	0	0	0
1008	8	4	6	4	3
1009	4	4	3	3	3
1010	8	4	8	7	7
1011	5	5	0	0	0
1012	2	2	0	0	0
1013	7	4	6	3	3
1014	4	3	3	3	2
1015	2	2	2	2	2
1016_1	4	4	3	3	3
1016_2	2	2	2	2	2
1017	5	4	4	4	4
1018	9	1	6	0	0
1019_1	2	2	4	4	4
1019_2	5	5	5	5	5
1020	4	4	3	3	3
<b>Total</b>	<b>98</b>	<b>75</b>	<b>82</b>	<b>66</b>	<b>59</b>
<b>Accuracy</b>		<b>76.53%</b>		<b>80.49%</b>	<b>71.95%</b>
<b>Marginal Acc</b>					<b>89.39%</b>

Table 5.23: Average Running Times (seconds) and Accuracies of The Three Main Blocks in The GARI System on Testing Dataset.

	<b>Color Correction</b>	<b>Segmentation</b>	<b>Classification</b>	<b>End-To-End</b>
<b>Time</b>	1.78	2.04	0.64	4.69
<b>Accuracy</b>	100%	80.49%	89.39%	71.95%

Table 5.23 summarizes the results of the end-to-end system. The Color Correction time is based on the entire image and its accuracy is based on the touchscreen tracing results.

Table 5.24: Number of Images and Users In the Different GARI Systems.

	<b>GARI Classic</b>	<b>GARI IND</b>	<b>GARI CCSO</b>	<b>Total</b>
<b>Images</b>	720	595	173	1,488
<b>Users</b>	73	138	61	272

### 5.1.5 Database of Gang Graffiti

As of March 2014, our databases of gang graffiti images in the different GARI systems (GARI Classic, GARI IND, GARI CCSO) accumulate 1,488 browsable images with associated thumbnails and reduced size versions, for a total of 1.82 GB of data. We have a total of 272 users from more than 20 departments in the United States distributed across the GARI systems.

We cooperated with the Indianapolis Metropolitan Police Department (IMPD) to acquire a separate set of 657 graffiti images for research purposes. This allows us to be able to accurately calibrate and analyze the images. These include images acquired with and without using a tripod and with and without fiducial markers. We used three digital cameras for this purpose: a 10Mpx Canon Powershot S95, a 4Mpx Panasonic Lumix DMC-FZ4, and a 5Mpx HTC Desire (Android mobile telephone).

Table 5.24 shows the distribution of images and users across the three GARI systems.

### 5.1.6 Database Query Performance

We tested the elapsed time between sending an image from the hand-held device, using the Android application, and receiving the results of the upload. On the client side, the process includes sending and receiving the image to the server via HTTPS and returning the graffiti image thumbnail and text retrieved to the user. On the server side, the process includes creating a session for the user, checking image existence in the database, copying the image to a specific directory, creating the thumbnail image and reduced size copies of the image, extracting up to 24 EXIF data

points from the image, creating a new entry in the PostgreSQL table and adding information in as many as 30 fields, and sending back a string with the results of the upload. Table 5.25 shows the details of ten graffiti image uploads using the same network conditions (WiFi). As one can see most of the elapsed time is due to the HTTPS connection since the user interface operations on the hand-held device (for the specific action of uploading an image to the server) do not slow down the process.

Table 5.25: Elapsed Time On the Hand-Held Device and the Server When Uploading an Image.

<b>Image Size</b>	<b>Server Time</b>	<b>Total Time</b>
146.7 KB	0.66 s	2.24 s
157.9 KB	0.65 s	2.33 s
179.8 KB	0.65 s	2.66 s
203.3 KB	0.66 s	2.42 s
207.9 KB	0.64 s	2.44 s
227.8 KB	0.65 s	2.34 s
609.9 KB	1.05 s	3.64 s
639.8 KB	1.47 s	4.71 s
653.6 KB	1.06 s	4.00 s
760.4 KB	1.07 s	4.31 s

## 5.2 MERGE<sup>3</sup>

We did experiments for our three proposed methods from Section 4. The first experiment evaluates the accuracy of the sign location detection and color recognition of the segment detection using geometric constraints (see Section 4.2). The second experiment evaluates the accuracy of the sign detection, color recognition, and the saliency map methods of the convex quadrilateral detection based on saliency map (see Section 4.3). The third experiment evaluates the accuracy of the sign location detection of the sign detection based on Fourier descriptors (see Section 4.4). The tests were executed on a desktop computer with a 2.8GHz CPU and 2GB RAM.

---

<sup>3</sup>The work presented in this section was done by the author jointly with Bin Zhao and Kharittha Thongkor.



The ground-truth information included the sign distance from the camera, sign color, projective distortion of the sign, image resolution, possible shadow affecting the sign, and sign location on the image. Note that we only used the color and not the text of the sign for sign identification for these experiments. The image dataset consisted of 50 images each containing one or more hazmat signs (62 hazmat signs in total). Figure 5.36 illustrates some of the images in the dataset. The images were acquired by first responders using three different cameras: a 8.2 Mpx Kodak Easyshare C813, a 16 Mpx Nikon Coolpix S800c, and a 5 Mpx camera on an HTC Wildfire mobile telephone. The images were acquired in the field, under various lightning conditions, distances, and perspectives. Among the 50 images, 23 were acquired at 10-50 feet, 23 at 50-100 feet, and 4 at 100-200 feet. Among the 62 hazmat signs, 2 had low resolution, 11 had projective distortion, 8 were blurred, and 6 were shaded.

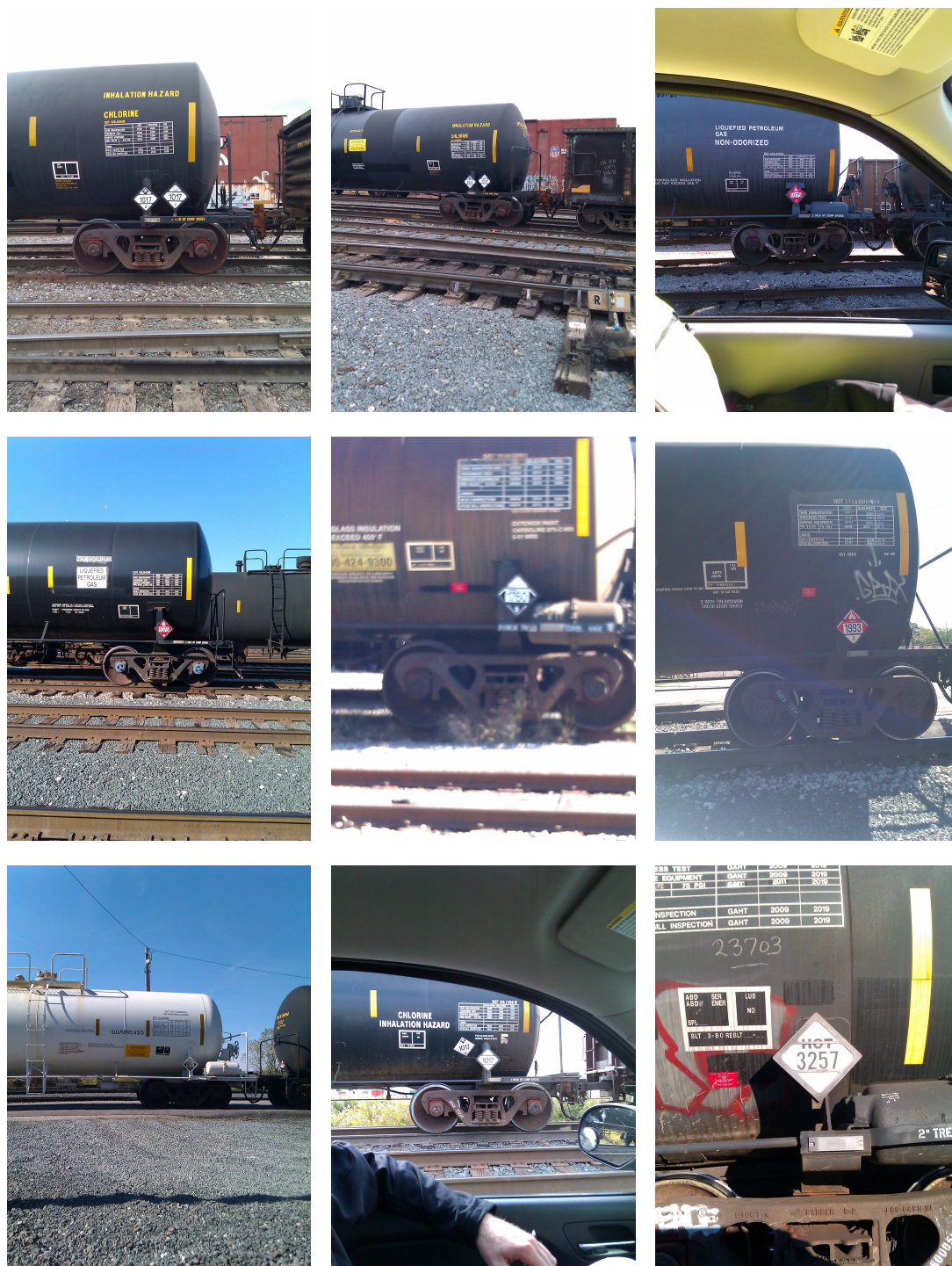


Fig. 5.36.: Example Images From The Test Dataset.

Table 5.26: Analysis Results: Segment Detection Using Geometric Constraints.

Total Signs	Signs Detected	Accuracy	Color Recognized	Accuracy
62	22	36.5%	12	19.4%

### 5.2.1 Segment Detection Using Geometric Constraints

The first experiment consisted of images from a dataset and manually comparing the results with ground-truth information. The method used for this experiment is segment detection using geometric constraints (see Section 4.2). Table 5.26 shows the results of the first experiment using our proposed method. We determined how many signs were successfully detected (*Signs Detected*) and how many were successfully identified (i.e., sign detected plus correct color (*Color Recognized*)). Note that the sign color recognition was done only if a sign was detected. Also note that although this method uses OCR on detected signs, its accuracy was not good enough to be tested on a wide range of images. Among the successfully detected signs we had a higher accuracy for color recognition. The proposed method recognized the correct color in 54.5% of the successfully detected signs. The low accuracy is caused by multiple factors, including segment overlapping, edge detection failure on low resolution images, distortion and rotation of the sign, and multi-colored signs. The proposed method had an average execution time of 2.30 seconds.

### 5.2.2 Convex Quadrilateral Detection Based on Saliency Map

The second experiment consisted of images from the same dataset from the first experiment, and manually comparing the results with ground-truth information. The method used for this experiment is convex quadrilateral detection based on saliency map (see Section 4.3). We did two experiments to investigate the speed and accuracy of our proposed method. The first experiment consisted of constructing saliency maps using different visual saliency models and evaluating their performance based on ground-truth information. The second experiment consisted of hazmat sign detec-

tion and recognition on our image dataset and manually comparing the results with ground-truth information.

Table 5.27 shows the results of our first experiment, including average execution times and scores. The saliency map methods evaluated in the experiment are: SBVA [312], GBVS [313], DVA [269], MSDA [271], IS [291], HFT [273]. We classified the resulting saliency maps into four categories: good, fair, bad, and lost. For each sign, we assigned 3 points to a good map (sign was mostly contained in a high saliency-valued region), 2 points to a fair map (sign was mostly contained in a middle saliency-valued region), 1 point to a bad map (sign was mostly contained in a low saliency-valued region), and 0 points to a lost map (sign was not contained in any saliency-valued region). Figure 5.37 illustrates examples of each category. The score of each saliency map method is calculated as the sum of the points assigned to all 62 hazmat signs, which ranges from 0 to 186. Compared with the SBVA and the GBVS methods using one color space, the IS and the HFT methods using one color space have comparable scores, while the IS and the HFT methods using two color spaces have higher scores. The IS(RGB+Lab), the HFT(RGB+Lab) and the IS+HFT(RGB+Lab) methods using two color spaces run 2.76, 1.93, and 1.14 times faster than the SBVA method and 4.48, 3.13, and 1.84 times faster than the GBVS method respectively. The results verified that the IS and the HFT methods can be combined to improve the score of IS+HFT method, while still running faster than SBVA and GBVS methods.

Table 5.28 shows the image analysis results of our second experiment. The overall sign detection accuracy is closely related to the number of pixels on a hazmat sign, which is mainly influenced by the distance from a camera in a mobile device to a hazmat sign and the resolution of the image captured by the camera. Compared with the proposed IS(RGB+Lab) and the HFT(RGB+Lab) methods using one saliency map method, our proposed IS+HFT(RGB+Lab) method using two saliency map methods has higher accuracy. The proposed IS+HFT(RGB+Lab) method has an overall sign detection accuracy of 64.5% for all 62 hazmat signs. Note that its

Table 5.27: Average Execution Time (in Seconds), Distribution and Score of Each Saliency Map Method (Color Spaces).

Saliency Map	Time	Good	Fair	Bad	Lost	Score
SBVA(I-RG-BY)	2.07	34	16	11	1	145
GBVS(I-RG-BY)	3.36	30	15	15	2	135
DVA(RGB)	0.43	19	2	11	30	72
MSDA(RGB)	3.74	22	7	27	6	107
IS(I-RG-BY)	0.43	23	4	17	18	94
IS(RGB)	0.36	45	8	4	5	155
IS(Lab)	0.39	27	5	20	10	111
HFT(I-RG-BY)	0.59	33	8	12	9	127
HFT(RGB)	0.53	38	5	8	11	132
HFT(Lab)	0.55	37	10	8	7	139
IS(RGB+Lab)	0.75	52	6	1	3	169
HFT(RGB+Lab)	1.08	41	6	8	7	143
<b>IS+HFT(RGB+Lab)</b>	1.83	55	4	2	1	175

overall accuracy is 71.9% for the 32 hazmat signs in the 50-100 feet range and 50.0% for the 6 hazmat signs in the 100-200 feet range. We can increase the overall accuracy by improving the adaptive thresholding method used in the saliency region segmentation and the morphological operations used in the convex quadrilateral shape detection. We determined the color recognition accuracy based on how many signs were correctly color recognized after a successful sign detection. The color recognition accuracies of the proposed methods using IS(RGB+Lab), HFT(RGB+Lab) and IS+HFT(RGB+Lab) are 37.1%, 30.6%, and 51.6% respectively. Note that the sign color recognition was done only if a sign was successfully detected, and that multi-colored signs may also cause our method to misidentify the sign color, given that we detect signs at individual color channels. Color recognition accuracy is affected by the absence of color calibration in the step of image preprocessing. The overall average execution times of the proposed methods using IS(RGB+Lab), HFT(RGB+Lab) and IS+HFT(RGB+Lab) are 2.60, 2.49, and 5.09 seconds in total respectively. The proposed IS+HFT(RGB+Lab) method is still suitable for real-time applications.

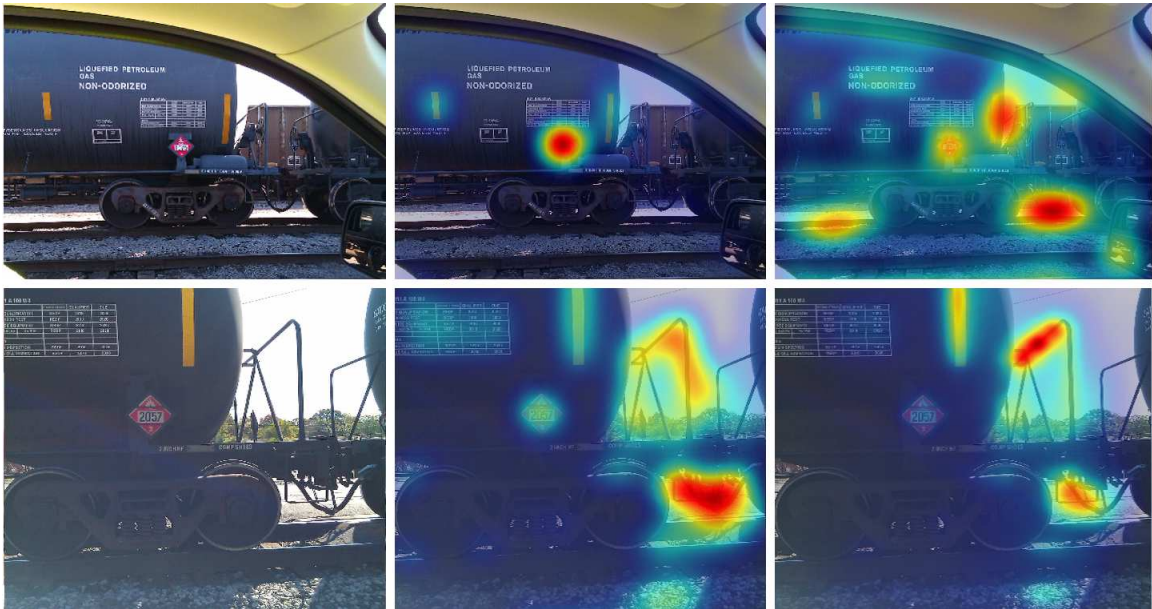


Fig. 5.37.: Saliency map categories (top to bottom, left to right): original image, good, fair; original image, bad, lost.

Table 5.28: Image Analysis Results: Convex Quadrilateral Detection Based on Saliency Map.

Proposed Method	Total Signs	Signs Detected	Overall Accuracy
IS(RGB+Lab)	62	32	51.6%
HFT(RGB+Lab)	62	24	38.7%
<b>IS+HFT(RGB+Lab)</b>	62	40	64.5%

Table 5.29: Analysis Results: Sign Location Detection Based on Fourier Descriptors.

<b>Total Signs</b>	<b>Signs Detected</b>	<b>Accuracy</b>
62	45	72.6%

Table 5.30: Image Analysis Results for the Three Proposed Methods. 1: Segment Detection Using Geometric Constraints, 2: Convex Quadrilateral Detection Based on Saliency Map, 3: Sign Location Detection Based on Fourier Descriptors.

<b>Proposed Method</b>	<b>Total Signs</b>	<b>Signs Detected</b>	<b>Overall Accuracy</b>	<b>Time</b>
1	62	22	36.5%	2.30
2	62	40	64.5%	5.09
3	62	45	72.6%	6.11

### 5.2.3 Sign Location Detection Based on Fourier Descriptors

We implemented the methods in [257] and our previous technique [314] and compared their accuracy against our method. Table 5.29 shows the results. Our method has a hazmat sign location detection rate of 72.58%, while the detection rates for [257] and [314] are 24.32% and 64.52%, respectively. Figure 5.38 illustrates some examples of sign location detection for each of the methods. The proposed method had an average execution time of 6.11 seconds.

Table 5.30 shows the analysis results for each of the three proposed methods for hazmat sign detection.



Fig. 5.38.: Examples of sign location detection. Column from left to right: results from [257], results from [314], results from proposed method.



## 6. CONCLUSIONS AND FUTURE WORK

### 6.1 Conclusions

In this thesis two integrated mobile systems are described. First, a system for gang graffiti image acquisition and recognition. We called this system Gang Graffiti Automatic Recognition and Interpretation or GARI. GARI includes motion blur prevention and detection, color correction based on light sensor, color recognition based on touchscreen tracing, color image segmentation based on Gaussian thresholding, and content-based gang graffiti image retrieval. We have also investigated the design and deployment of an integrated image-based database system. Second, a system for hazmat sign detection and recognition. We called this system Mobile Emergency Response Guidebook or MERGE. MERGE includes segment detection using geometric constraints, convex quadrilateral detection based on saliency map, and sign location detection based on Fourier descriptors.

The main contributions of GARI and MERGE in the area of image analysis are as follows:

- We presented a motion blur prevention and detection method based on mobile device sensors.
- We presented a color correction method based on mobile device light sensor.
- We described a color recognition method based on touchscreen tracing.
- We presented a color image segmentation method based on Gaussian thresholding, block-wise Gaussian segmentation enhancement, background stripe removal, and connected component reconnection.

- We presented a feature extraction method based on local shape context descriptors from SIFT keypoint locations.
- We presented a gang graffiti content based image retrieval method based on bag-of-words model.
- We presented a segment detection method based on geometric constraints.
- We presented a convex quadrilateral detection method based on saliency map.
- We presented a sign location detection based on Fourier descriptors.

The main contributions of GARI and MERGE in the design and deployment of the integrated image-based database system are as follows:

- We developed an integrated image-based database system where data from users and images is connected to gang graffiti information for analysis and tracking.
- We developed an integrated image-based database system where data from users and images is connected to hazmat sign information for image analysis and forensics.
- We created a web-based interface for first responders and researchers to upload images and browse gang related information by location, date and time, using interactive maps for better visualization. It is accessible from any device capable of connecting to the Internet, including iPhone and Blackberry.
- We created a web-based interface for first responders and researchers to upload images and browse hazardous material information by location, date and time for forensic analysis. It is accessible from any device capable of connecting to the Internet, including iPhone and Blackberry.
- We created Android and iOS applications for first responders on the field to upload images to the server, use image analysis and conduct forensic tasks, browse related information, and use location-based services to populate interactive maps.

## 6.2 Project Status

As of March 2014 we have developed Android and iOS applications and a web-based interface for both the GARI and MERGE systems. The GARI Android/iOS applications include color recognition, image acquisition and upload, content based image retrieval, and database browsing through lists, interactive maps and augmented reality interfaces. The GARI web-based interface includes image upload and database browsing through lists and interactive maps. The MERGE Android/iOS applications include sign recognition and interpretation and internal database browsing using the 2012 version of the Emergency Response Guidebook (ERG). The MERGE web-based interface includes the same capabilities. Both GARI and MERGE web-based interfaces can be accessed from any device capable of connecting to the Internet (e.g., Blackberry, laptop/desktop computers).

Table 6.1 shows the Android/iOS versions of the GARI and MERGE mobile applications as of March 2014. Note that GARI has multiple versions, since it has been deployed for different Police Departments across the country. GARI Classic and GARI Classic Test are versions based at Purdue University and used for testing purposes. GARI IND is used by the Indianapolis Metropolitan Police Department (IMPD). GARI CCSO is used by the Cook County Police Department (CCPD). CGAP stands for Citizen Gang Alert Program. It will be released to the public so regular citizens can report gang graffiti directly to the police.

Table 6.1: Android/iOS versions of the GARI and MERGE mobile applications.

	<b>Android</b>	<b>iOS</b>
<b>GARI Classic</b>	2.84 - February 2014	1.3 - November 2013
<b>GARI Classic Test</b>	2.76TEST - February 2014	1.3TEST - November 2013
<b>GARI IND</b>	2.76IND - February 2014	1.4IND - January 2014
<b>GARI CCSO</b>	2.76CCSO - February 2014	1.3CCSO - November 2013
<b>CGAP</b>	1.16 - February 2014	1.3 - November 2013
<b>MERGE</b>	3.0 - February 2014	1.5 - March 2014

Our current image analysis system for GARI includes five methods. First, mobile-based motion blur prevention and detection. Second, color correction based on mobile light sensor. Third, color recognition based on touchscreen tracing. Fourth, automatic graffiti component segmentation, which includes color image segmentation based on Gaussian thresholding, block-wise Gaussian segmentation enhancement, background stripe removal, and graffiti component reconnection. Fifth, content based gang graffiti image retrieval. The first two are done on the client, while the last three are currently done on the server.

Our current image analysis system for MERGE includes three methods, all done on the server. First, segment detection using geometric constraints. Second, convex quadrilateral detection based on saliency map. Third, sign location detection based on Fourier Descriptors.

Our tests on database query performance for GARI suggest that the bottleneck for the upload and retrieval process is from the network connection. This is because we require the full resolution image, which can be up to several MB of data, to be sent to the server for analysis.

Our databases of gang graffiti images in the various GARI systems (GARI Classic, GARI IND, GARI CSSO) have 1,488 browsable images with associated thumbnails and reduced size versions (total of 1.82 GB of data). We have also acquired 657 images for research purposes. The Android and iPhone applications have a memory size of 6.4 MB and 1.7 MB respectively. The CGAP version of the application requires only 1.1 MB and 779 KB respectively.

Our proposed color correction method based on the mobile light sensor has proved to be faster than using fiducial markers and more accurate than using a fiducial marker every week. Our accuracy and speed tests for the content based gang graffiti image retrieval for GARI were done in two scenarios: scene recognition and gang graffiti component classification. The experimental results showed that using SIFT descriptors for scene recognition and LSC descriptors for component classification produce very accurate outcomes. The experiments also showed that the image retrieval is fast

in both scenarios. The end-to-end system has an accuracy of 71.95% and an average execution time of 4.69 seconds as follows: 100% color recognition accuracy, 80.49% automatic segmentation accuracy on color corrected images, and 89.39% gang graffiti component classification accuracy on successfully segmented components.

Our image analysis tests for MERGE showed that the sign location detection based on Fourier Descriptors is more accurate than the convex quadrilateral detection based on saliency map method and the segment detection using geometric constraints. Although it runs slower, its average execution time of 6.11 seconds makes it suitable for real-time operation.

## 6.3 Future Work

### 6.3.1 GARI

Although the Color Correction Based on Mobile Light Sensor achieves good accuracy the current method to associate a color correction matrix  $M$  to a lux value is through a lookup table. We should investigate automatic generation of color correction matrices from the lux value by describing the evolution of the elements in  $M$  with the lightning step. Figure 6.1 illustrates such evolution with the current number of lux samples (612).

Our experiments shown that the bottleneck for the upload and retrieval process is the network connection. Therefore, we could pre-process the image on the mobile device to reduce the amount of data to be sent to the server. In this case we would need to investigate the trade-offs between battery life, network bandwidth, storage capacity, and processor performance [315,316].

The Block-Wise Gaussian Segmentation Enhancement currently uses a fixed block size for local image processing. In the future we could improve the enhancement by adapting the block size to the local width of the graffiti component. We can use the Stroke Width Transform (SWT) proposed in [311] for this purpose.

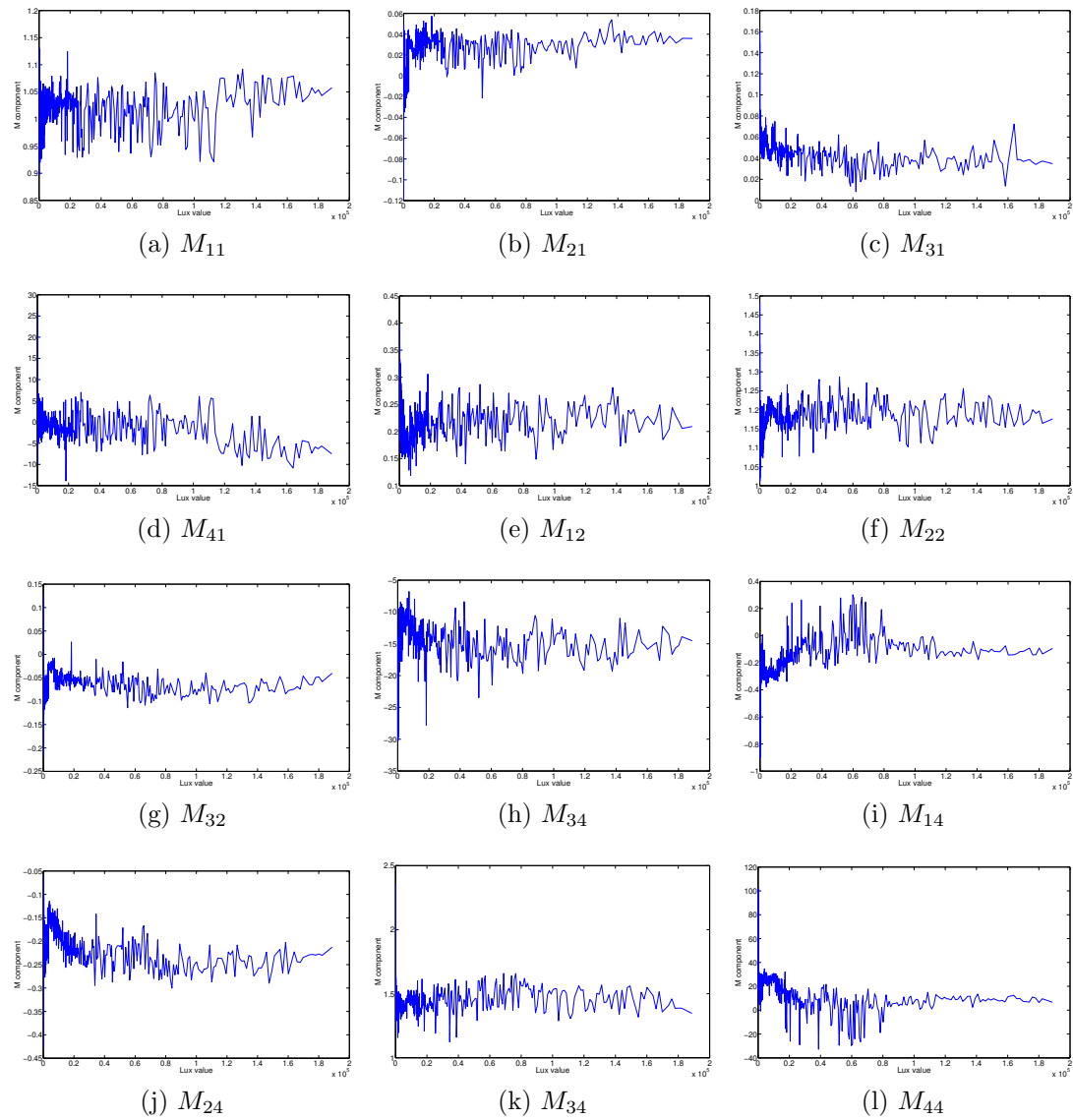


Fig. 6.1.: Evolution of the Elements in  $M$  With the Lightning Step (Lux Value).

Our Gang Graffiti Component Classification method is currently able to predict 14 different classes. This is because we want to have at least 15 samples of a particular class to ensure a minimum confidence. In the future, when more images are available from our users we will have more ground-truth samples to extend the number of classes. With more ground-truth data we can also investigate new features for graffiti component classification, such as Zernike moments (global and local) or the curvature scale space descriptors (CSSD) found in MPEG-7.

On the client side, the Android/iPhone users can help improve the classification system by manually correcting the predicted results. The corrections can be sent back to the server and used to automatically retrain the vocabulary tree to account for the changes.

When two or more graffiti components are merged (e.g., 6-point star with pitchforks) we are not currently able to classify them as separate objects. In fact, the new merged component may not be classified as any of the individual sub-components contained in it. We could investigate methods to retrieve multiple objects from a single entity, such as [317] or [318].

Even though our image retrieval methods achieve high accuracy, the procedure to obtain the vocabulary tree involves the segmentation of a high-dimensional space in hierarchical clusters using k-means clustering. This can cause unwanted results due to effects of the curse of dimensionality [237, 238]. We may want to investigate other methods that are more reliable. A tree-like structure can be built by repeatedly projecting the set of  $\mathbb{R}^{128}$  descriptors into  $\mathbb{R}$  using a normalized random vector  $v \in \mathbb{R}^{128}$  until the projection can be clearly separated into two regions or classes. We can use the same method recursively until we obtain the desired number of classes. The resulting tree can then act as a vocabulary tree.

The final output of our current end-to-end system is a list of candidate gang graffiti components and their confidence scores. We can create associations between graffiti components and their descriptions in order to improve the interpretation and help first responders identify gangs, gang members, and track gang activity. However, this is

not as easy as creating a table with one to one correspondences between components and descriptions. Depending on the geographical location of the graffiti the same graffiti component can have different meanings. Although we do not have direct evidence, this may be also true for colors. A more comprehensive database could also include information related to the locations of graffiti components with respect to each other to provide more context information. Also, we can enlarge the number of fields and relationships in the database so as to link gangs to their respective colors, acronyms, gang members, locations, or activity over time.

### **6.3.2 MERGE**

Our long term goal for MERGE is to develop a system based on a mobile device such as a mobile telephone, capable of using location-based services, combined with image analysis, to automatically detect hazardous material signs from images taken up to 500 feet, and provide real-time information to first responders to identify the hazardous materials and determine what specialty equipment, procedures and precautions should be taken in the event of an emergency. This can be done by improving our current sign location detection method and use a more robust color recognition technique. We can also combine the saliency map method from Convex Quadrilateral Detection Based on Saliency Map with the shape descriptors from Sign Location Detection Based on Fourier Descriptors in one method.

We can use the same color correction and blur detection methods from GARI to improve the color recognition and reduce the impact of motion blur. An optical character recognition method would help interpret the text inside the hazmat signs when we have enough image resolution. We can also investigate color recognition methods for multi-colored signs.



## 6.4 Publications Resulting From This Work

### Conference Papers

1. Bin Zhao, **Albert Parra** and Edward J. Delp, “Mobile-Based Hazmat Sign Detection System,” *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 735-738, December 2013, Austin, TX.
2. **Albert Parra**, Bin Zhao, Joonsoo Kim and Edward J. Delp, “Recognition, Segmentation and Retrieval of Gang Graffiti Images on a Mobile Device,” *Proceedings of the IEEE International Conference on Technologies for Homeland Security*, pp. 178-183, November 2013, Waltham, MA.
3. **Albert Parra**, Bin Zhao, Andrew Haddad, Mireille Boutin and Edward J. Delp, “Hazardous Material Sign Detection and Recognition,” *Proceedings of the IEEE International Conference on Image Processing*, pp. 2640-2644, September 2013, Melbourne, Australia.
4. **Albert Parra**, Mireille Boutin and Edward J. Delp, “Location-Aware Gang Graffiti Acquisition and Browsing on a Mobile Device,” *Proceedings of the IS&T/SPIE Electronic Imaging on Multimedia on Mobile Devices*, pp. 830402-1-13, January 2012, San Francisco, CA.

## LIST OF REFERENCES

## LIST OF REFERENCES

- [1] *ERG*. [www.phmsa.dot.gov/hazmat/library/erg](http://www.phmsa.dot.gov/hazmat/library/erg)
- [2] A. Parra, “An integrated mobile system for gang graffiti image acquisition and recognition,” M.S. Thesis, Purdue University, West Lafayette, IN, December 2011.
- [3] “Graffiti Tracker.” [graffititracker.net](http://graffititracker.net)
- [4] “Tracking and Automated Graffiti Reporting System.” [www.594graffiti.com](http://www.594graffiti.com)
- [5] “Graffiti Reduction & Interception Program.” [www.gripsystems.org](http://www.gripsystems.org)
- [6] “Graffiti Tracking System.” [www.graffititrackingsystem.com](http://www.graffititrackingsystem.com)
- [7] A. K. Jain, J.-E. Lee, and R. Jin, “Graffiti-ID: Matching and retrieval of graffiti images,” *Proceedings of the 1st ACM Workshop on Multimedia in Forensics*, pp. 1–6, October 2009, Beijing, China.
- [8] W. Tong, J.-E. Lee, R. Jin, and A. K. Jain, “Gang and moniker identification by graffiti matching,” *Proceedings of the 3rd ACM Workshop on Multimedia in Forensics and Intelligence*, pp. 1–6, November 2011, Scottsdale, AZ.
- [9] A. Jain, J. Lee, and R. Jin, “Tattoo-ID: Automatic tattoo image retrieval for suspect and victim identification,” *Advances in Multimedia Information Processing, PCM*, pp. 256–265, December 2007, Hong Kong, China.
- [10] J.-E. Lee, A. Jain, and R. Jin, “Scars, marks and tattoos (SMT): Soft biometric for suspect and victim identification,” *Proceedings of the Biometrics Symposium*, pp. 1–8, September 2008, Tampa, FL.
- [11] A. K. Jain, J.-E. Lee, R. Jin, and N. Gregg, “Content-based image retrieval: An application to tattoo images,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 2745–2748, November 2009, Cairo, Egypt.
- [12] J.-E. Lee, R. Jin, A. K. Jain, and W. Tong, “Image retrieval in forensics: Tattoo image database application,” *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 40–49, 2012.
- [13] A. Jain, R. Jin, and J.-E. Lee, “Tattoo image matching and retrieval,” *IEEE Transactions on Computers*, vol. 45, no. 5, pp. 93–96, May 2012.
- [14] H. Han and A. Jain, “Tattoo based identification: Sketch to image matching,” *Proceedings of the International Conference on Biometrics (ICB)*, pp. 1–8, June 2013, Madrid, Spain.

- [15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, November 2004, Hingham, MA.
- [16] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, October 2000, Los Alamitos, CA.
- [17] C. Yang, P. C. Wong, W. Ribarsky, and J. Fan, "Efficient graffiti image retrieval," *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, pp. 36:1–36:8, June 2012, Hong Kong, China.
- [18] D. Manger, "Large-scale tattoo image retrieval," *Proceedings of the Conference on Computer and Robot Vision*, pp. 454–459, May 2012, Toronto, Canada.
- [19] M. Zarem, E. Vuillermet, and J. DeAguiar, "Intelligent reverse geocoding," August 2007, US Patent App. 11/367,911.
- [20] W. Niblack, *An Introduction to Digital Image Processing*. Prentice-Hall, 1986.
- [21] WISER. [wiser.nlm.nih.gov](http://wiser.nlm.nih.gov)
- [22] D. Gossow, J. Pellenz, and D. Paulus, "Danger sign detection using color histograms and SURF matching," *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 13–18, October 2008, Sendai, Japan.
- [23] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, M. Andriluka, O. Schwahn, U. Klingauf, S. Roth, B. Schiele, and O. Stryk, "A semantic world model for urban search and rescue based on heterogeneous sensors," *Proceedings of the 14th RoboCup International Symposium*, vol. 6556, pp. 180–193, June 2010, Singapore, Singapore.
- [24] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Journal of Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.
- [25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, June 2005, San Diego, CA.
- [26] GARI. [www.gang-graffiti.org](http://www.gang-graffiti.org)
- [27] MERGE. [www.hazmat-signs.org](http://www.hazmat-signs.org)
- [28] National Gang Intelligence Center (NGIC), *2011 National Gang Threat Assessment - Emerging Trends*. United States Department of Justice, April 2011.
- [29] National Drug Intelligence Center (NDIC), *Attorney General's Report to Congress on the Growth of Violent Street Gangs in Suburban Areas*. United States Department of Justice, April 2008.
- [30] J. Kim, A. Parra, and E. J. Delp, "Tattoo image matching using local and global shape context," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, October 2014, Paris, France (submitted).

- [31] Japan Electronic Industry Development Association (JEIDA), "Design rule for camera file system, version 1.0." 1998.
- [32] D. Ley and R. Cybriwsky, "Urban graffiti as territorial markers," *Annals of the Association of American Geographers*, vol. 64, no. 4, pp. 491–505, December 1974.
- [33] J. Ferrell, *Crimes of Style: Urban Graffiti and the Politics of Criminality*. Garland, New York, 1993.
- [34] W. Miller, *Crime by Youth Gangs and Groups in the United States*. U.S. Dept. of Justice, Office of Justice Programs, Office of Juvenile Justice and Delinquency Prevention, 1992.
- [35] United States Department of Transportation, *Code of Federal Regulations, Title 49, DOT Hazmat*. Labelmaster, October 2012.
- [36] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.
- [37] R. L. Lagendijk and J. Biemond, *The Image and Video Processing Handbook*. Academic Press, 1999, ch. Basic methods for image restoration and identification, pp. 125–139.
- [38] R. Y. Landge and R. Sharma, "Blur detection methods for digital images - A survey," *International Journal of Computer Applications Technology and Research*, vol. 2, no. 4, pp. 494–498, 2013.
- [39] J. Ko and C. Kim, "Low cost blur image detection and estimation for mobile devices," *Proceedings of the International Conference on Advanced Communication Technology*, vol. 03, pp. 1605–1610, February 2009, Phoenix Park, Ireland.
- [40] B. Cardani, "Optical image stabilization for digital cameras," *IEEE Transactions on Control Systems*, vol. 26, no. 2, pp. 21–22, April 2006.
- [41] J.-H. Moon and S. Y. Jung, "Implementation of an image stabilization system for a small digital camera," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 206–212, May 2008.
- [42] S. Nasiri, M. Kiadeh, Y. Zheng, S. Lin, and S. Shi, "Optical image stabilization in a digital still camera or handset," May 2012, US Patent 8,170,408.
- [43] A. Ciancio, A. L. N. T. da Costa, E. A. B. Da Silva, A. Said, R. Samadani, and P. Obrador, "No-reference blur assessment of digital pictures based on multifeature classifiers," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 64–75, January 2011.
- [44] C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Low complexity image quality measures for dietary assessment using mobile devices," *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, pp. 351–356, December 2011, Dana Point, CA.
- [45] X. Marichal, W. Ma, and H. Zhang, "Blur determination in the compressed domain using DCT information," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 386–390, October 1999, Kobe, Japan.

- [46] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, January 1974.
- [47] H. Tong, M. Li, H. Zhang, and C. Zhang, "Blur detection for digital images using wavelet transform," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 17–20, June 2004, Taipei, Taiwan.
- [48] P. Porwik and A. Lisowska, "The haar-wavelet transform in digital image processing: its status and achievements," *Machine graphics and vision*, vol. 13, no. 1/2, pp. 79–98, 2004.
- [49] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "Perceptual blur and ringing metrics: Application to JPEG2000," *Signal Processing and Image Communication*, vol. 19, no. 2, pp. 163–172, 2004.
- [50] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley-Interscience, 1973.
- [51] N. Narvekar and L. Karam, "A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2678–2683, March 2011.
- [52] O. Šindelář and F. Šroubek, "Image deblurring in smartphone devices using built-in inertial measurement sensors," *Journal of Electronic Imaging*, vol. 22, no. 1, pp. 011 003:1–011 003:8, 2013.
- [53] P. R. Sanketi and J. M. Coughlan, "Anti-blur feedback for visually impaired users of smartphone cameras," *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 233–234, 2010, Orlando, FL.
- [54] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Transactions on Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, September 2001.
- [55] G. Sharma and R. Bala, *Digital color imaging handbook*. CRC press, 2002.
- [56] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: Survey and experiments," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2475–2489, September 2011.
- [57] M. Bleier, C. Riess, S. Beigpour, E. Eibenberger, E. Angelopoulou, T. Troger, and A. Kaup, "Color constancy and non-uniform illumination: Can existing algorithms work?" *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 774–781, November 2011, Barcelona, Spain.
- [58] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms - Part II: Experiments with image data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 985–996, September 2002.
- [59] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin Institute*, vol. 310, no. 1, pp. 1–26, 1980.

- [60] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, September 2007.
- [61] C. Xu, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Image enhancement and quality measures for dietary assessment using mobile devices," *Proceedings of the IS&T/SPIE Conference on Computational Imaging X*, vol. 8296, pp. 82 960Q–1–82 960Q–10, January 2012, San Francisco, CA.
- [62] S. Srivastava, C. Xu, and E. J. Delp, "White synthesis with user input for color balancing on mobile camera systems," *Proceedings of the IS&T/SPIE Conference on Multimedia on Mobile Devices*, vol. 8304, pp. 83 040F:1–83 040F:8, January 2012, Burlingame, CA.
- [63] D. A. Forsyth, "A novel algorithm for color constancy," *International Journal of Computer Vision*, vol. 5, no. 1, pp. 5–36, August 1990.
- [64] G. Finlayson, "Color in perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1034–1038, October 1996.
- [65] G. Finlayson and S. Hordley, "Improving gamut mapping color constancy," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1774–1783, October 2000.
- [66] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms - Part I: Methodology and experiments with synthesized data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 972–984, September 2002.
- [67] H. Joze and M. Drew, "White patch gamut mapping colour constancy," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 801–804, September 2012, Orlando, FL.
- [68] G. Finlayson, S. Hordley, and P. Hubel, "Color by correlation: a simple, unifying framework for color constancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1209–1221, November 2001.
- [69] C. Rosenberg, M. Hebert, and S. Thrun, "Color constancy using KL-divergence," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 239–246, July 2001, Vancouver, Canada.
- [70] G. Sapiro, "Color and illuminant voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1210–1215, November 1999.
- [71] P. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, June 2008, Anchorage, AK.
- [72] S. Beigpour, C. Riess, J. van de Weijer, and E. Angelopoulou, "Multi-illuminant estimation with conditional random fields," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 83–96, January 2014.
- [73] M. Sajjaa and G. Fischer, "Automatic white balance: WhitebalPR using the dichromatic reflection model," *Proceedings of the IS&T/SPIE Conference on Digital Photography*, vol. 7250, pp. 72 500D–72 500D–12, January 2009, San Jose, CA.

- [74] F. Zaraga and G. Langfelder, "White balance by tunable spectral responsivities," *Journal of the Optical Society of America*, vol. 27, no. 1, pp. 31–39, January 2010.
- [75] A. Ilie and G. Welch, "Ensuring color consistency across multiple cameras," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1268–1275, October 2005, Beijing, China.
- [76] E. A. Johnson, "Touch display - A novel input/output device for computers," *Electronics Letters*, vol. 1, no. 8, p. 219, 1965.
- [77] N. Matsushita and J. Rekimoto, "HoloWall: Designing a finger, hand, body, and object sensitive wall," *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, pp. 209–210, October 1997, Banff, Canada.
- [78] S. Izadi, H. Brignull, T. Rodden, Y. Rogers, and M. Underwood, "Dynamo: A public interactive surface supporting the cooperative sharing and exchange of media," *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, pp. 159–168, November 2003, Vancouver, Canada.
- [79] J. Rekimoto, "SmartSkin: An infrastructure for freehand manipulation on interactive surfaces," *Proceedings of the 20th Annual SIGCHI Conference on Human Factors in Computing Systems*, pp. 113–120, April 2002, Minneapolis, MN.
- [80] A. Pirhonen, S. Brewster, and C. Holguin, "Gestural and audio metaphors as a means of control for mobile devices," *Proceedings of the 20th Annual SIGCHI Conference on Human Factors in Computing Systems*, pp. 291–298, April 2002, Minneapolis, MN.
- [81] E. Hoggan, S. A. Brewster, and J. Johnston, "Investigating the effectiveness of tactile feedback for mobile touchscreens," *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pp. 1573–1582, April 2008, Florence, Italy.
- [82] K. S. Deoras, M. R. Wolfson, R. L. Searls, S. R. Hilfer, J. B. Sheffield, and T. H. Shaffer, "Use of a touch sensitive screen and computer assisted image analysis for quantitation of developmental changes in pulmonary structure," *Pediatr Pulmonol*, vol. 9, no. 2, pp. 109–18, 1990.
- [83] J. Dai and C.-K. Chung, "Touchscreen everywhere: On transferring a normal planar surface to a touch-sensitive display," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–14, November 2013.
- [84] J. Krauskopf and G. Karl, "Color discrimination and adaptation," *Vision Research*, vol. 32, no. 11, pp. 2165–2175, January 1992.
- [85] K.-M. Cho, J.-H. Jang, and K.-S. Hong, "Adaptive skin-color filter," *Pattern Recognition*, vol. 34, no. 5, pp. 1067–1073, May 2001.
- [86] R. Jusoh, N. Hamzah, M. Marhaban, and N. Alias, "Skin detection based on thresholding in RGB and hue component," *Proceedings of the 2010 IEEE Symposium on Industrial Electronics Applications*, pp. 515–517, October 2010, Penang, Malaysia.



- [87] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.
- [88] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.
- [89] A. Rosenfeld and A. Kak, *Digital Picture Processing Vol. 2*. Academic Press, New York, 1982.
- [90] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.
- [91] T. Q. Chen, Y. L. Murphey, R. Karlsen, and G. Gerhart, "Color image segmentation in color and spatial domain," *Proceedings of the 16th International Conference on Developments in Applied Artificial Intelligence*, pp. 72–82, June 2003, Laughborough, United Kingdom.
- [92] W. Skarbek and A. Koschan, "Colour image segmentation - A survey," Technical University of Berlin, Department of Computer Science, Tech. Rep., 1994.
- [93] H. Cheng, X. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.
- [94] L. Lucchese and S. Mitra, "Color image segmentation: A state-of-the-art survey," *Proceedings of the Indian National Science Academy*, vol. 67 A, pp. 207–221, March 2001, New Delhi, India.
- [95] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 925–936, July 2005.
- [96] Y. He, N. Khanna, C. Boushey, and E. Delp, "Image segmentation for image-based dietary assessment: A comparative study," *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, July 2013, Iasi, Romania.
- [97] S. R. Vantaram and E. Saber, "Survey of contemporary trends in color image segmentation," *Journal of Electronic Imaging*, vol. 21, no. 4, pp. 040 901–1–040 901–28, October 2012.
- [98] R. Tan and K. Ikeuchi, "Separating reflection components of textured surfaces using a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 178–193, February 2005.
- [99] G. Healey, "Segmenting images using normalized color," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 64–73, January 1992.
- [100] B. A. Maxwell and S. A. Shafer, "Physics-based segmentation of complex objects using multiple hypotheses of image formation," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 269–295, November 1997.
- [101] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, pp. 604–610, October 2005, Montbonnot, France.

- [102] Y. Tarabalka, J. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, August 2009.
- [103] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 313–320, June 2003, Urbana, IL.
- [104] A. Fred and A. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, June 2005.
- [105] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 917–930, December 2010.
- [106] S. Phung, A. Bouzerdoum, and S. Chai, D., "Skin segmentation using color pixel classification: analysis and comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148–154, January 2005.
- [107] C.-I. Chang, Y. Du, J. Wang, S.-M. Guo, and P. Thouin, "Survey and comparative analysis of entropy and relative entropy thresholding techniques," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 153, no. 6, pp. 837–850, December 2006.
- [108] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, July 2003.
- [109] V. Grau, A. U. J. Mewes, M. Alcaniz, R. Kikinis, and S. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 447–458, April 2004.
- [110] A. Round, A. Duller, and P. Fish, "Colour segmentation for lesion classification," *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2, pp. 582–585, November 1997, Chicago, IL.
- [111] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, August 2001.
- [112] M. Plissiti, D. Fotiadis, L. Michalis, and G. Bozios, "An automated method for lumen and media-adventitia border detection in a sequence of ivus frames," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 2, pp. 131–141, June 2004.
- [113] N. Funakubo, "Feature extraction of color texture using neural networks for region segmentation," *Proceedings of the 20th Annual Conference of IEEE Industrial Electronics*, vol. 2, pp. 852–856, September 1994, Bologna, Italy.

- [114] T. Carron and P. Lambert, "Color edge detector using jointly hue, saturation and intensity," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 977–981, November 1994, Austin, TX.
- [115] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001.
- [116] Y. He, N. Khanna, C. J. Boushey, and E. Delp, "Snakes assisted food image segmentation," *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSp)*, pp. 181–185, September 2012, Banff, Canada.
- [117] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2013, San Jose, CA.
- [118] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal Of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [119] I. Milevskiy and J.-Y. Ha, "A fast algorithm for korean text extraction and segmentation from subway signboard images utilizing smartphone sensors." *Journal of Computing Science and Engineering*, vol. 5, no. 3, pp. 161–166, September 2011.
- [120] D. H. Rao and P. Panduranga, "A survey on image enhancement techniques: Classical spatial filter, neural network, cellular neural network, and fuzzy filter," *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 2821–2826, December 2006, Mumbai, India.
- [121] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.
- [122] M. Alam, J. Bogner, R. Hardie, and B. Yasuda, "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 915–923, October 2000.
- [123] M. Chabert and B. Lacaze, "Non uniform sampling for remote sensing images," *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4718–4721, July 2012, Munich, Germany.
- [124] A. Patti and Y. Altunbasak, "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 179–186, January 2001.
- [125] J. J. Zou and H. Yan, "A deblocking method for BDCT compressed images based on adaptive projections," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 430–435, March 2005.
- [126] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 387–395, March 1999.

- [127] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007.
- [128] H. Kong, J.-Y. Audibert, and J. Ponce, “General road detection from a single image,” *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, August 2010.
- [129] D. Rajan and S. Chaudhuri, “Simultaneous estimation of super-resolved scene and depth map from low resolution defocused observations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1102–1117, September 2003.
- [130] H. Aly and E. Dubois, “Image up-sampling using total-variation regularization with a new observation model,” *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, October 2005.
- [131] F. Salem and A. Yagle, “Non-parametric super-resolution using a bi-sensor camera,” *IEEE Transactions on Multimedia*, vol. 15, no. 1, pp. 27–40, January 2013.
- [132] N. Nguyen, P. Milanfar, and G. Golub, “Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement,” *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1299–1308, September 2001.
- [133] F. Sroubek, G. Cristobal, and J. Flusser, “A unified approach to superresolution and multichannel blind deconvolution,” *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2322–2332, September 2007.
- [134] E. Faramarzi, D. Rajan, and M. Christensen, “Unified blind method for multi-image super-resolution and single/multi-image blur deconvolution,” *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2101–2114, June 2013.
- [135] C.-T. Lin, K.-W. Fan, H.-C. Pu, S.-M. Lu, and S.-F. Liang, “An HVS-directed neural-network-based image resolution enhancement scheme for image resizing,” *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 605–615, August 2007.
- [136] D. Marin, A. Aquino, M. Gegundez-Arias, and J. Bravo, “A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 1, pp. 146–158, January 2011.
- [137] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. Kerre, W. Philips, and I. Lemahieu, “Noise reduction by fuzzy image filtering,” *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 429–436, August 2003.
- [138] S. Schulte, M. Nachtegael, V. De Witte, D. Van der Weken, and E. Kerre, “A fuzzy impulse noise detection and reduction method,” *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1153–1162, May 2006.
- [139] M. Selvi and A. George, “FBFET: Fuzzy based fingerprint enhancement technique based on adaptive thresholding,” *Proceedings of the International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–5, July 2013, Tiruchengode, India.

- [140] T. Shih, L. Lin, and W. Lee, "Detection and removal of long scratch lines in aged films," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 477–480, July 2006, Toronto, Canada.
- [141] Y.-T. Kao, T. Shih, H.-Y. Zhong, and L.-K. Dai, "Scratch line removal on aged films," *Proceedings of the 9th IEEE International Symposium on Multimedia*, pp. 147–151, December 2007, Taichung, Taiwan.
- [142] Z. Qingyue and D. Youdong, "Scratch line detection and restoration based on canny operator," *Proceedings of the Asia-Pacific Conference on Information Processing (APCIP)*, vol. 2, pp. 148–151, July 2009, Shenzhen, Hong Kong.
- [143] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 417–424, 2000, New Orleans, LA.
- [144] Q. Miao, P. Xu, T. Liu, Y. Yang, J. Zhang, and W. Li, "Linear feature separation from topographic maps using energy density and the shear transform," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1548–1558, April 2013.
- [145] N. I. N. Ismail and A. M. S. Noor, *A Novel Technique for Contour Reconstruction to DEM*, ser. Research Monograph. Pusat Pengurusan Penyelidikan, Universiti Teknologi Malaysia, 2009.
- [146] E. Hancer and R. Samet, "Advanced contour reconnection in scanned topographic maps," *Proceedings of the International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–5, October 2011, Baku, Azerbaijan.
- [147] R. Samet and E. Hancer, "A new approach to the reconstruction of contour lines extracted from topographic maps," *Journal of Visual Communication and Image Representation*, vol. 23, no. 4, pp. 642–647, May 2012.
- [148] A. N. G. L. Filho and C. A. B. Mello, "A novel method for reconstructing degraded digits," *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 733–738, October 2012, Seoul, South Korea.
- [149] A. N. G. L. Filho and C. A. B. Mello, "Degraded digit restoration based on physical forces," *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 195–199, August 2013, Washington, DC.
- [150] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.
- [151] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002.
- [152] A. Natsev, R. Rastogi, and K. Shim, "WALRUS: a similarity retrieval algorithm for image databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 3, pp. 301–316, March 2004.

- [153] P. Hiremath and J. Pujari, "Content based image retrieval using color, texture and shape features," *Proceedings of the International Conference on Advanced Computing and Communications*, pp. 780–784, December 2007, Guwahati, India.
- [154] J. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 235–240, February 2008.
- [155] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: An experimental comparison," *Journal of Information Retrieval*, vol. 11, no. 2, pp. 77–107, April 2008.
- [156] Y. Cai and G. Baciu, "Detecting, grouping, and structure inference for invariant repetitive patterns in images," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2343–2355, June 2013.
- [157] O. Penatti and R. da Silva Torres, "Color descriptors for web image retrieval: A comparative study," *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pp. 163–170, October 2008, Campo Grande, Brazil.
- [158] O. A. B. Penatti, E. Valle, and R. d. S. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *Journal of Visual Communication and Image Representation*, vol. 23, no. 2, pp. 359–380, February 2012.
- [159] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.
- [160] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," *Proceedings of the 4th ACM international conference on Multimedia*, pp. 65–73, 1997, Boston, MA.
- [161] G. Paschos, I. Radev, and N. Prabakar, "Image content-based retrieval using chromaticity moments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1069–1072, September 2003.
- [162] A. Utenpattanant, O. Chitsobhuk, and A. Khawne, "Color descriptor for image retrieval in wavelet domain," *Proceedings of the 8th International Conference on Advanced Communication Technology (ICACT)*, vol. 1, pp. 821–824, February 2006, Phoenix Park, Ireland.
- [163] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.
- [164] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, June 2001.
- [165] K.-L. Lee and L.-H. Chen, "An efficient computation method for the texture browsing descriptor of MPEG-7," *Image and Vision Computing*, vol. 23, no. 5, pp. 479–489, May 2005.

- [166] V. Risojević, S. Momić, and Z. Babić, “Gabor descriptors for aerial image classification,” *Proceedings of the 10th International Conference on Adaptive and Natural Computing Algorithms - Volume Part II*, pp. 51–60, 2011, Ljubljana, Slovenia.
- [167] H. G. Feichtinger and T. Strohmer, *Gabor Analysis and Algorithms: Theory and Applications*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 1998.
- [168] D. Zhang and G. Lu, “Evaluation of MPEG-7 shape descriptors against other shape descriptors,” *Multimedia System*, vol. 9, pp. 15–30, July 2003.
- [169] C. T. Zahn and R. Z. Roskies, “Fourier Descriptors for plane closed curves,” *IEEE Transactions on Computers*, vol. 21, no. 3, pp. 269–281, March 1972.
- [170] E. Persoon and K. S. Fu, “Shape discrimination using Fourier Descriptors,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 3, pp. 170–179, March 1977.
- [171] Y. Zhao and S. Belkasim, “Multiresolution Fourier Descriptors for multiresolution shape analysis,” *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 692–695, October 2012.
- [172] F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space,” *Proceedings of the International Workshop on Image Databases and Multimedia Search*, pp. 35–42, 1996, Amsterdam, Netherlands.
- [173] A. Dyana and S. Das, “MST-CSS (Multi-Spectro-Temporal Curvature Scale Space), a novel spatio-temporal representation for content-based video retrieval,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 8, pp. 1080–1094, August 2010.
- [174] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, February 1962.
- [175] D. Xu and H. Li, “Geometric moment invariants,” *Pattern Recognition*, vol. 41, no. 1, pp. 240–249, 2008.
- [176] M. R. Teague, “Image analysis via the general theory of moments,” *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, August 1980.
- [177] S. Li, M.-C. Lee, and C.-M. Pun, “Complex zernike moments features for shape-based image retrieval,” *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 39, no. 1, pp. 227–237, January 2009.
- [178] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, October 2005.
- [179] N. Pinto, Y. Barhomi, D. Cox, and J. DiCarlo, “Comparing state-of-the-art visual features on invariant object recognition tasks,” *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 463–470, January 2011, Kona, HI.

- [180] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV)*, pp. 1–8, October 2007, Rio de Janeiro, Brazil.
- [181] Z. Chen, F. Yang, A. Lindner, G. Barrenetxea, and M. Vetterli, "How is the weather: Automatic inference from images," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1853–1856, September 2012, Orlando, FL.
- [182] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, July 2002.
- [183] E. N. Mortensen, H. Deng, and L. Shapiro, "A SIFT descriptor with global context," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 184–190, 2005, San Diego, CA.
- [184] S. Belongie, J. Malik, and J. Puzicha, "Shape context: A new descriptor for shape matching and object recognition," *Proceedings of the Neural Information Processing Systems Conference*, pp. 831–837, 2000, Denver, CO.
- [185] S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509–522, April 2002.
- [186] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 5:1–5:60, May 2008, New York, NY.
- [187] N. Singhai and S. K. Shandilya, "A survey on: content based image retrieval systems," *International Journal of Computer Applications*, vol. 2, no. 4, pp. 22–26, 2010.
- [188] M. Jain and S. Singh, "A survey on: Content based image retrieval systems using clustering techniques for large data sets," *International Journal of Managing Information Technology*, vol. 3, no. 4, pp. 23–29, 2011.
- [189] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.
- [190] F. Zhu, M. Bosch, N. Khanna, C. Boushey, and E. Delp, "Multilevel segmentation for food classification in dietary assessment," *Proceedings of 7th International Symposium on Image and Signal Processing and Analysis*, pp. 337–342, September 2008, Dubrovnik, Croatia.
- [191] D. Ilea and P. Whelan, "CTex - an adaptive unsupervised segmentation algorithm based on color-texture coherence," *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1926–1939, October 2008.
- [192] J. Li and J. W., "Studying digital imagery of ancient paintings by mixtures of stochastic models," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 340–353, March 2004.



- [193] H. Muller, T. Pun, and D. Squire, "Learning from user behavior in image retrieval: Application of market basket analysis," *International Journal of Computer Vision*, vol. 56, pp. 65–77, January 2004.
- [194] J. He, H. Tong, M. Li, H.-J. Zhang, and C. Zhang, "Mean version space: a new active learning method for content-based image retrieval," *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 15–22, October 2004, New York, NY.
- [195] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "An efficient and effective region-based image retrieval framework," *IEEE Transactions on Image Processing*, vol. 13, no. 5, pp. 699–709, May 2004.
- [196] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168, June 2006, Washington, DC.
- [197] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," *Proceedings of the ACM international conference on Multimedia*, pp. 107–118, October 2001, Ottawa, Canada.
- [198] Z. Su, H. Zhang, S. Li, and S. Ma, "Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 924–937, August 2003.
- [199] Y. Wu, Q. Tian, and T. Huang, "Discriminant-EM algorithm with application to image retrieval," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 222–227, June 2000, Hilton Head Island, NC.
- [200] X. He, O. King, W.-Y. Ma, M. Li, and H.-J. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 39–48, January 2003.
- [201] F. Jing, M. Li, H.-J. Zhang, and B. Zhang, "Relevance feedback in region-based image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 672–681, May 2004.
- [202] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, pp. 536–544, April 2003.
- [203] A. Jaimes, K. Omura, T. Nagamine, and K. Hirata, "Memory cues for meeting video retrieval," *Proceedings of the the 1st ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, pp. 74–85, October 2004, New York, NY.
- [204] C. Yang, J. Yang, and D. Feng, "Magazine image retrieval with camera-phone," *Lecture Notes in Electrical Engineering, Recent Progress in Data Engineering and Internet Technology*, vol. 156, pp. 55–60, 2013.
- [205] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1624–1636, November 2011.

- [206] J. M. Saavedra and B. Bustos, "Sketch-based image retrieval using keyshapes," *Multimedia Tools and Applications*, pp. 1–30, September 2013.
- [207] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 121–132, February 1997.
- [208] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 1, pp. 28–41, January 2005.
- [209] D.-C. Tseng, Y.-F. Li, and C.-T. Tung, "Circular histogram thresholding for color image segmentation," *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, vol. 2, pp. 673–676, August 1995, Montreal, Canada.
- [210] D.-C. Tseng and C.-H. Chang, "Color segmentation using perceptual attributes," *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, vol. 3, pp. 228–231, September 1992, La Haye, Holland.
- [211] J. Brand and J. Mason, "Skin probability map and its use in face detection," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 1034–1037, October 2001, Thessaloniki, Greece.
- [212] Z. Xue, D. Shen, and S. Wong, "Tissue probability map constrained CLASSIC for increased accuracy and robustness in serial image segmentation," *Proceedings of the SPIE Symposium on Medical Imaging*, vol. 7258, pp. 725 904–1–9, February 2009, Lake Buena Vista, FL.
- [213] J. Jiang, Y. Zhao, and S.-G. Wang, "Color correction of smartphone photos with prior knowledge," *Proceedings of the IS&T/SPIE Electronic Imaging on Imaging and Printing in a Web 2.0 World III*, vol. 8302, pp. 83 020H:1–83 020H:6, January 2012, Burlingame, CA.
- [214] R. M. Boynton, *Human Color Vision*. Holt Rinehart and Winston, 1979.
- [215] E. Schubert, *Light-emitting Diodes*. Cambridge University Press, 2003, ch. Human eye sensitivity and photometric quantities, pp. 275–291.
- [216] J. Chae, I. Woo, S. Kim, R. Maciejewski, F. Zhu, E. Delp, C. Boushey, and D. Ebert, "Volume estimation using food specific shape templates in mobile image-based dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, vol. 7873, pp. 78 730K–1–78 730K–8, January 2011, San Francisco, CA.
- [217] M. Ruffi, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkboards on blurred and distorted images," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 3121–3126, September 2008, Nice, France.
- [218] J. J. McCann, "Color spaces for color-gamut mapping," *Journal of Electronic Imaging*, vol. 8, no. 4, pp. 354–364, October 1999.
- [219] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae. Second Edition*, ser. Wiley Series in Pure and Applied Optics. Wiley, John, and Sons, New York, N.Y., 1982.

- [220] F. López, J. Valiente, R. Baldrich, and M. Vanrell, “Fast surface grading using color statistics in the CIELab space,” *Proceedings of the Second Iberian Conference on Pattern Recognition and Image Analysis (IBPRIA)*, pp. 666–673, June 2005, Storil, Portugal.
- [221] *Recommendation ITU-R BT.709, Parameter values for the HDTV standards for production and international programme exchange*, International Telecommunications Union, Geneva, Switzerland, 1990.
- [222] G. Strang, *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.
- [223] C. Poynton, *Digital Video and HDTV Algorithms and Interfaces*, 1st ed. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2003.
- [224] L. Lam, S. W. Lee, and C. Y. Suen, “Thinning methodologies - A comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, September 1992.
- [225] Z. Guo and R. W. Hall, “Parallel thinning with two-subiteration algorithms,” *Communications of the ACM*, vol. 32, no. 3, pp. 359–373, March 1989.
- [226] P. V. C. Hough, “Machine analysis of bubble chamber pictures,” *Proceedings of the International Conference on High Energy Accelerators and Instrumentation*, pp. 554–558, September 1959, Geneva, Switzerland.
- [227] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, January 1972.
- [228] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [229] E. Hancer and R. Samet, “Advanced contour reconnection in scanned topographic maps,” *Proceedings of the 5th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–5, October 2011, Baku, Azerbaijan.
- [230] W. Wang, H. Pottmann, and Y. Liu, “Fitting B-spline curves to point clouds by curvature-based squared distance minimization,” *ACM Transactions on Graphics*, vol. 25, no. 2, pp. 214–238, April 2006.
- [231] D. G. Lowe, “Object recognition from local scale-invariant features,” *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 1150–1157, September 1999, Kerkyra, Greece.
- [232] P. Perona, “Deformable kernels for early vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 488–499, May 1995.
- [233] E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide baseline stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, May 2010.

- [234] T.-S. Chen, T.-H. Tsai, Y.-T. Chen, C.-C. Lin, R.-C. Chen, S.-Y. Li, and H.-Y. Chen, "A combined k-means and hierarchical clustering method for improving the clustering efficiency of microarray," *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 405–408, December 2005, Hong Kong, China.
- [235] T. Su and J. Dy, "A deterministic method for initializing k-means clustering," *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 784–786, November 2004, Boca Raton, FL.
- [236] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.
- [237] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- [238] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" *Proceedings of the 7th International Conference on Database Theory*, pp. 217–235, 1999, London, United Kingdom.
- [239] S. Arya, D. M. Mount, and O. Narayan, "Accounting for boundary effects in nearest-neighbor searching," *Discrete & Computational Geometry*, vol. 16, no. 2, pp. 155–176, 1996.
- [240] S. Berchtold, C. Böhm, D. A. Keim, and H.-P. Kriegel, "A cost model for nearest neighbor search in high-dimensional data space," *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 78–86, 1997, Tucson, AR.
- [241] D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-nearest neighbour classification by using alternative voting rules," *Analytica Chimica Acta*, vol. 136, no. 0, pp. 15–27, 1982.
- [242] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009, Miami, FL.
- [243] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual weighting for vocabulary tree based image retrieval," *Proceedings of the 2011 International Conference on Computer Vision*, pp. 209–216, 2011, Washington, DC.
- [244] B. Momjian, *PostgreSQL: Introduction and Concepts*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 2001.
- [245] "Android Developers. Platform Versions as of February 4, 2014." [developer.android.com/about/dashboards](http://developer.android.com/about/dashboards)
- [246] E. Lafortune, "ProGuard: Optimizer and obfuscator in the Android SDK," 2006. [proguard.sourceforge.net](http://proguard.sourceforge.net)
- [247] J. D. Touch, "Performance analysis of MD5," *ACM SIGCOMM Computer Communication Review*, pp. 77–86, October 1995.

- [248] C. Grigorescu and N. Petkov, "Distance sets for shape filters and shape recognition," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1274–1286, October 2003.
- [249] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, pp. 453–464, March 1999.
- [250] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, November 1998.
- [251] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 70–75, September 2004, Stockholm, Sweden.
- [252] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 264–278, June 2007.
- [253] R. Malik, J. Khurshid, and S. Ahmad, "Road sign detection and recognition using colour segmentation, shape analysis and template matching," *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3556–3560, August 2007, Hong Kong, China.
- [254] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1498–1506, December 2012.
- [255] O. R. Mitchell and T. A. Grogan, "Global and partial shape discrimination for computer vision," *Optical Engineering*, vol. 23, no. 5, pp. 484–491, October 1984.
- [256] R. C. Gonzalez, *Digital Image Processing*, 2nd ed. New Jersey: Prentice Hall, 2000.
- [257] F. Larsson, M. Felsberg, and P.-E. Forssen, "Correlating Fourier Descriptors of local patches for road sign recognition," *IET Computer Vision*, vol. 5, pp. 244–254, January 2011.
- [258] P. van Otterloo, *A Contour-Oriented Approach to Shape Analysis*, 2nd ed. Englewood Cliffs, New Jersey: Prentice-Hall International, 2000.
- [259] R. Chellappa and R. Bagdazian, "Fourier coding of image boundaries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 102–105, January 1984.
- [260] C. Singh and P. Sharma, "Performance analysis of various local and global shape descriptors for image retrieval," *Multimedia Systems*, vol. 19, no. 4, pp. 339–357, July 2013.
- [261] I. Kunttu, L. Lepisto, J. Rauhamaa, and A. Visa, "Multiscale Fourier Descriptor for shape-based image retrieval," *Proceedings of the IEEE Conference on Pattern Recognition*, pp. 765–768, August 2004, Cambridge, United Kingdom.

- [262] N. M. Tahir, A. Hussain, and M. M. Mustafa, "Fourier Descriptor for pedestrian shape recognition using support vector machine," *Proceedings of the IEEE International Symposium on Signal Processing and Information*, pp. 636–641, December 2007, Cairo, Egypt.
- [263] O. van Kaick, G. Hamarneh, H. Zhang, and P. Wighton, "Contour correspondence via ant colony optimization," *Proceedings of the Pacific Conference on Computer Graphics and Applications*, pp. 271–280, October 2007, Maui, HI.
- [264] M. Jie, Z. Zhiwei, T. HongMei, and Z. QuanMing, "Fast Fourier Descriptor method of the shape feature in low resolution images," *Proceedings of the IEEE Conference Wireless Communications Networking and Mobile Computing*, pp. 1–4, September 2010, Chengdu, China.
- [265] A. Broggi, P. Cerri, P. Medici, P. Porta, and G. Ghisio, "Real time road signs recognition," *IEEE Intelligent Vehicles Symposium*, pp. 981–986, June 2007, Istanbul, Turkey.
- [266] L. Song and Z. Liu, "Color-based traffic sign detection," *International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, pp. 353–357, June 2012, Chengdu, China.
- [267] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, January 2013.
- [268] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 545–552, December 2006, Vancouver, Canada.
- [269] X. Hou and L. Zhang, "Dynamic visual attention: Searching for coding length increments," *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 681–688, December 2008, Vancouver, Canada.
- [270] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 409–416, June 2011, Colorado Springs, CO.
- [271] C. Kim and P. Milanfar, "Visual saliency in noisy images," *Journal of Vision*, vol. 13, no. 4, pp. 1–14, March 2013.
- [272] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, January 2012.
- [273] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, April 2013.
- [274] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Pappadimitris, "Road sign detection in images: A case study," *Proceedings of the International Conference on Pattern Recognition*, pp. 484–488, August 2010, Istanbul, Turkey.

- [275] A. Mogelmose, M. Trivedi, and T. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, December 2012.
- [276] K. L. Bouman, G. Abdollahian, M. Boutin, and E. J. Delp, "A low complexity sign detection and text localization method for mobile applications," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 922–934, October 2011.
- [277] D. Pao, H. Li, and R. Jayakumar, "Shapes recognition using the straight line Hough transform: theory and generalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1076–1089, November 1992.
- [278] S. Houben, "A single target voting scheme for traffic sign detection," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 124–129, June 2011, Baden-Baden, Germany.
- [279] H. Fleyeh and P. Zhao, "A contour-based separation of vertically attached traffic signs," *Proceedings of the Annual Conference of Industrial Electronics*, pp. 1811–1816, November 2008, Orlando, FL.
- [280] L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, Y.-J. Tseng, K.-C. Fan, and C.-C. Lee, "Road sign detection using eigen colour," *IET Computer Vision*, no. 3, pp. 164–177, September 2008.
- [281] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 959–973, August 2003.
- [282] N. Barnes, A. Zelinsky, and L. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 322–332, June 2008.
- [283] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
- [284] C. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff, "Real-time recognition of U.S. speed signs," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 518–523, June 2008, Eindhoven, Netherlands.
- [285] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary AdaBoost detection and Forest-ECOC classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 113–126, March 2009.
- [286] A. Rostampour and P. Madhvapathy, "Shape recognition using simple measures of projections," *Proceedings of the Annual International Phoenix Conference on Computers and Communications*, pp. 474–479, March 1988, Scottsdale, AR.
- [287] P. Gil-Jimenez, S. Lafuente-Arroyo, H. Gomez-Moreno, F. Lopez-Ferreras, and S. Maldonado-Bascon, "Traffic sign shape classification evaluation. part II. FFT applied to the signature of blobs," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 607–612, June 2005, Las Vegas, NV.

- [288] A. W. Haddad, S. Huang, M. Boutin, and E. J. Delp, "Detection of symmetric shapes on a mobile device with applications to automatic sign interpretation," *Proceedings of the IS&T/SPIE Electronic Imaging on Multimedia on Mobile Devices*, vol. 8304, January 2012, San Francisco, CA.
- [289] "Ocrad - GNU Project - Free Software Foundation (FSF)." [www.gnu.org/software/ocrad](http://www.gnu.org/software/ocrad)
- [290] A. Parra, A. W. Haddad, M. Boutin, and E. Delp, "A method for translating printed documents using a hand-held device," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2011, Barcelona, Spain.
- [291] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 194–201, January 2012.
- [292] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, April 1985.
- [293] J. Sklansky, "Finding the convex hull of a simple polygon," *Pattern Recognition Letters*, vol. 1, no. 2, pp. 79–83, December 1982.
- [294] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, January 1979.
- [295] C. Correa, C. Valero, and P. Barreiro, "Row crop's identification through Hough transform using images segmented by robust fuzzy possibilistic c-means," *Proceedings of the Spanish Association for Artificial Intelligence*, November 2011, La Laguna, Spain.
- [296] H.-D. Cheng and Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," *IEEE Transactions on Image Processing*, vol. 9, no. 12, pp. 2071–2082, 2000.
- [297] P. Soille, *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, 1999.
- [298] G. Anelli, A. Broggi, and G. Destri, "Decomposition of arbitrarily-shaped morphological structuring elements using genetic algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 217–224, 1998.
- [299] H. Park and R. Chin, "Decomposition of arbitrarily-shaped morphological structuring elements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 2–15, 1995.
- [300] R. F. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ: Prentice-Hall, Inc., 2003.
- [301] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1123–1129, 2000.



- [302] F. Essannouni and D. Aboutajdine, “Fast frequency template matching using higher order statistics,” *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 826–830, 2010.
- [303] I. Bartolini, P. Ciaccia, and M. Patella, “WARP: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 142–147, 2005.
- [304] R. P. Brent, “Fast multiple-precision evaluation of elementary functions,” *Journal of the ACM*, vol. 23, pp. 242–251, April 1976.
- [305] D. Pascale, “RGB coordinates of the Macbeth ColorChecker,” *The BabelColor Company*, pp. 1–15, June 2006, Montreal, Canada.
- [306] S. V. Stehman, “Selecting and interpreting measures of thematic classification accuracy,” *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.
- [307] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [308] E. M. Voorhees, “Variations in relevance judgments and the measurement of retrieval effectiveness,” *Information Processing & Management*, vol. 36, no. 5, pp. 697–716, 2000.
- [309] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, September 2010.
- [310] J. Huang, H. Liu, J. Shen, and S. Yan, “Towards efficient sparse coding for scalable image annotation,” *Proceedings of the 21st ACM International Conference on Multimedia*, pp. 947–956, October 2013, Barcelona, Spain.
- [311] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with Stroke Width Transform,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2963–2970, June 2010, San Francisco, CA.
- [312] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, November 1998.
- [313] J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency,” *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 545–552, December 2006, Vancouver, Canada.
- [314] B. Zhao, A. Parra, and E. J. Delp, “Mobile-based hazmat sign detection system,” *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 735–738, December 2013, Austin, TX.
- [315] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, “CHoG: Compressed histogram of gradients: A low bit-rate feature descriptor,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2504–2511, June 2009, Miami, FL.

- [316] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, February 2013.
- [317] K. Mikolajczyk, B. Leibe, and B. Schiele, “Multiple object class detection with a generative model,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 26–36, June 2006, New York, NY.
- [318] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, “Layered object detection for multi-class segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3113–3120, June 2010, San Francisco, CA.
- [319] M. Asmare, V. Asirvadam, and L. Iznita, “Color space selection for color image enhancement applications,” *Proceedings of the International Conference on Signal Acquisition and Processing*, pp. 208–212, April 2009, Kuala Lumpur, Malaysia.
- [320] M. Tkalcic and J. Tasic, “Colour spaces: Perceptual, historical and applicational background,” *Proceedings of the IEEE Region 8 Eurocon 2003: Computer as a Tool*, vol. 1, pp. 304–308, September 2003, Ljubljana, Slovenia.
- [321] G. H. Joblove and D. Greenberg, “Color spaces for computer graphics,” *ACM SIGGRAPH Computer Graphics*, vol. 2, no. 3, pp. 20–25, August 1978.
- [322] A. R. Smith, “Color gamut transform pairs,” *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 12–19, 1978, New York, NY.
- [323] A. Hanbury, “A 3D-polar coordinate colour representation well adapted to image analysis,” *Proceedings of the 13th Scandinavian Conference on Image Analysis*, pp. 804–811, June-July 2003, Halmstad, Sweden.
- [324] J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1982.
- [325] M. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*. Springer, 2005.
- [326] *Recommendation ITU-R BT.601, Encoding Parameters of Digital Television for Studios*, International Telecommunications Union, Geneva, Switzerland, 1992.
- [327] J. D. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed. Addison-Wesley, 1990.

## APPENDICES

## A. RGB TO Y'CH COLOR SPACE CONVERSION

An image captured using our Android application is saved as 32-bit RGB JPEG file, where each pixel is a packed 32-bit integer containing the alpha, R, G and B color components from most to least significant bits respectively. Note that a JPEG image does not have an alpha channel and it is automatically set to zero by the Android bitmap Application Programming Interface (API). From these packed RGB bits we create a three-dimensional array to store the R, G and B components in their unpacked bit representations.

The RGB color space is psychologically non-intuitive because humans have problems with the visualization of a color defined in RGB [319]. The attributes of hue and saturation are the most natural way for humans to perceive colors [320]. The separation of the luma component from the chrominance information is advantageous in image processing. Therefore, we chose to transform the pixels in the image from the RGB color space to our new HSL-based color space, which we call the Y'CH color space, where we carefully define the three dimensions as luma, chroma and hue. We choose chroma over saturation because it better represents human perception of the variation in color purity with respect to luma. In the literature, saturation is defined as relative chroma [321, 322], and the difference must be taken into consideration. For example, the HSL color space is symmetrical with respect to luma, taking the shape of a cylinder. When using chroma the cylinder gets narrower as we move from the center of the neutral axis, forming a shape similar to a bicone [321]. Note that Figures 3.8 and 3.10 illustrate the Y'CH color space solid representation as a bicone for simplicity. However, its true shape is shown in Figure A.3, where not all the primaries lie in the same plane.

We can convert from RGB to our Y'CH in many ways. In this section we describe two approaches. The first one uses just arithmetic operations, while the second also

uses trigonometric operations. We conclude in Section 5 that the first approach is asymptotically faster and hence it is the method that we implemented in our Android application described in Section 3.8.3.

Our first approach for transforming from RGB to Y'CH, which we call the *arithmetic approach*, is illustrated in Figure A.1. First, we interpret the RGB cube as being tilted so that the black and white vertices are positioned at the top and the bottom of the neutral axis (vertical axis), respectively. Second, we project the tilted cube onto a plane perpendicular to the neutral axis, thus forming a hexagon. The chroma ( $C$ ) and hue ( $H$ ) components in our model are defined with respect to this hexagonal projection (Figure A.2). Chroma is the distance from the origin of the hexagon to its edge. We can define it as the difference between the largest and the smallest values of an RGB triplet [323] as shown in Equation A.1. Hue is the angle that represents the angular distance from the red edge of the projection (i.e., set to zero radians) to a particular RGB projection [324, 325], as shown in Equation A.2. Note that this theoretical hue, which we define as  $H'$ , is undefined for projections onto the neutral axis (i.e.,  $C = 0$ ). Also note that these definitions of chroma and hue correspond to a geometric warping of the hexagon into a circumference.

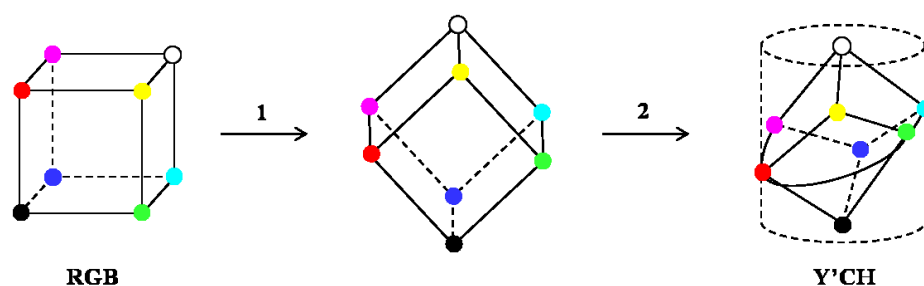


Fig. A.1.: Steps For Transforming from RGB to Y'CH Using The Arithmetic Approach.

$H'$  is then converted to degrees, which we define as  $H$ , by multiplying by 60. This multiplication accounts for  $\frac{360^\circ}{6}$ , which can be interpreted as the hexagonal analog of the unit circumference conversion from radians to degrees. That is, since  $2\pi$  is the perimeter of the unit circumference, we define the conversion as  $rad = \frac{360}{2\pi} \times deg$ .

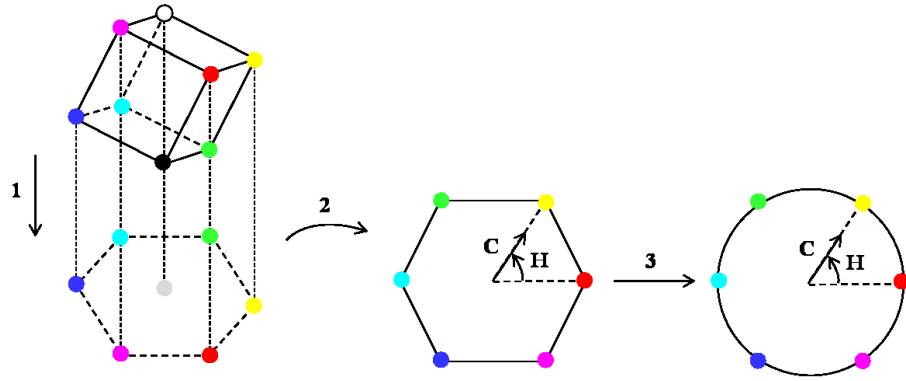


Fig. A.2.: Warping of the Hexagon Projection Into A Circumference in Our  $Y'CH$  Color Space.

Since 6 is the perimeter of the unit hexagon, we can define  $rad = \frac{360}{6} \times deg = 60 \times deg$ . Note that we define  $H = 0$  when  $C = 0$  in order to deal with the undefined hue angle for vector of magnitude zero.

Finally, our luma ( $Y'$ ) is the weighted average of gamma-corrected RGB color components. We define it using the Rec. 601 NTSC primaries [326], as shown in Equation A.3.

$$\begin{aligned}
 C &= \max(R, G, B) - \min(R, G, B) \\
 &= M - m.
 \end{aligned} \tag{A.1}$$

$$H' = \begin{cases} \frac{G-B}{C} & \text{if } M=R \\ \frac{B-R}{C} + 2 & \text{if } M=G \\ \frac{R-G}{C} + 4 & \text{if } M=B \\ \text{undefined} & \text{if } C=0 \end{cases} \tag{A.2}$$

$$Y = 0.299R + 0.587G + 0.114B. \tag{A.3}$$

Using these equations, our Y'CH color space is defined in  $0 \leq H < 360$  (or  $0 \leq H < 2\pi$  in radians),  $0 \leq C \leq 1$  and  $0 \leq Y \leq 1$ . The resulting representation is illustrated in step 3 of Figure A.1, where each colored dot represents a fully chromatic primary. Given our definitions of luma, chroma and hue, the color space representation does not have a symmetric shape. Figure A.3 illustrates a 3D view of the Y'CH solid. Figures A.4 to A.6 illustrate different cross-sections of constant hue, where the far left and far right corners represent fully chromatic colors. Note that the primaries do not lie in a common luma plane. Also note in Figure A.5 the effect of setting  $H = 0$  where  $C = 0$ , instead of being undefined. The neutral axis ( $C = 0$ ) does not contain luma values, since the cross-section is not located at  $H = 0$ . Figure A.4, however, since it is located at  $H = 0$ , we do not see any discontinuity.

Figure A.7 illustrates the bottom view of our Y'CH color space representation, where the hue of different primaries can be identified.

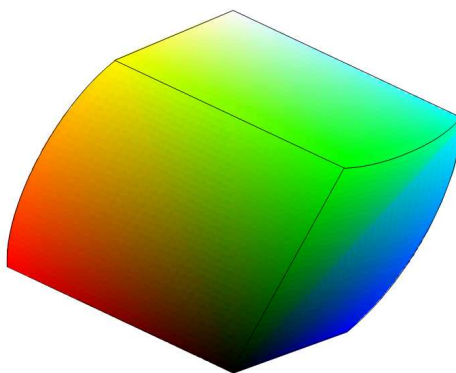


Fig. A.3.: 3D view of Our Y'CH Color Space (Using the Arithmetic Approach).

Our second approach for transforming from RGB to Y'CH, which we call the *trigonometric approach*, consists of defining the Y'CH color space using cylindrical coordinates, thus skipping the hexagon warping. First, we convert from RGB to Y'IQ using a linear transformation of the RGB cube [327], as shown in Equation A.4. With this conversion we directly obtain the Y'CH luma, which is defined again using the Rec. 601 NTSC primaries. Then, we can derive the hue and the chroma from a

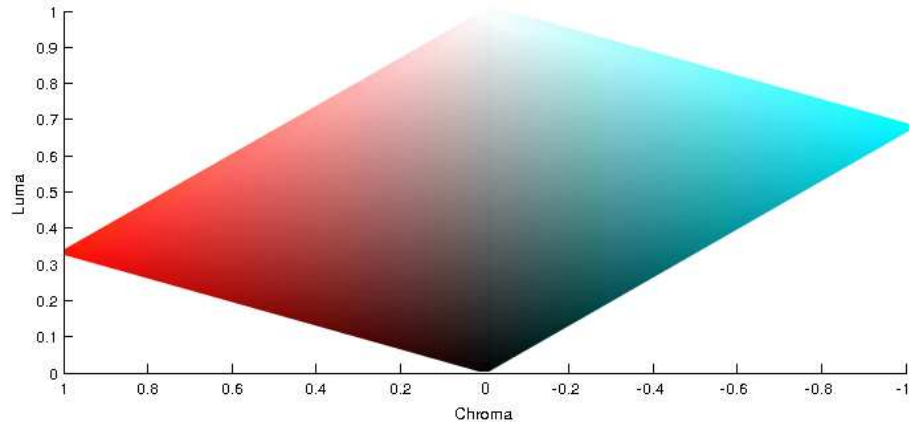


Fig. A.4.: Cross-Section of Constant Hue  $H = 0$  rad in Our  $Y'CH$  Color Space.

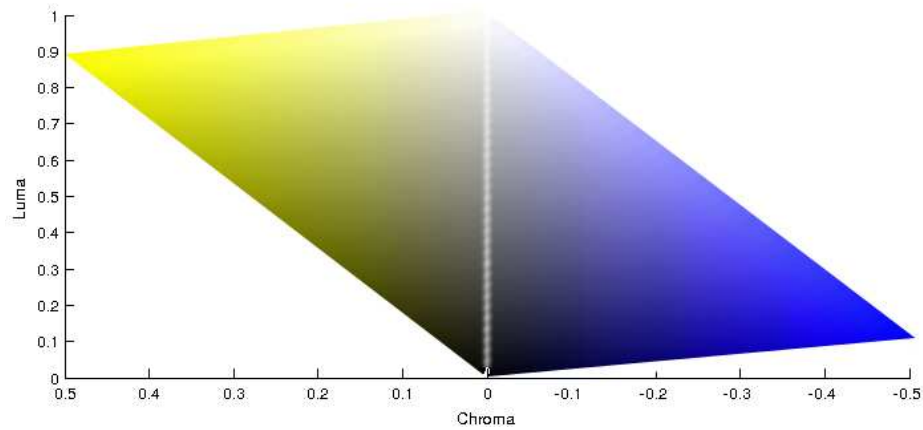


Fig. A.5.: Cross-Section of Constant Hue  $H = \frac{\pi}{3}$  rad in Our  $Y'CH$  Color Space.

cylindrical transformation of  $I$  and  $Q$  [323] as shown in Equation A.6. Note that the function  $atan2$  in Equation A.6 is the two-argument arctangent, defined in Equation A.7.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (\text{A.4})$$



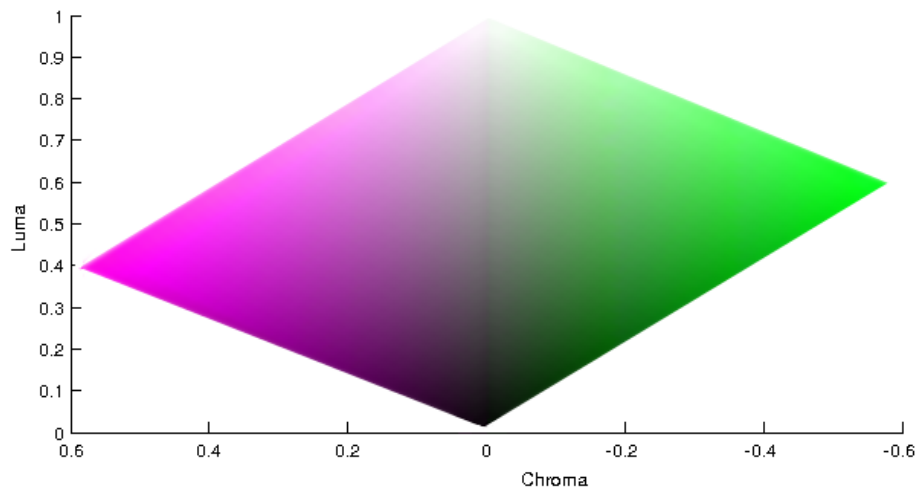


Fig. A.6.: Cross-Section of Constant Hue  $H = \frac{2\pi}{3}$  rad in Our Y'CH Color Space.

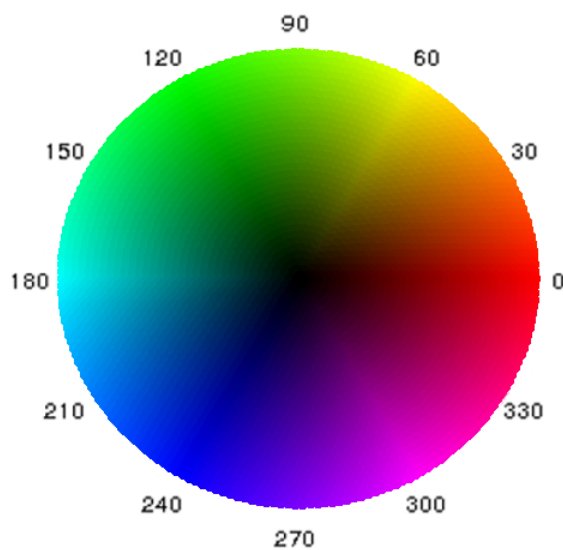


Fig. A.7.: Bottom View of Our Y'CH Color Space (Using the Arithmetic Approach).

$$H = \text{atan2}(Q, I) \quad (\text{A.5})$$

$$C = \sqrt{I^2 + Q^2}, \quad (\text{A.6})$$

$$\operatorname{atan2}(I, Q) = \begin{cases} \arctan\left(\frac{Q}{I}\right) & I > 0 \\ \pi + \arctan\left(\frac{Q}{I}\right) & Q \geq 0, I < 0 \\ -\pi + \arctan\left(\frac{Q}{I}\right) & Q < 0, I < 0 \\ \frac{\pi}{2} & Q > 0, I = 0 \\ -\frac{\pi}{2} & Q < 0, I = 0 \\ \text{undefined} & Q = 0, I = 0 \end{cases} \quad (\text{A.7})$$

Figure B.16 illustrates the bottom view of our Y'CH color space representation where the hue of different primaries can be identified. Note the hexagon shape.

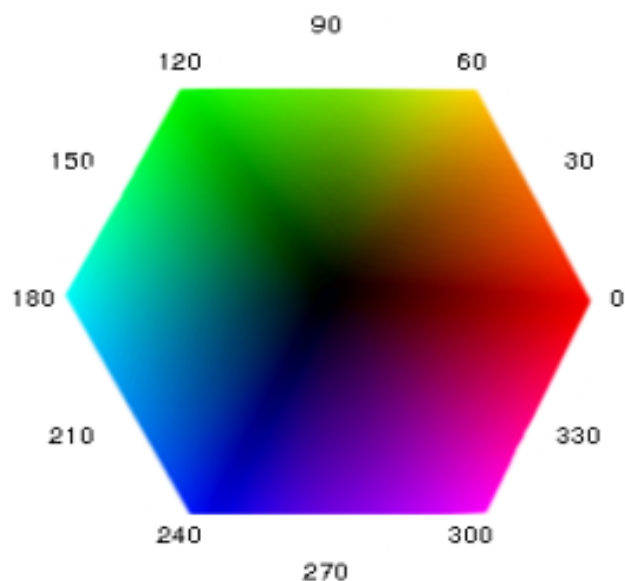


Fig. A.8.: Bottom View of Our Y'CH Color Space (Using the Trigonometric Approach).

Note that a HSL-based color space, such as Y'CH, has the disadvantage that it does not account for the complexity of the human color perception. However, since we are doing color recognition this is not an issue.

## B. EXAMPLES OF GRAFFITI COLOR IMAGE SEGMENTATION

This Appendix shows examples of Color Image Segmentation Based on Gaussian Thresholding.



Fig. B.1.: Red text:  $\tilde{H} = 0.49$  and  $\sigma_H^2 = 0.05$ .



Fig. B.2.:  $T_C = 0.04$ .



Fig. B.3.: White text:  $\tilde{Y} = 0.83$  and  $\sigma_Y^2 = 0.003$ .



Fig. B.4.:  $T_{Y_b} = 0$ ,  $T_{Y_w} = 1$ .



Fig. B.5.: Black text:  $\tilde{Y} = 0.13$  and  $\sigma_{\tilde{Y}}^2 = 0.001$ .



Fig. B.6.:  $T_{Yb} = 0$ ,  $T_{Yw} = 0.2$ .



Fig. B.7.: Blue text:  $\tilde{H} = 2.56$  and  $\sigma_H^2 = 0.034$ .



Fig. B.8.:  $T_C = 0.04$ .



Fig. B.9.: Blue text:  $\tilde{H} = 2.60$  and  $\sigma_H^2 = 0.020$ .

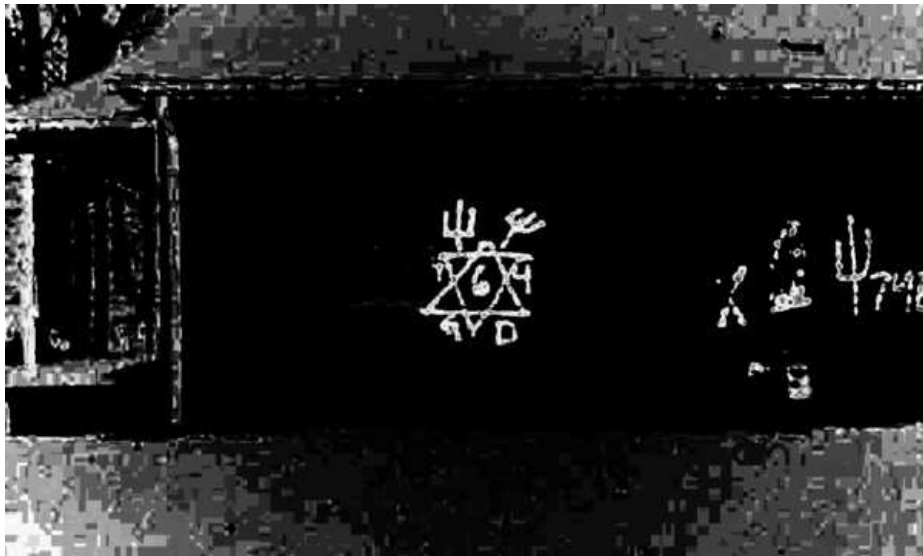


Fig. B.10.:  $T_C = 0.05$ .





Fig. B.11.: Blue text:  $\tilde{H} = 2.73$  and  $\sigma_H^2 = 0.049$ .



Fig. B.12.:  $T_C = 0.02$ .



Fig. B.13.: Black text:  $\tilde{Y} = 0.17$  and  $\sigma_{\tilde{Y}}^2 = 0.008$ .

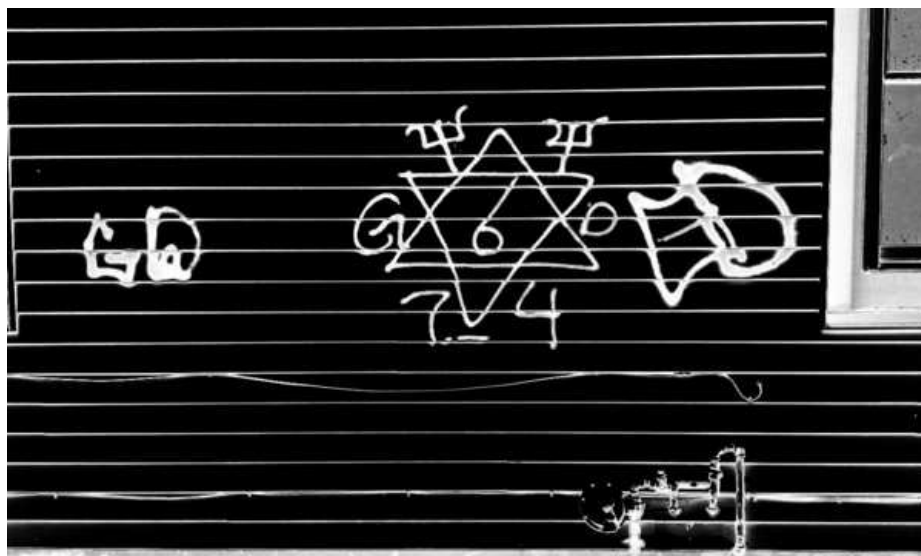


Fig. B.14.:  $T_{Yb} = 0$ ,  $T_{Yw} = 1$ .



Fig. B.15.: Black text:  $\tilde{Y} = 0.19$  and  $\sigma_{\tilde{Y}}^2 = 0.002$ .



Fig. B.16.:  $T_{Yb} = 0$ ,  $T_{Yw} = 1$ .

## C. IMAGE THRESHOLDING METHODS

This Appendix shows the comparison of three different image thresholding methods with respect to the 20 test images used in Section 5.1.4. The thresholding methods are: 1) Our proposed combination of Color Image Segmentation Based on Gaussian Thresholding and Block-Wise Gaussian Segmentation Enhancement, 2) Niblack thresholding, 3) Otsu's method. The input of our proposed method is not just the image, but additional parameters returned from our proposed Color Recognition Based on Touchscreen Tracing (Section 3.4): `boolHL` indicates if the recognized color is based on hue or luma; `medH` is the hue median; `medY` is the luma median; `varH` is the hue variance; `varY` is the luma variance. The Niblack thresholding is setup with a filter radius of 25 pixels and standard deviation threshold of 0.2. Otsu's method does not need any additional configuration.

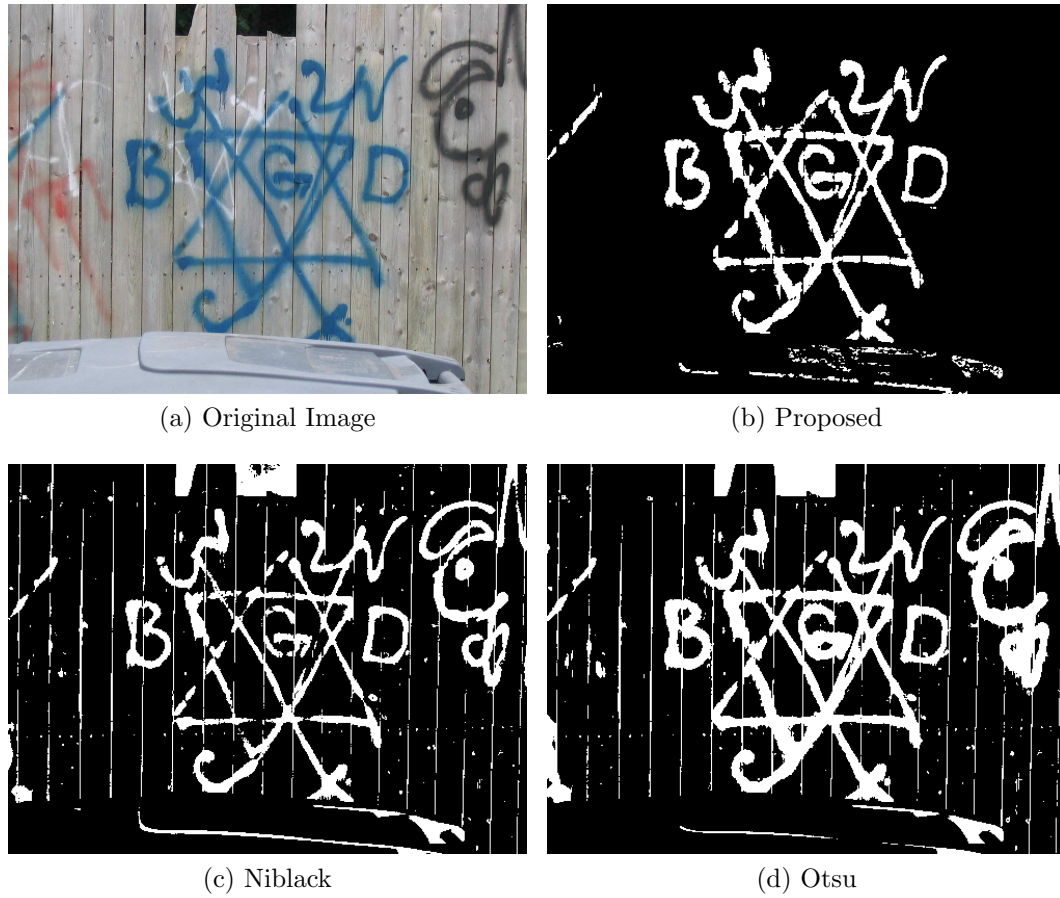


Fig. C.1.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1 \ 3.6046, 0.3486, 0.0012, 0.0013]$ .

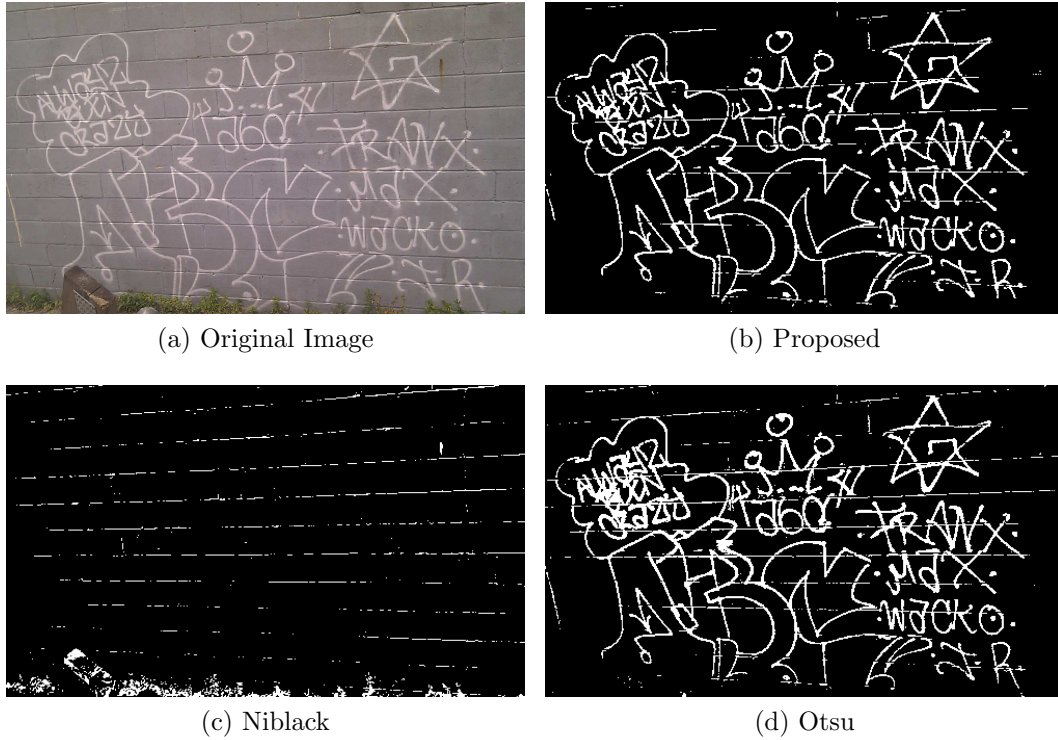
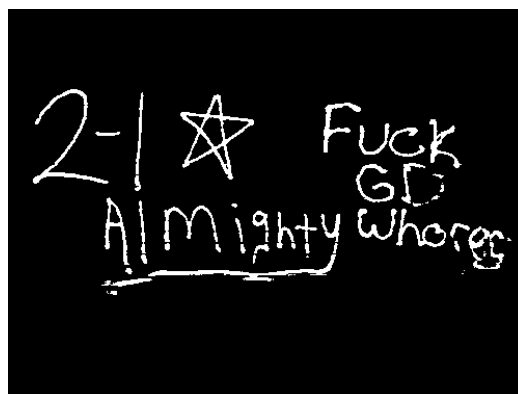


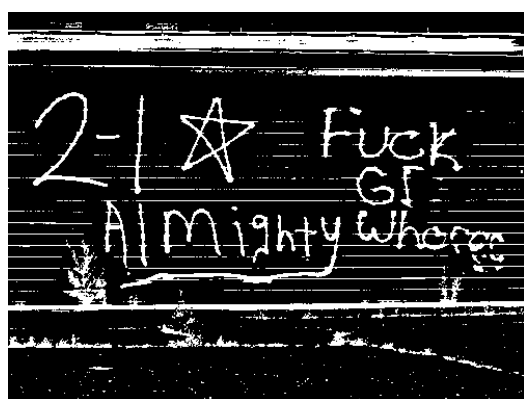
Fig. C.2.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 6.0868, 0.7381, 0.0075, 0.0033]$ .



(a) Original Image



(b) Proposed



(c) Niblack



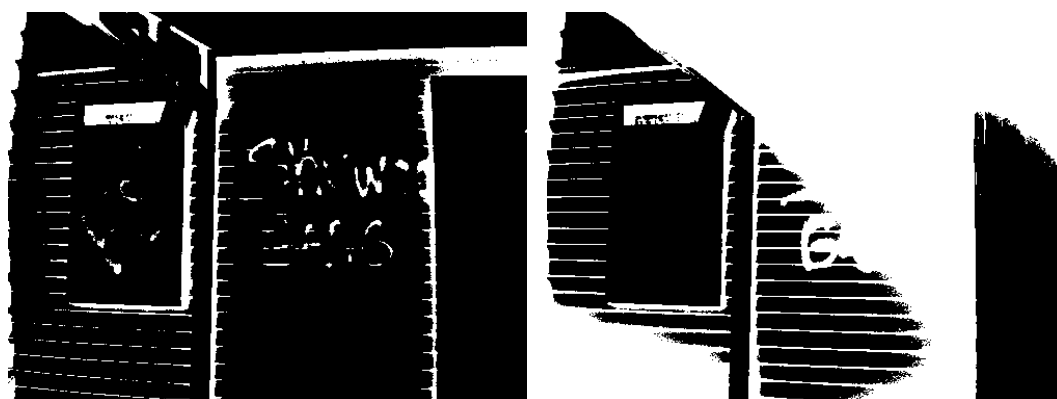
(d) Otsu

Fig. C.3.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 6.0868, 0.3298, 0.0018, 0.0010]$ .



(a) Original Image

(b) Proposed



(c) Niblack

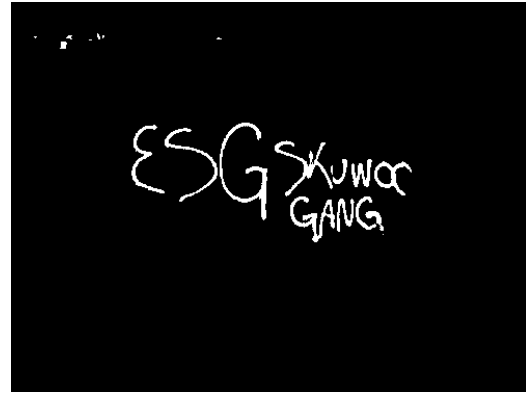
(d) Otsu

Fig. C.4.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.2448, 0.3145, 0.0107, 0.0023]$ .

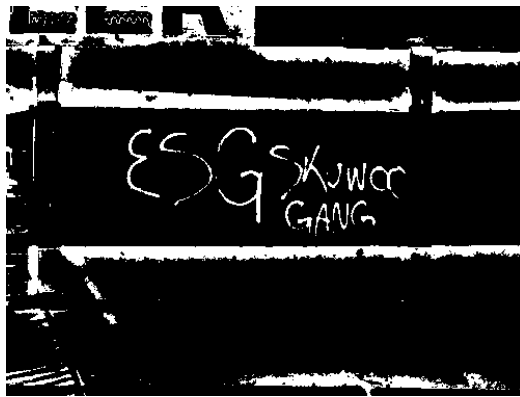




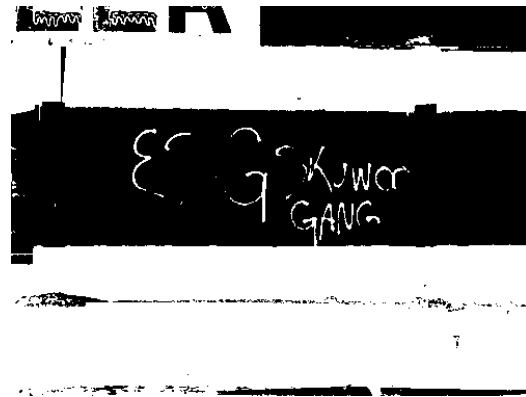
(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

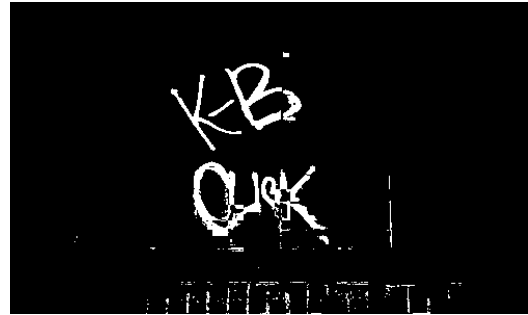
Fig. C.5.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 6.0974, 0.5332, 0.0244, 0.0011]$ .



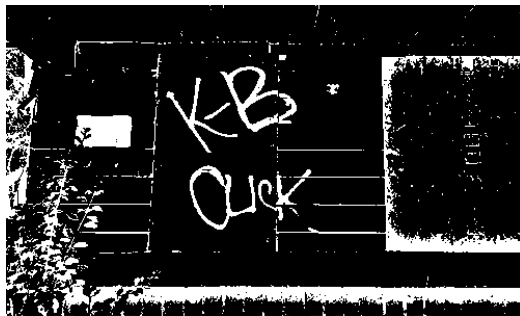
Fig. C.6.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 6.1730, 0.7483, 0.0093, 0.0037]$ .



(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.7.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.1145, 0.2670, 0.0080, 0.0028]$ .

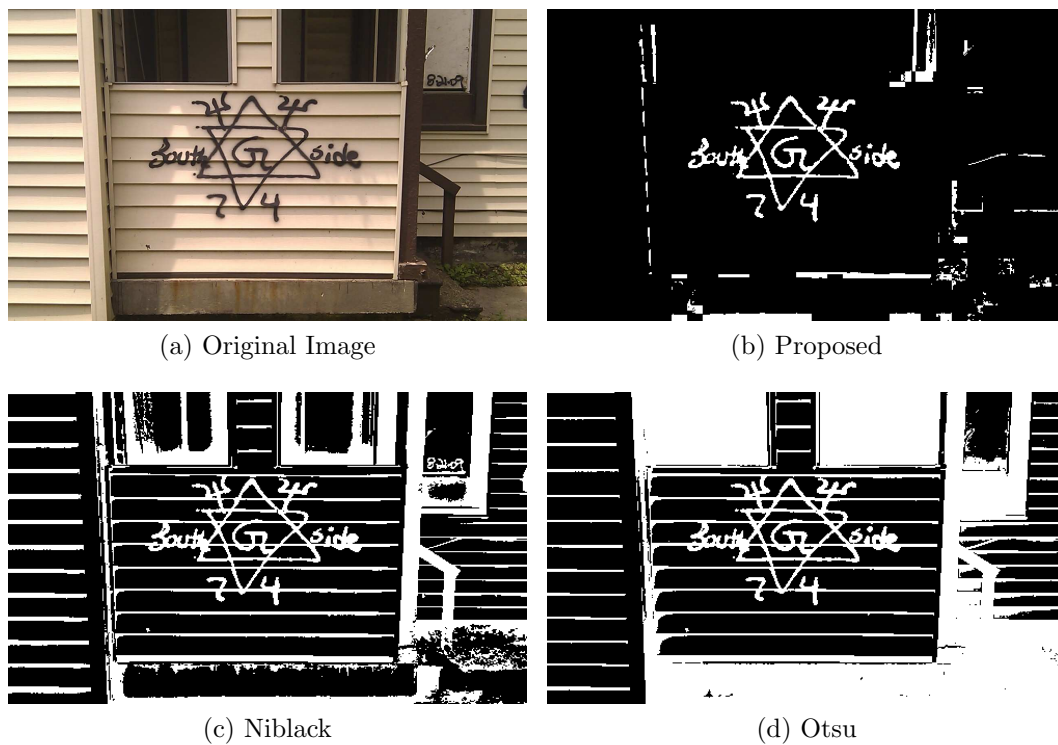


Fig. C.8.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.1848, 0.2120, 0.0656, 0.0017]$ .

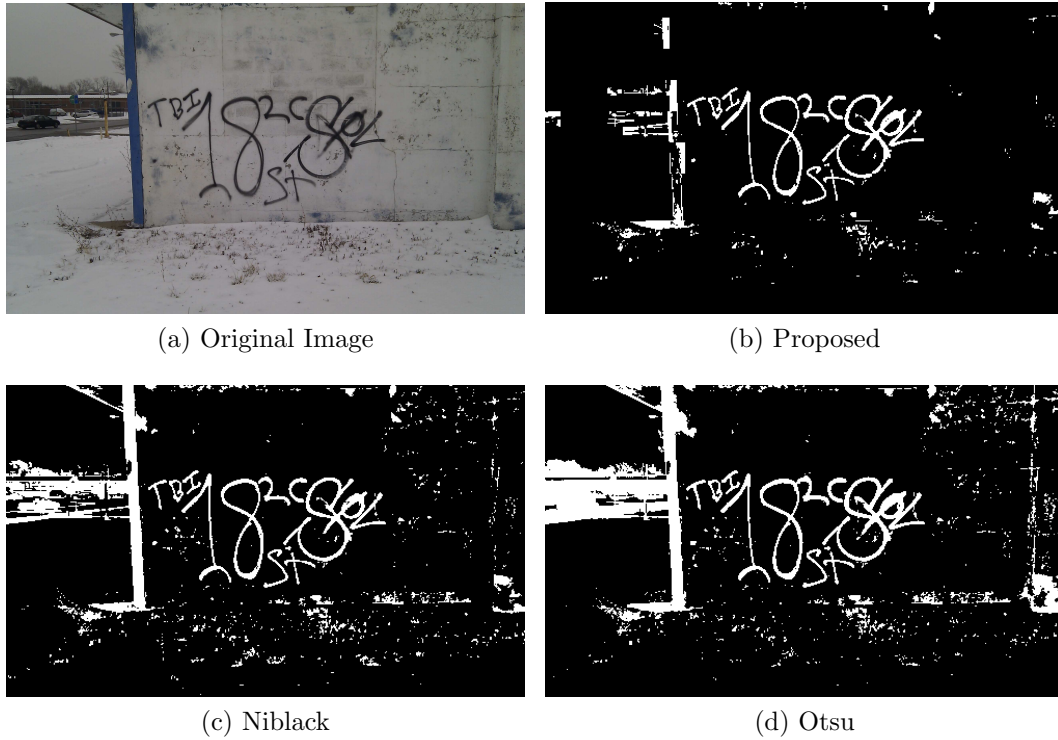
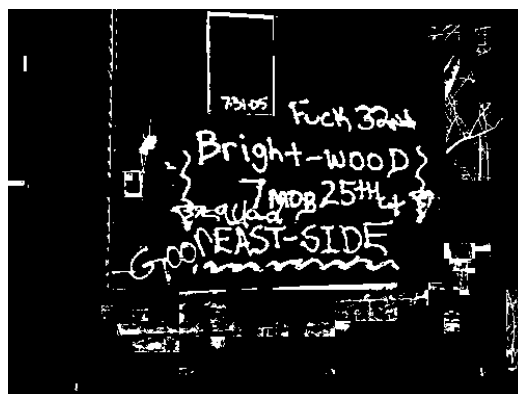


Fig. C.9.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 4.8869, 0.1329, 1.2905, 0.0029]$ .



(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.10.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 3.6070, 0.1894, 2.3252, 0.0013]$ .



(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.11.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 2.7925, 0.3618, 0.1469, 0.0028]$ .



Fig. C.12.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 1.0472, 0.2784, 2.6779, 0.0161]$ .



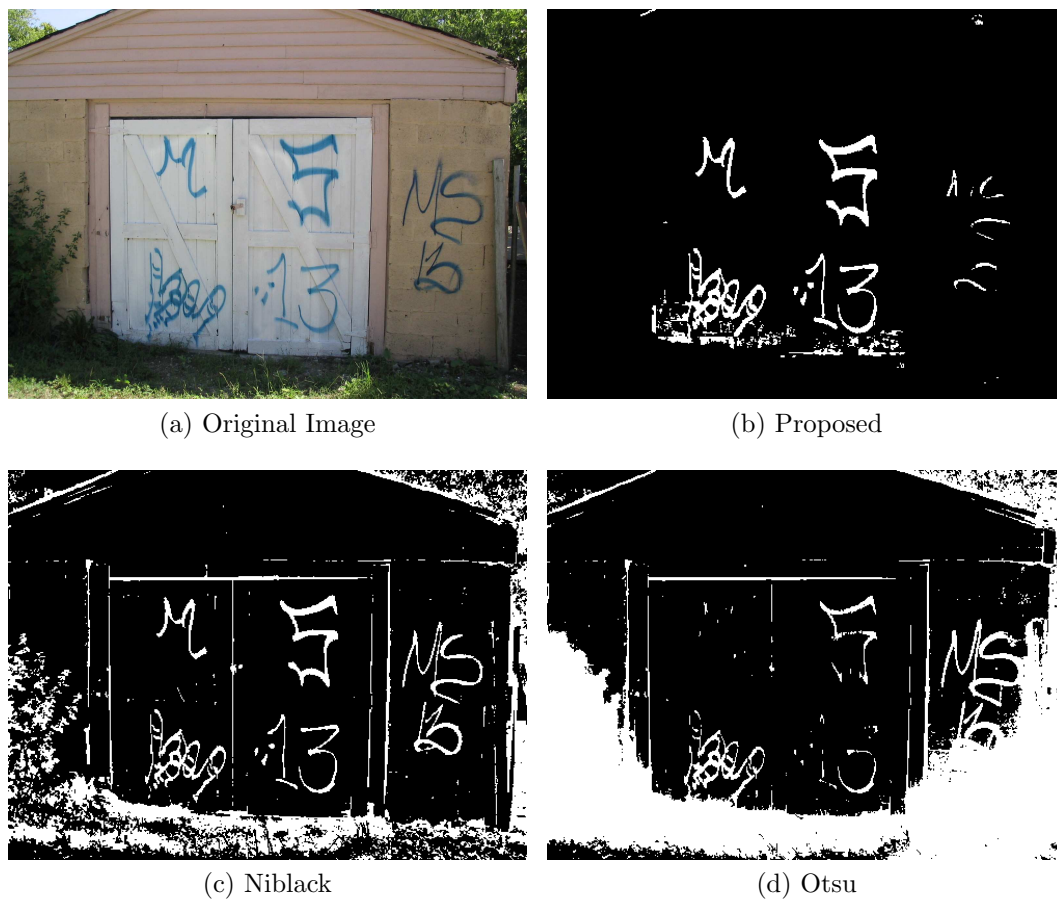


Fig. C.13.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 3.5358, 0.4344, 0.0016, 0.0028]$ .

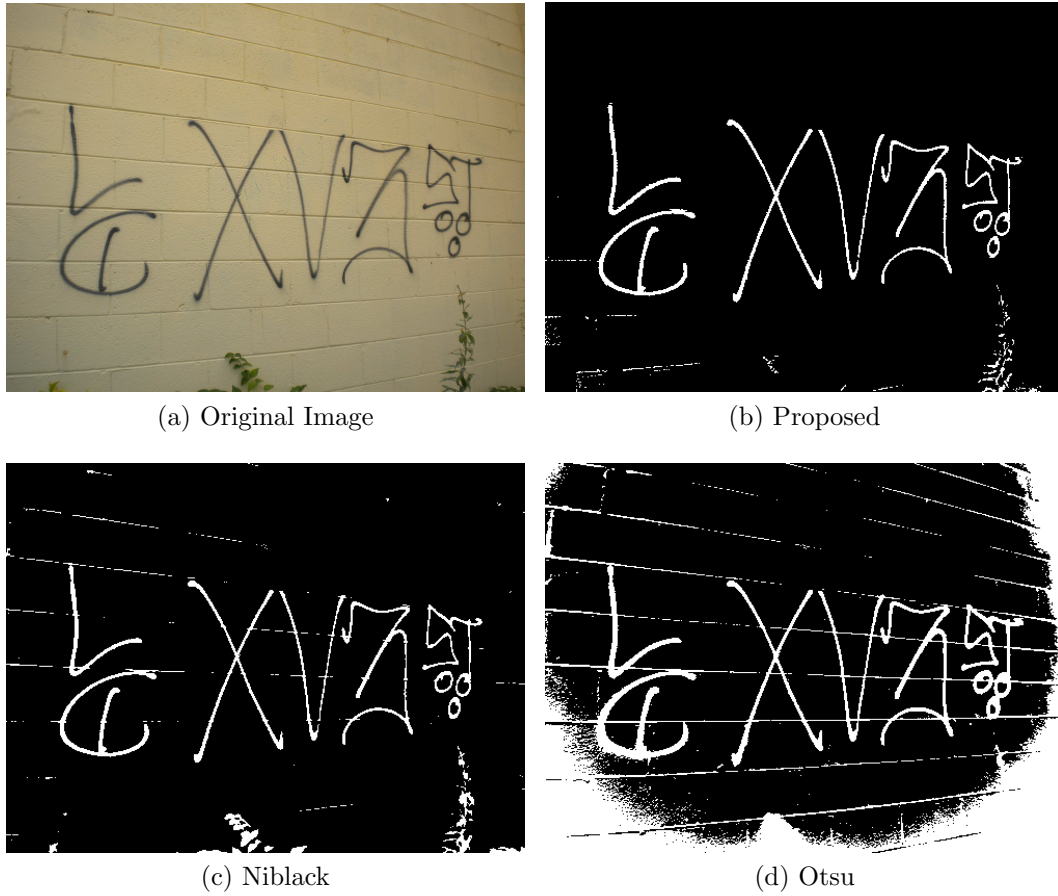
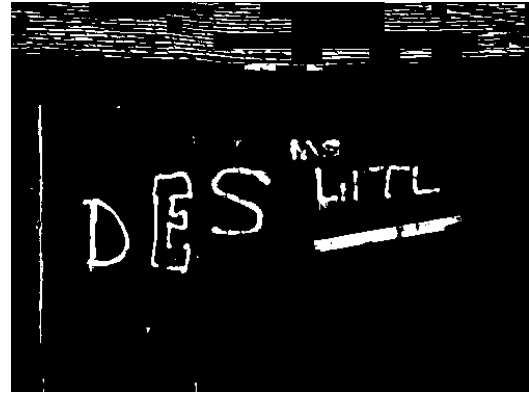


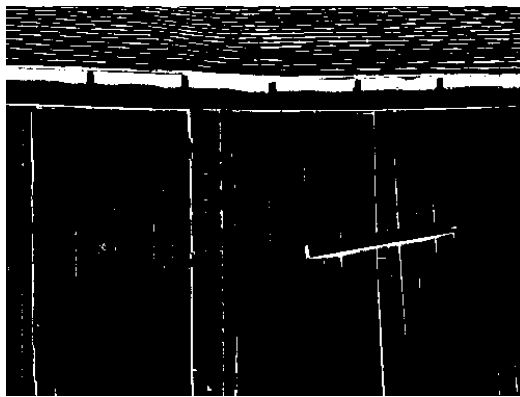
Fig. C.14.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.7854, 0.3680, 0.0250, 0.0019]$ .



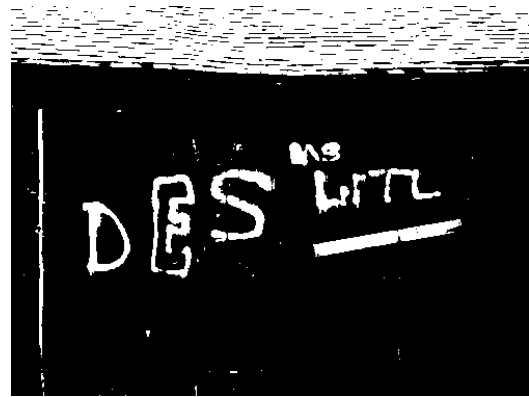
(a) Original Image



(b) Proposed



(c) Niblack

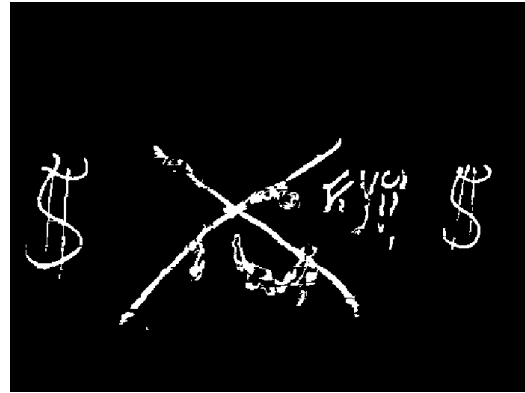


(d) Otsu

Fig. C.15.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 4.8171, 0.8821, 0.3069, 0.0046]$ .



(a) Original Image



(b) Proposed



(c) Niblack

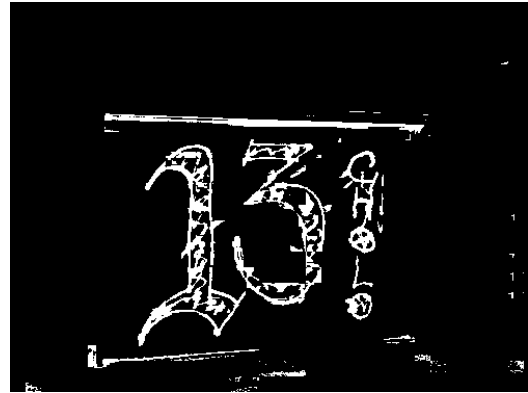


(d) Otsu

Fig. C.16.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.0423, 0.3018, 0.0012, 0.0018]$ .



(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.17.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 0.1309, 0.2317, 0.3181, 0.0093]$ .

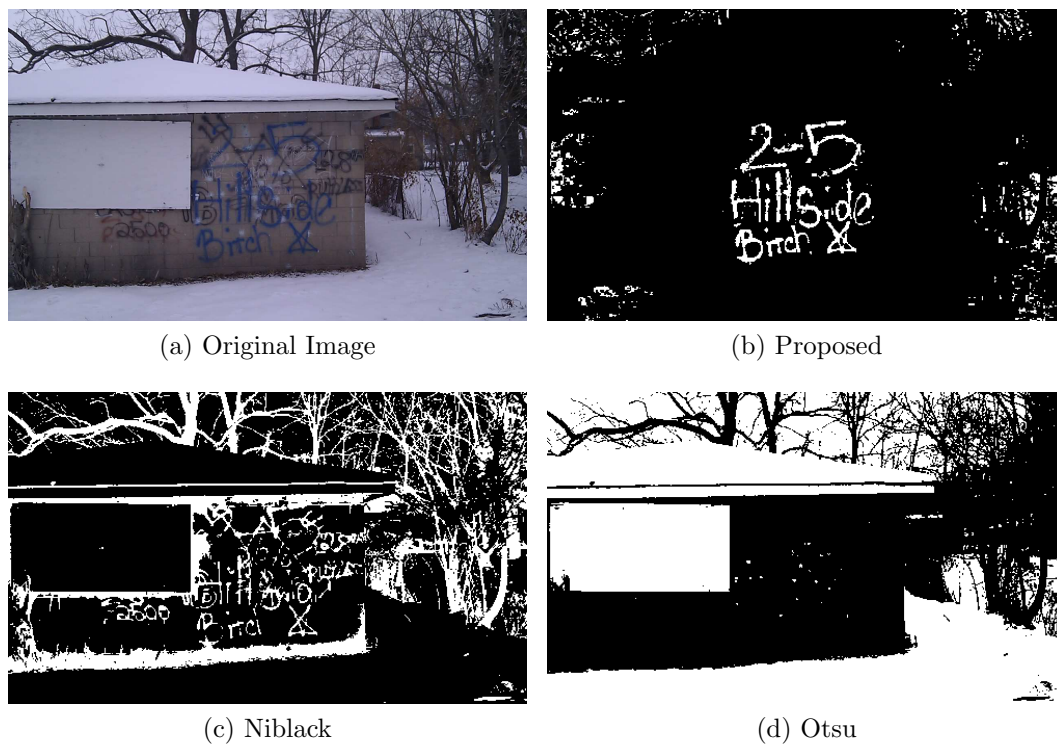
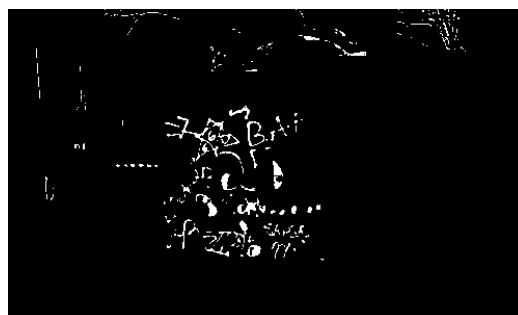


Fig. C.18.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 4.0075, 0.1993, 0.0021, 0.0015]$ .



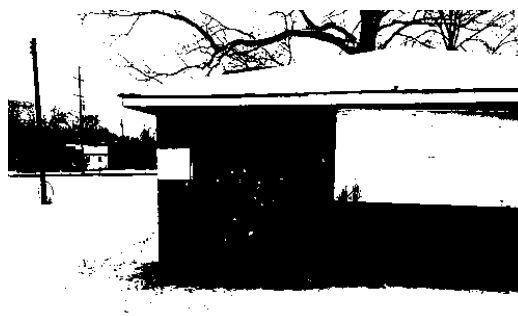
(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

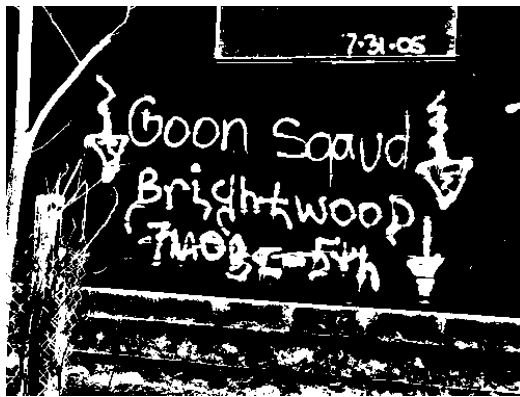
Fig. C.19.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 3.9924, 0.1886, 0.1030, 0.0014]$ .



(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.20.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [1, 0.1496, 0.3147, 0.0049, 0.0022]$ .

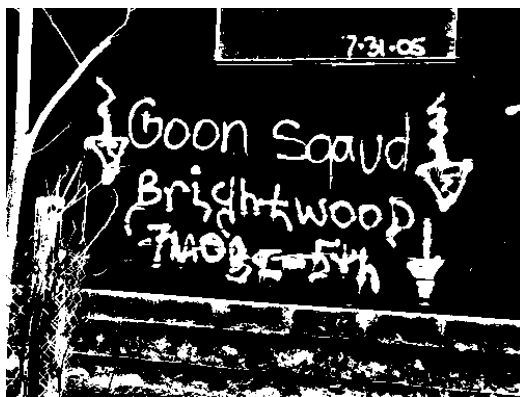




(a) Original Image



(b) Proposed



(c) Niblack



(d) Otsu

Fig. C.21.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 1.0472, 0.1529, 1.7701, 0.0005]$ .

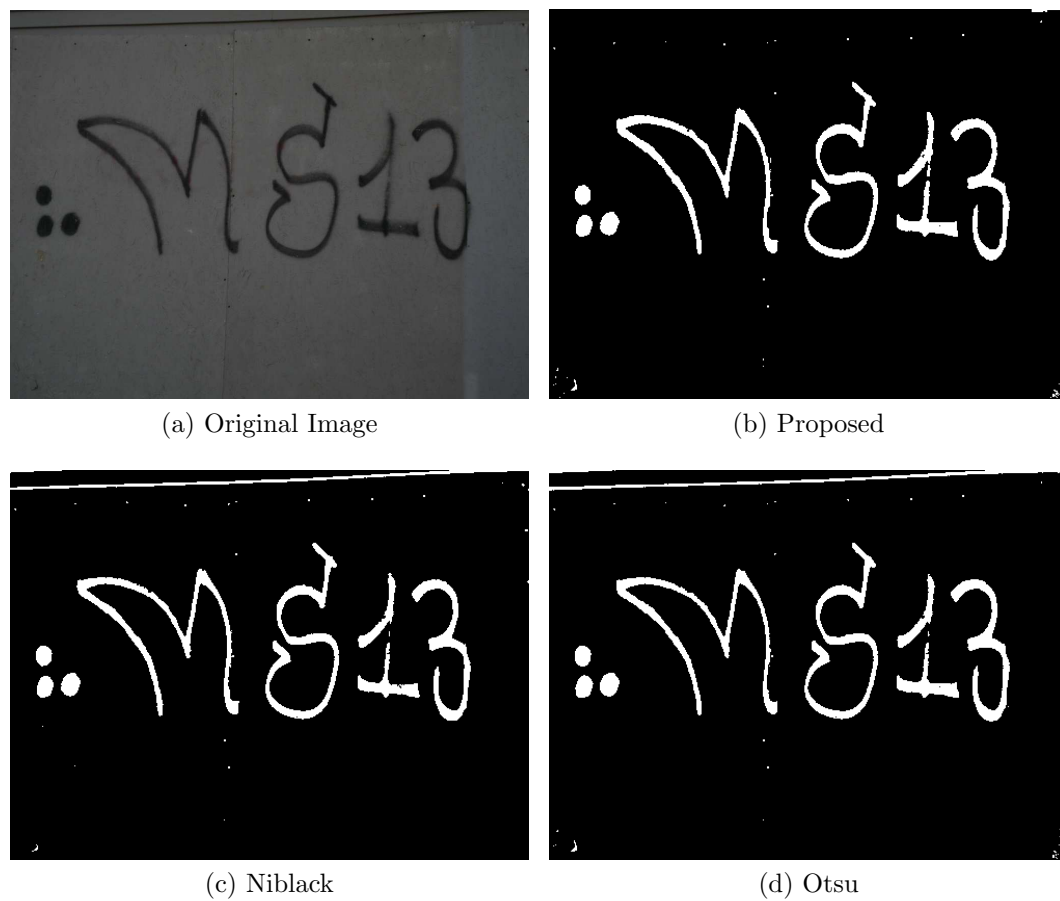


Fig. C.22.: For Proposed Method:  $[\text{boolHL}, \text{medH}, \text{medY}, \text{varH}, \text{varY}] = [0, 2.6180, 0.1305, 2.3481, 0.0019]$ .

## D. GARI DATABASE TABLES

This Appendix describes the GARI database tables in more detail.

Table D.1: EXIF data fields in Table *images*.

EXIF field	Description
filesize	Size of the image (bytes)
filedatetime	Date and time of capture
resolutionheight	Height of image (px)
resolutionwidth	Width of image (px)
focallength	Focal Length of camera's optical system
isoequiv	ISO equivalent value used
cameramake	Camera make
cameramodel	Camera model
gpsaltitude	GPS altitude
gpslongitude	GPS longitude
gpslatitude	GPS latitude
xresolution	DPI in the width direction
yresolution	DPI in the height direction
ycbcrpositioning	Position of the YCbCr components
fnumber	F number
compressedbitsperpixel	Compressed bits per pixel
exposuretime	Exposure time (seconds)
exposurebias	Exposure bias (APEX)
aperture	Lens aperture (APEX)
meteringmode	Metering mode

flash	Status of flash when the image was shot
interoperabilityoffset	Interoperability offset
sensingmethod	Sensing method
customrendered	Use of special processing on image data
whitebalance	White balance
digitalzoomratio	Digital zoom ratio
exposuremode	Exposure mode

Table D.2: Image location fields in Table *images*.

Field	Description
country	Country (given GPS coordinates)
state	State (given GPS coordinates)
county	County (given GPS coordinates)
city	City (given GPS coordinates)
zip	ZIP code (given GPS coordinates)
address	Address (given GPS coordinates)

Table D.3: Graffiti analysis fields in Table *images*.

Field	Description
gangnameia	Gang name from IA <sup>1</sup>
gangnamegt	Gang name from GT <sup>2</sup>
gangidia	Gang ID from IA
gangidgt	Gang ID from GT
gangmembernameia	Gang member name from IA
gangmembernamegt	Gang member name from GT

<sup>1</sup>IA: Image Analysis<sup>2</sup>GT: Ground Truth

gangmemberidia	Gang member ID from IA
gangmemberidgt	Gang member ID from GT

Table D.4: Image information fields in Table *images*.

Field	Description
imageid	Image ID
path	Path to the image file
firstrespondername	First responder name
firstresponderid	First responder ID
comment	Comments about graffiti
webupload	File uploaded from desktop version (boolean)
realcoords	Image has real GPS coordinates (boolean)
filedatetimeupload	Date and time the file was uploaded to the database
lastmodified	Date and time a fields was last modified
lastmodifiedname	First responder that last modified a field
istattoo	Boolean to indicate if image is graffiti or tattoo
isprison	Boolean to indicate if image was taken at a prison
prisonname	Name of the prison where the image was taken

Table D.5: User information fields in Table *users*.

Field	Description
id	User ID
password	MD5 hash of user's password
name	User's name
admin	User is administration (boolean)
first	First login (boolean)

gmail	Gmail address
email	Alternative email address
affiliation	User affiliation
android	Has Android application (boolean)
comments	Comments about user

Table D.6: Image blobs information fields in Table *imageBlobs*.

<b>Field</b>	<b>Description</b>
imageid	Image ID
blobid	Blob ID for a particular image ID
componentid	Component ID for a particular blob ID
colorid	Color ID for a particular component ID
crossedout	Boolean to determine if the component is crossed-out
upsidedown	Boolean to determine if the component is upside-out

## E. MERGE DATABASE TABLES

This Appendix describes the MERGE database tables in more detail.

Table E.1: EXIF data fields in Table *images*.

EXIF field	Description
filesize	Size of the image (bytes)
filedatetime	Date and time of capture
resolutionheight	Height of image (px)
resolutionwidth	Width of image (px)
focallength	Focal Length of camera's optical system
isoequiv	ISO equivalent value used
cameramake	Camera make
cameramodel	Camera model
gpsaltitude	GPS altitude
gpslongitude	GPS longitude
gpslatitude	GPS latitude
xresolution	DPI in the width direction
yresolution	DPI in the height direction
ycbcrpositioning	Position of the YCbCr components
fnumber	F number
compressedbitsperpixel	Compressed bits per pixel
exposuretime	Exposure time (seconds)
exposurebias	Exposure bias (APEX)
aperture	Lens aperture (APEX)
meteringmode	Metering mode

flash	Status of flash when the image was shot
interoperabilityoffset	Interoperability offset
sensingmethod	Sensing method
customrendered	Use of special processing on image data
whitebalance	White balance
digitalzoomratio	Digital zoom ratio
exposuremode	Exposure mode

Table E.2: Image location fields in Table *images*.

Field	Description
country	Country (given GPS coordinates)
state	State (given GPS coordinates)
county	County (given GPS coordinates)
city	City (given GPS coordinates)
zip	ZIP code (given GPS coordinates)
address	Address (given GPS coordinates)

Table E.3: Image information fields in Table *images*.

Field	Description
imageid	Image ID
path	Path to the image file
firstrespondername	First responder name
filedatetimeupload	Date and time the file was uploaded to the database
issign	Boolean to indicate if image is sign or scene



Table E.4: User information fields in Table *users*.

<b>Field</b>	<b>Description</b>
id	User ID
password	MD5 hash of user's password
name	User's name
admin	User is administration (boolean)
first	First login (boolean)
gmail	Gmail address
email	Alternative email address
affiliation	User affiliation
android	Has Android application (boolean)
comments	Comments about user

Table E.5: Fields in Table *class*.

<b>Field</b>	<b>Description</b>
clid	Class ID
text	Text describing class number and name
clnumber	Class number

Table E.6: Fields in Table *colorids*.

<b>Field</b>	<b>Description</b>
colorid	Color ID
colorname	Color name

Table E.7: Fields in Table *colorpages*.

---

Field	Description
colorid	Color ID
guide	Guide page number

Table E.8: Fields in Table *placard*.

Field	Description
pid	Placard ID
unid	UNID
clid	Class ID
sid	Symbol ID

Table E.9: Fields in Table *symbol*.

Field	Description
sid	Symbol ID
text	Symbol description

Table E.10: Fields in Table *textcolors*.

Field	Description
textid	Text ID for hazardous material types
colorid	Color ID

Table E.11: Fields in Table *textids*.

Field	Description
textid	Text ID
text	Hazardous material description

Table E.12: Fields in Table *textpages*.

Field	Description
textid	Text ID
guide	Guide page number

Table E.13: Fields in Table *unids*.

Field	Description
unids	UNID
guide	Guide page number
material	Material type
iso	Included in the International Organization for Standardization (ISO) (boolean)

Table E.14: Fields in Table *vw01\_orange\_page*.

Field	Description
guide_number_cd	Guide page number
guide_page_name_txt	Guide page title
category_txt	Hazmat sign category
sub_category_txt	Hazmat sign subcategory
detail_txt	Page details

Table E.15: Fields in Table *vw03\_yellow\_page*.

Field	Description
un_number	UNID
guide_number_cd	Guide page number
polymerization_ind	Polymerization index

dangerous_good_name.txt	Dangerous good description
dangerous_good_id	Dangerous good ID

Table E.16: Fields in Table *vw05\_water\_reactive\_materials*.

Field	Description
un_number	UNID
guide_number_cd	Guide page number
dangerous_good_name.txt	Dangerous good description
chemical_symbol	Chemical symbol
tih_gas_produced	Toxic-by-Inhalation (TIH) gas produced
dangerous_good_id	Dangerous good ID
polymerization_ind	Polymerization index

Table E.17: Fields in Table *vw06\_tiiapad*.

Field	Description
dangerous_good_id	Dangerous good ID
dangerous_good_name.txt	Dangerous good description
un_number	UNID
circumstance_type.txt	Situation when condition applies
guide_number_cd	Guide page number
polymerization_ind	Polymerization index
simetric	Small spills - Isolation distance (metric)
spdmetric	Small spills - Protective distance - Day (metric)
spnmetric	Small spills - Protective distance - Night (metric)
limetric	Large spills - Isolation distance (metric)

lpdmetric	Large spills - Protective distance - Day (metric)
lpnmetric	Large spills - Protective distance - Night (metric)
siimperial	Small spills - Isolation distance (imperial)
spdimperial	Small spills - Protective distance - Day (imperial)
spnimperial	Small spills - Protective distance - Night (imperial)
liimperial	Large spills - Isolation distance (imperial)
lpdimperial	Large spills - Protective distance - Day (imperial)
lpnimperial	Large spills - Protective distance - Night (imperial)

## F. GARI IMAGE ACQUISITION PROTOCOL

This Appendix describes the protocol used for acquiring test images for the GARI database. The images are used for testing various functions of the GARI system.

- Persons involved
    - 2 GARI staff members
    - 1 or more persons from Police Department
  - Equipment/Materials needed
    - Pens or pencils
    - 2 Digital Camera (1MPx and above)
    - 2 Tripods
    - 2 Mobile Telephone with Android OS
      - \* Built-in camera (1MPx and above)
      - \* GPS receiver
      - \* optional: Data plan
    - 1 GPS receiver
    - Graffiti Information Forms
    - Fiducial Markers
    - Image Checklist
    - 1 Purdue University owned laptop
    - 1 External hard drive
- 1) Preliminaries (Internet connection required)
    - a) Check time setting on the two Android mobile telephones, the two digital cameras, and the GPS receiver using the Purdue University owned laptop, and ensure they are in sync with the GARI server.

- b) Make sure the two Android mobile telephones, the two digital cameras, and the GPS receiver batteries are fully charged.
  - c) Verify all equipment/materials above are available.
  - d) Make sure the settings of the two digital cameras are set to default by finding the appropriate menu option.
  - e) Turn flash feature off on the two Android mobile telephones built-in cameras and the two digital cameras.
  - f) Make sure zoom and macro features are not enabled on the two Android mobile telephones built-in cameras and the two digital cameras.
  - g) Assign each person an ID number, and record it on the Graffiti Information Form.
  - h) Record person's name and affiliation on the Graffiti Information Form.
- 2) Set up environment
- a) Stand up in front of the graffiti, far enough so that the cameras can capture all the content, preferably perpendicular to the surface containing the graffiti. Some angle margin is permitted ( $\theta$  spherical degrees), as shown in Figure F.1 and Figure F.1. This angle should be small enough so that the graffiti contents can be identified properly.
  - b) Make sure weather condition does not prevent seeing the graffiti.
  - c) Place the fiducial marker in a spot that would be 20 inches away and parallel to the surface containing the graffiti, as shown in Figure F.1 and Figure F.2. It should not block the graffiti contents.
  - d) Make sure there are not any objects between the camera and the graffiti that obstruct partially or totally the view of the graffiti.
  - e) Record Date (MM/DD/YYYY), Time (HH:MM:SS) and GPS coordinates (latitude, longitude and altitude, with six digit precision) on the Graffiti Information Form. Obtain the information from the GPS receiver.

- f) Record neighborhood description on the Graffiti Information Form. Specify street name(s) and landmarks in the area near the graffiti.
- g) Proceed to take image. For each graffiti, take six images, using
- Android mobile telephone 1
  - Android mobile telephone 1
  - Android mobile telephone 2
  - Digital camera 1 with tripod
  - Digital camera 1 without tripod
  - Digital camera 2 with tripod
  - Digital camera 2 without tripod
- h) For each graffiti, record the device(s) used on the Graffiti Information Form.
- 3) Taking an image of a graffiti
- 3.1) Taking image of a graffiti using an Android mobile telephone
- a) Launch GARI application on the Android mobile telephone and assign an Image Taker ID, corresponding to the one assigned in step 1. Preliminaries.
  - b) Select the “Capture Image” option from the GARI application main menu. The camera activity is then initialized.
  - c) Prepare for taking the image (position of the camera as desired, within the recommended distance and angle from the graffiti). Make sure all the contents of the graffiti and the entire fiducial marker can be seen on the device screen.
  - d) Take an image of the graffiti, trying to maintain the device’s position, as much as possible.
  - e) If the image does not meet the requirements noted in the Image Checklist, the image should be retaken.
  - f) If location available through WiFi/GSM/GPS the GPS coordinates will be automatically stored in the image. If no location method available,



will receive a message: “No NETWORK/GPS found. Check coordinates manually!”. Ignore it, since the GPS coordinates have already been recorded on the Graffiti Information Form.

- g) Crop the image if desired.
- h) Select the “Send to Server” option from the GARI application main menu. If no Internet connection available, will receive a message: “No internet connection available”. It means the image has not been uploaded to the server. However, the image is still in the Android mobile telephone SD card, and it can be copied to a computer at the end of the session (Section 5.a of the protocol), and uploaded in the future. If the image has not been uploaded to the server, check the box “Not Successfully Uploaded” on the Graffiti Information Form.

### 3.2) Taking image of a graffiti using a digital camera

- a) If a tripod is used, attached it to the digital camera, and adjust it so the digital camera is at the same position as if it is held without using the tripod.
- b) Prepare for taking the image (position of the camera as desired, within the recommended distance and angle from the graffiti). Make sure all the contents of the graffiti and the entire fiducial marker can be seen on the device screen.
- c) Take an image of the graffiti, trying to maintain the device’s position, as much as possible.

### 4) Completing the Graffiti Information Form (Figure F.3)

- a) Fill the “Ground-truth graffiti information” section on the Graffiti Information Form with ground-truth information associated with the graffiti, if known. It includes:
  - Graffiti color(s): color or colors of the graffiti contents.

- Gang Name(s): name of the gang or gangs that participated on the drawing of the graffiti.
- Gang Member(s): name of the gang member or gang members that participated on the drawing of the graffiti.
- Target Gang Name(s): name of the gang or gangs that are targeted in the graffiti.
- Target Gang Member(s): name of the gang member or gang members that are targeted in the graffiti.
- Symbol(s): description of the symbol(s) in the graffiti, including color, position in the graffiti (e.g. next to the gang name), orientation (e.g. upside down fork), and possible meaning.
- Other content(s): description of other relevant contents of the graffiti (e.g. crossed letters, nicknames), including color, position in the graffiti (e.g. crossed C on the right of BERO), and possible meaning.
- Comments: additional information of the graffiti that does not fit in the previous subsections of the “Ground-truth graffiti information” section.

b) Fill the “General Comments” section on the *Graffiti Information Form* with additional comments that do not fit in all the previous sections.

5) End of the session procedures

- a) Copy all the images taken with the Android mobile telephones (stored in the GARI folder) and with the two digital cameras to a Purdue University owned laptop and to an external hard drive.
- b) Take cards out of the digital cameras and reformat them.
- c) Ensure the Purdue University owned laptop and the two digital cameras are synced.
- d) Recharge laptop and camera batteries.
- e) Store fiducial markers and other materials in a safe place for later use.

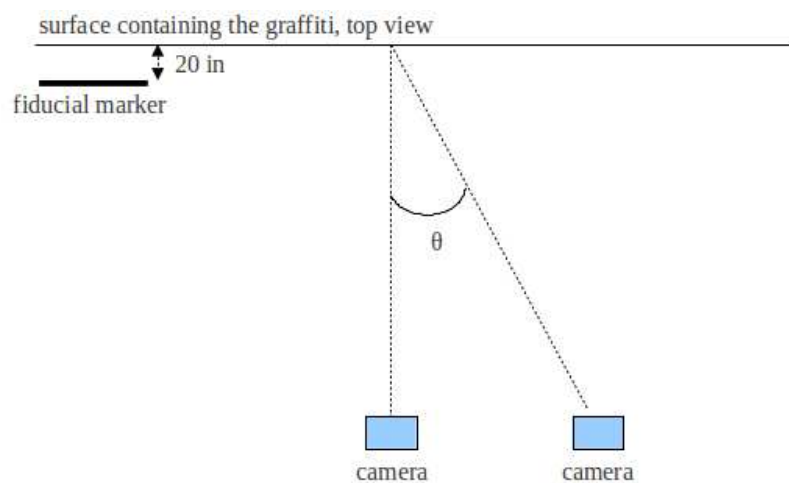


Fig. F.1.: Top view of the setup environment.

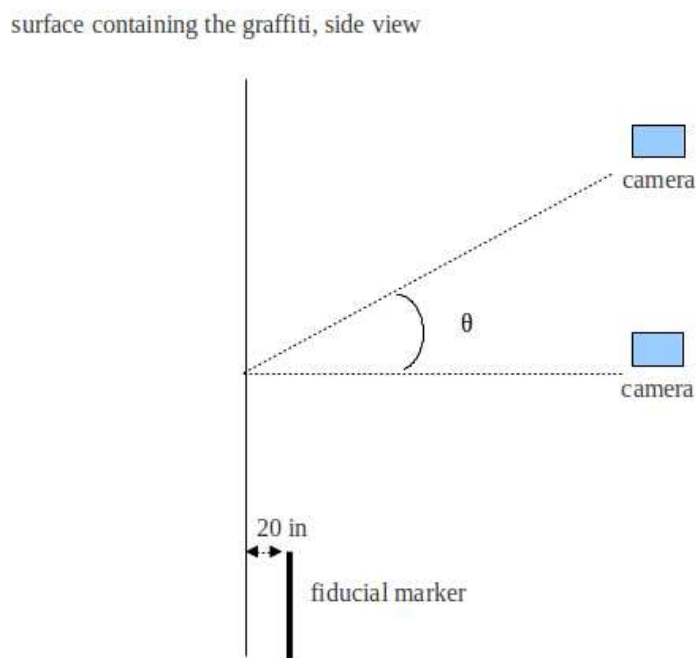


Fig. F.2.: Side view of the setup environment.

### Graffiti Information Form

#### First Responder information

Name: ID: Affiliation:

---

#### Basic image information

Date: / / Time: : :

*GPS Coordinates*

Latitude: Longitude: Altitude:

Neighborhood Description:

*Device(s) used*

Android Mobile Telephone  Camera with tripod  Camera without tripod

---

#### Ground-truth graffiti information

Color(s):

Gang Name(s):

Gang Member(s):

Target Gang Name(s):

Target Gang Member(s):

Symbol(s):

Other Content(s):

Comments:

---

General Comments:

Fig. F.3.: Graffiti Information Form.

## G. MERGE IMAGE ACQUISITION PROTOCOL

This Appendix describes the protocol used for acquiring test images for the MERGE database. The images are used for testing various functions of the MERGE system.

- Persons involved
  - 1 MERGE staff member
- Equipment/Materials needed
  - Pens or pencils
  - 1 Mobile Telephone with Android OS
    - \* Built-in camera (1MPx and above)
    - \* 3G/4G/WiFi data connection
    - \* GPS
  - 1 Digital Camera with Android OS
    - \* 3G/4G/WiFi data connection
    - \* GPS
  - Image Recording Forms
  - External Hard Drive

### 1) Preliminaries (Internet connection required)

- a) Check Date and Time settings on the Android mobile telephone and the digital camera, and ensure date, time, and time zone are set to automatic (network-provided).
- b) Make sure the Android mobile telephone and the digital camera's batteries are fully charged.
- c) Make sure the GPS is enabled on the Android mobile telephone and the digital camera.

- d) Verify all equipments/materials above are available.
  - e) Turn flash feature off on the Android mobile telephone and the digital camera.
  - f) Note: The Image Taker will need to fill out an Image Recording Form for each hazmat sign.
- 2) Set up environment
- a) Stand in front of the hazmat sign, far enough so that the camera can capture all the content, up to 200 feet from the sign for the Android mobile phone, and up to 500 feet from the sign for the digital camera. Stand preferably perpendicular to the surface containing the sign. Limited angles are permitted (45 degrees), as shown in Figure G.1.
  - b) Make sure weather conditions do not obstruct the view of the hazmat sign.
  - c) Make sure there are no objects between the camera and the hazmat sign that partially or completely obstruct the view of the hazmat sign.
- 3) Taking Images of Hazmat Signs
- a) Launch the MERGE application on the Android mobile telephone and the digital camera, and login using the Image Taker's ID and password. If this is the first time that the Image Taker is logging into the application, an Internet connection will be required to connect with the MERGE database on the server. From then on, the Image Taker's credential will be stored on the Android device for future use without an Internet connection.
  - b) Select the "Capture Image" option from the MERGE main screen. The camera activity is then initialized. Note that a new directory with the name MERGE will be created on the Android device's image gallery, where all the images taken using the MERGE application will be stored. Please refer to this directory when copying the images to the external hard drive (Section 5a).

- c) Prepare for taking the image (position the camera as desired, within the recommended distance and angle from the hazmat sign). Make sure all the contents of the hazmat sign can be seen on the device screen.
- d) Take an image of the hazmat sign, trying to hold the device as much as stable. The image can be retaken as many times as needed by tapping on the retake option on the camera activity.
- e) Tap on the OK button on the camera activity to save the current image. The image will be automatically uploaded to the server and analyzed. The Image Taker should see a notification dialog with the text “Uploading image...” followed by another notification dialog with the text “Analyzing image...”. If no Internet connection is available at the time, a warning dialog with the text “No Internet connection available” will be shown to the Image Taker. However, the image is stored in the Android device, and it can be uploaded and analyzed in the future using the “Browse Image” option from the MERGE main screen. If the image has not been uploaded to the server, check the box “Not Successfully Uploaded” on the Image Recording Form.
- f) If no Internet connection is available at the time, a warning dialog with the text “No Internet connection available” will be shown to the Image Taker. In this case, the captured image is stored in the device, and it can be uploaded and analyzed in the future using the “Browse Image” option from the MERGE main screen.
- g) Please take different images for the same sign, at different distances (10-150 ft) and angles of view (0-45°), and then write down an Image ID shown on the top bar / pop-up window on the result screen, an approximate Angle of View between your viewpoint and the perpendicular plane of the hazmat sign’s surface, and an approximate Distance from your viewpoint to the hazmat sign on the Image Recording Form (e.g., 123456, 15°, and 125 ft).
- h) Please take at least one image with No Zoom when using the digital camera, and then check the box “No Zoom” on the Image Recording Form. Also

take some images using the Optical Zoom when using the digital camera (NO Digital Zoom), and then check the box “Zoom” and mark on an approximate Zoom Value in a box on the Image Recording Form (e.g., 3/4 of the entire optical zoom range).

4) Completing the Image Recording Form (Figure G.2)

- a) Record Date (MM/DD/YYYY), Starting Time (HH:MM:SS), the Make and Model of the device used to capture the images (e.g., HTC Desire) and the Image Taker’s Name and Affiliation on the Image Recording Form.
- b) Complete the “Ground Truth Information” section on the Image Recording Form with ground-truth information associated with each hazmat sign in the captured image. This includes:
  - The Total number of existing hazmat signs in the captured image
  - For each existing hazmat sign
    - Hazmat sign number of an existing hazmat sign in the captured image
    - Color(s): color(s) found in the hazmat sign (NOT including hazmat sign frame)
    - UN Identification number (UNID) (Figure G.3a)
    - Symbol (Figure G.3b)
    - Class (Figure G.3c)
    - Text (Figure G.3d)
    - Comments: Additional information of the hazmat sign that does not fit in the previous fields.
- c) Complete the “Image Analysis Results” section on the Image Recording Form with information retrieved from the server after a captured or browsed image has been analyzed. This includes:
  - The Image ID of the captured image
  - The Total number of highlighted hazmat signs from image analysis
  - For each returned hazmat sign



- Hazmat sign number of a highlighted hazmat sign shown in the result screen
- Color(s): color(s) shown in the result screen
- Text: text shown in the result screen
- No hazmat signs found: Check this box if a dialog containing “No hazmat signs found” is shown to the Image Taker after uploading an image to the server, meaning that no hazmat signs have been found in the current image.

Figures G.4 and G.5 show two examples of completed Image Recording Forms for two different cases shown in Figure G.6.

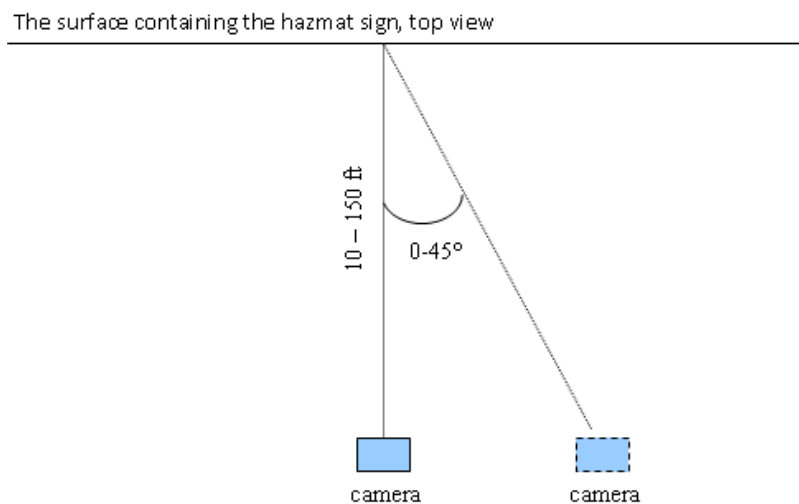


Fig. G.1.: Top view of the setup environment.

### Image Recording Form

**Image Taker Name:** \_\_\_\_\_ **ID:** \_\_\_\_\_ **Affiliation:** \_\_\_\_\_  
**Date:**        /        /        **Starting Time:**        :        :  
**Device Make:** \_\_\_\_\_ **Device Model:** \_\_\_\_\_

Ground Truth Information			Angle of View		°	Distance		ft		
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
Image Analysis Results			No Zoom [ ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
								[ ]		
Ground Truth Information			Angle of View		°	Distance		ft		
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
Image Analysis Results			No Zoom [ ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
								[ ]		
Ground Truth Information			Angle of View		°	Distance		ft		
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
Image Analysis Results			No Zoom [ ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
								[ ]		
Ground Truth Information			Angle of View		°	Distance		ft		
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
Image Analysis Results			No Zoom [ ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
								[ ]		

Fig. G.2.: Image Recording Form.

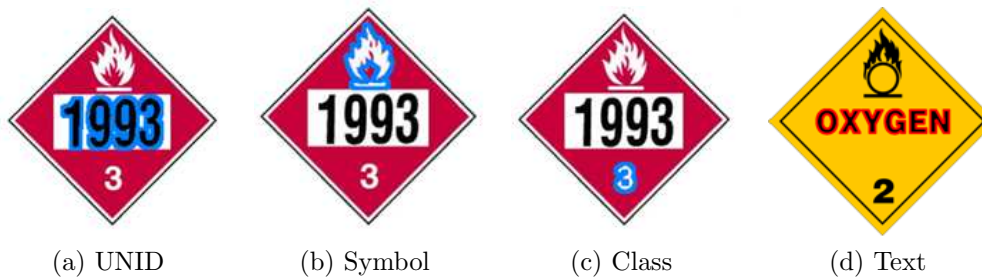


Fig. G.3.: Hazmat sign identifiers.

Ground Truth Information			Angle of View		Distance					
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
120130	1	2	WHITE	1017	Skull	2	No Text			
Image Analysis Results			No Zoom [ X ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
120130	1	2	WHITE	N/A	N/A	N/A	No Text	[ ]		
Ground Truth Information			Angle of View		Distance					
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	Comments		
120130	2	2	WHITE	1017	Skull	2	No Text			
Image Analysis Results			No Zoom [ X ]		Zoom [ ]		1/4	1/2	3/4	Full
Image ID	Hazmat Sign Number	Total Num. of Hazmat Signs	Color(s)	UNID	Symbol	Class	Text	No hazmat signs found		
120130	2	2	WHITE	N/A	N/A	N/A	No Text	[ ]		

Fig. G.4.: Example of Completed Image Recording Form for Figure G.6 (left).



VITA

## VITA

Albert Parra Pozo was born in Barcelona, Catalonia, Spain. He received the B.S. degree in Superior Telecommunications Engineering from the Universitat Politècnica de Catalunya (UPC) in 2010. He was a visitor scholar in the Video and Image Processing Laboratory (VIPER) at Purdue University between 2009 and 2010. He received the M.S. degree in Electrical and Computer Engineering from Purdue University in 2011. He joined the Ph.D program in Electrical and Computer Engineering at Purdue University in January 2012. He has worked as a Research Assistant in the VIPER lab under the direction of Professor Edward J. Delp since 2010, being sponsored by the U.S. Department of Homeland Security's VACCINE Center. He is a student member of the IEEE and the IEEE Signal Processing Society, and student member of the Association for Computing Machinery (ACM). He has been reviewer of the IEEE Journal on Transactions on Multimedia.

Albert Parra Pozo's publications are:

1. Chang Xu, Ye He, **Albert Parra**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Image-Based Food Volume Estimation," *Proceedings of the ACM International Conference on Multimedia*, October 2013, Barcelona, Spain.
2. Bin Zhao, **Albert Parra** and Edward J. Delp, "Mobile-Based Hazmat Sign Detection System," *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 735-738, December 2013, Austin, TX.
3. **Albert Parra**, Bin Zhao, Joonsoo Kim and Edward J. Delp, "Recognition, Segmentation and Retrieval of Gang Graffiti Images on a Mobile Device," *Proceedings of the IEEE International Conference on Technologies for Homeland Security*, pp. 178-183, November 2013, Waltham, MA.
4. **Albert Parra**, Bin Zhao, Andrew Haddad, Mireille Boutin and Edward J. Delp, "Hazardous Material Sign Detection and Recognition," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2640-2644, September 2013, Melbourne, Australia.
5. Mark Q. Shaw, **Albert Parra**, Jan P. Allebach, "Improved Video Compression using Perceptual Modeling," *Proceedings of the IS&T Color and Imaging Conference*, pp. 9-14, November 2012, Los Angeles, CA.
6. Mark Q. Shaw, **Albert Parra**, Jan P. Allebach. Techniques for Video Compression. U.S. Patent Application. PCT/US12/48514, filed July 2012. Patent Pending.
7. **Albert Parra**, Mireille Boutin and Edward J. Delp, "Location-Aware Gang Graffiti Acquisition and Browsing on a Mobile Device," *Proceedings of the IS&T/SPIE Electronic Imaging on Multimedia on Mobile Devices*, pp. 830402-1-13, January 2012, San Francisco, CA.
8. **Albert Parra**, Andrew W. Haddad, Mireille Boutin, Edward J. Delp, "A Hand-Held Multimedia Translation and Interpretation System for Diet Management,"

*Proceedings of the IEEE International Workshop on Multimedia Services and Technologies for E-health in conjunction with the International Conference on Multimedia and Expo (ICME)*, pp. 1-6, July 2011, Barcelona, Spain.

9. **Albert Parra**, Andrew W. Haddad, Mireille Boutin, Edward J. Delp, “A Method for Translating Printed Documents Using a Hand-Held Device,” *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, July 2011, Barcelona, Spain.
10. **Albert Parra**, Mireille Boutin, Edward J. Delp, “iPod-Based System for the Automatic Translation and Interpretation of Spanish Language Menus,” *Demonstration in Light-Weight Image Processing on Cellular Phones and PDAs, Show and Tell Demonstrations of the 35th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2010, Dallas, TX.