

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Chang Xu

Entitled

VOLUME ESTIMATION AND IMAGE QUALITY ASSESSMENT WITH APPLICATION IN
DIETARY ASSESSMENT AND EVALUATION

For the degree of Doctor of Philosophy



Is approved by the final examining committee:

Edward J. Delp

Carol J. Boushey

Zygmunt Pizlo

Michael D. Zoltowski

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

Edward J. Delp

Approved by Major Professor(s): _____

Approved by: Michael R. Melloch

05/02/2014

Head of the

Graduate Program

Date

VOLUME ESTIMATION AND IMAGE QUALITY ASSESSMENT
WITH APPLICATION IN DIETARY ASSESSMENT AND EVALUATION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Chang Xu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2014

Purdue University

West Lafayette, Indiana

To my parents, Fengling Li and Zhenshan Xu
For their endless support and love.

ACKNOWLEDGMENTS

Upon the completion of this dissertation, I would like to express my deepest gratitude to all those people who have made this dissertation possible.

First and foremost, I would like to thank my advisor Professor Edward J. Delp for given me the opportunity of joining his research lab and working under his supervise. During my years with him at Video and Image Processing Laboratory (VIPER), he taught me a lot about research and team work and I especially thank him for his help on my writings and for carefully reading and commenting on countless revisions of all my manuscripts. I also thank him not only for his consistent encouragement, guidance, and trust, his diverse knowledge and insight has helped me to broad my area.

I would also like to thank Professor Carol J. Boushey. I appreciate her help me on the knowledge outside my field and engagement with all of the small details of my research project. She provided me a valuable lesson on how to work with people outside my area and how to exchange ideas from different fields to obtain a better success.

I thank my advisory committee members: Professor Zygmunt Pizlo, and Professor Michael Zoltowski for their valuable suggestions, questions, and support. I would like to also thank Purdue University, the Graduate School, and the School of Electrical and Computer Engineering for taking me into the Doctoral program.

I would like to thank the National Institutes of Health for funding this research and their various forms of support in this dissertation.

I also thank all of VIPER lab members who benefit me with their wisdom, encouragement, and friendship. It has been a great pleasure to work with them. Particularly, I would like to acknowledge Dr. Nitin Khanna, who is always there to help and give a lot of valuable advises. Thanks to Ye He, my partner on our project and a valuable friend. And thanks to the rest of my wonderful current and former colleagues, who have all given their academic and social support: Ziad Ahmad, Dr. Marc Bosch Ruiz, Soonam Lee, Dr. Kevin

Lorenz, Dr. Ying Chen Lou, Neeraj Gadgil, Joonsoo Kim, Deen King-Smith, Dr. Aravind K. Mikkilineni, Dr. Ka Ki Ng, Albert Parra Pozo, Dr. Satyam Srivastava, Khalid Tahboub, Dr. Meilin Yang, Bin Zhao, and Dr. Fengqing Zhu.

I thank to the entire TADA team: Dr. Martin Okos, Dr. Heather Eicher-Miller, Dr. Nitin Khanna, Ziad Ahmad, Marc Bosch, Junghoon Chae, Shivangi Kelkar, SungYe Kim, Karl Otsmo, TusaRebecca Schap, Bethany Six, Scott Stella, Insoo Woo, Ye He, and Fengqing (Maggie) Zhu. Thanks to our great partners in Australia: Professor Deborah Kerr, Katherine Kerr, Greg Kerr, and Professor Mark Pickering.

None of this would have been possible without the love and patience of my family. My sincere thanks to my parent for their love, concern, support and strength ever since I was born. Finally, I am grateful to Chun Wang, for his love, encouragement, understanding and support to complete this work.

This work was sponsored by grants from the National Institutes of Health under grants NIDDK 1R01DK073711-01A1, NCI 1U01CA130784-01, and NIEH/NIH 2R01ES012459-06.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	x
ABBREVIATIONS	xiv
ABSTRACT	xvi
1 INTRODUCTION	1
1.1 Traditional Methods for Dietary Assessment	1
1.2 Technology Assistant Dietary Assessment System	2
1.3 Segmentation and Identification in the TADA System	6
1.4 Traditional Methods for 3D Volume Reconstruction	7
1.5 Contributions Of This Thesis	7
1.6 Publications Resulting From This Work	9
2 SINGLE VIEW AND MULTI-VIEW VOLUME ESTIMATION	12
2.1 Introduction	12
2.2 Fundamental Concepts	15
2.2.1 2D Projective Geometry and Transformation	15
2.2.2 Pinhole Camera Model	16
2.2.3 Camera Calibration	17
2.2.4 3D Volume Reconstruction	19
2.3 Template-Based Volume Estimation Using A Single View	23
2.4 Model-Based Food Volume Estimation using 3D Pose Registration for A Single View	26
2.4.1 Prism Model Volume Estimation	31
2.4.2 3D Model Generation	33
2.4.3 Pose Registration	35

	Page
2.4.4 Volume Model Finalization	37
2.5 Stereo Vision/Two View	38
2.6 Multi-View/Video Reconstruction	42
2.6.1 Shape From Silhouettes	42
2.7 Experimental Results and Conclusion	46
2.7.1 Model-Based vs. Template-Based	46
2.7.2 Single-View vs. Multi-View	52
2.7.3 Single-View Volume Estimation on Free-Living Images	54
2.7.4 Conclusion	58
3 SEGMENTATION REFINEMENT AND VOLUME ESTIMATION	65
3.1 Volume Estimation In the TADA System	65
3.2 Segmentation Refinement with User Feedback	66
3.3 Food Density and Nutrition	73
3.4 Experimental Results and Conclusion	77
4 IMAGE QUALITY MEASURES AND COLOR CORRECTION	81
4.1 Overview of Image Quality Measures	81
4.2 Fiducial Marker Detection	83
4.2.1 Candidate Region Detection	85
4.2.2 Internal Corners Estimation Using Quadrangles	89
4.3 Blur Detection	94
4.3.1 Related Work	98
4.3.2 Our Proposed Method	101
4.4 Experimental Results	102
4.5 Overview Of Color Correction	103
4.6 Scene Illumination Detection Using Color Mapping	106
4.6.1 Color Space Models	107
4.7 Experimental Results	110
5 WHITE BALANCE ON MOBILE DEVICES	115

	Page
5.1 Introduction	115
5.2 Review Of White Balancing Techniques	117
5.3 White Balancing With User Input	122
5.3.1 Model Based Color Correction	122
5.3.2 White Synthesis From Arbitrary Colors	126
5.3.3 Deployment on Mobile Imaging Systems	128
5.4 Experimental Results	129
6 SUMMARY AND FUTURE WORK	133
6.1 Conclusions	133
6.2 Future Work	135
6.3 Publications Resulting From This Work	138
LIST OF REFERENCES	140
VITA	152

LIST OF TABLES

Table	Page
2.1 Comparison of a template-based method and our model-based method for three plastic food items (banana, bagel, and orange juice) and one real food - Rice Krispy Treat. The mean and standard deviation are shown as $\mu(\sigma)$	47
2.2 Estimated weight for 19 foods items using the estimated volume and apparent density compared with the ground truth weight. The mean and standard deviation of each value is also shown. (n = number of food images that contains a particular food item)	48
2.3 Estimated weight for 19 foods items using the estimated volume and apparent density compared with the ground truth weight. The mean and standard deviation of each value is also shown. (n = number of food images that contains a particular food item)	49
2.4 Comparison of our multi-view volume estimation method in [59] and our single-view volume estimation methods for three plastic food items (banana, bagel, and orange juice) and one real food - Rice Krispy Treat. The estimation error is shown in ().	54
2.5 Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)	59
2.6 Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)	60
2.7 Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)	61
2.8 Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)	62

Table	Page
2.9 Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)	63
2.10 Percentage weight error from different volume estimation models	64
3.1 Estimated volume for six plastic foods items using user feedback	80
4.1 Mean RGB channels errors (Δ) between the reference image and transformed images	112

LIST OF FIGURES

Figure	Page
1.1 The architecture of the TADA system.	3
1.2 An example of the color fiducial marker used in the TADA system.	5
1.3 Overview of approaches used for image based 3D reconstruction.	8
2.1 An example of a food image with the fiducial marker in the scene. Each food item is segmented and identified using the TADA system.	13
2.2 An illustration of 2D transformations.	17
2.3 Imaging geometry of the pinhole camera.	18
2.4 The stereo camera calibration process.	21
2.5 An example of three world points M seen in three images.	22
2.6 Guide colors along with prompts in the mpFR used to assist the user in taking an image at preferred angles.	24
2.7 Food volume estimation using food specific shape templates (from [27]). . .	24
2.8 Volume estimation using prism shape template approximation; (a) the original image, (b) contour of the food item (scrambled eggs) is segmented (c) the contour is used with the prismatic approximation model (d) the reconstructed volume model of the food item (from [26]).	27
2.9 Shape dictionary: food items with their corresponding 3D shapes.	28
2.10 Examples of the same food type (carrot) with different heights.	32
2.11 The percentage weight estimation of broccoli with respect to the height. . .	33
2.12 Our single-view volume estimation system.	34
2.13 The geometric relationships between the camera center and the food object.	37
2.14 An example of projecting a 3D banana model to a 2D image plane with two pose angles. The elevation angle ϕ is varied from 0° to 180° , and the azimuth angle θ is varied from 30° to 60°	37
2.15 Epipolar geometry for toe-in views.	39

Figure	Page
2.16 An illustration of correspondence between a point on view 1 with a line on view two.	40
2.17 An example of feature matching utilizing the epipolar constraint. The correspondent points in left and right images are connected by the green line. (a) is showing all the correspondent points using SIFT matching, (b) shows the point matches fulfill the epipolar constraint, (c) shows the point matches fail the epipolar constraint.	41
2.18 Training angles for a food object and the reconstructed 3D model.	44
2.19 An example of voxel based marching intersection.	45
2.20 Examples of meal images from the 24-hr controlled user study.	50
2.21 The comparison result of our new model based method with previous template based method.	51
2.22 Food weight error using automated volume analysis by food type from images taken by 15 adolescents (11-18 y) during meals over a 24-hr period. The error bars indicate the standard deviation.	52
2.23 Examples of meal images from the freeliving user study [49,90].	55
2.24 Examples of 56 food items from the freeliving user study. (From left to right and top to bottom: apple, bagel, banana, broccoli, carrots, celery, chicken wrap, chocolate chip, clementine, cream cheese, ding dong, doritos, english muffin, frozen meal meatloaf, frozen meal turkey, fruit cocktail, garlic toast, goldfish, granola bar, grapes, ham sandwich, ice cream, jelly, lasagna, margarine, mashed potato, mayonnaise, milk, muffin, mustard, no fat dressing, noodle soup, orange, orange juice, pancake, peanut butter, pears, peas, pizza, potato chips, pretzel, pudding, ranch dressing, rice krispy bar, salad mix, saltines, sausage, snickerdoodle, snickers, strawberry, string cheese, syrup, watermelon, wheat bread, wheaties, yogurt.	56
2.25 Average percentage of weight error using automated volume analysis with FNDDS density and Okos density.	57
3.1 The architecture of the TADA system.	65
3.2 Block diagram of the proposed volume estimation in the TADA system. . .	67
3.3 Screen shots of the mpFR app during the review process [9].	68
3.4 A curve C can be represented by a level set function ϕ by propagating in normal direction (from [95]).	70
3.5 An example of user feedback and segmentation refinement.	72

Figure	Page
3.6 Examples of the failure cases using active contours.	73
3.7 Example of true, apparent and bulk density of the same food (puffed cornflour pellets) (from [31]).	75
3.8 The TADA food databases.	76
3.9 An example of the volume, weight, and density information available in I-TADA for a particular image.	78
3.10 An example of all the nutrient values for a particular portion of a food item in FNDDS database.	79
4.1 Our proposed quality measure and image enhancement system.	82
4.2 The color fiducial marker used in the TADA system.	83
4.3 Overview of our proposed checkerboard detection method.	85
4.4 Checkerboard candidate region detection.	86
4.5 An demonstration of connected components labeling (from [113]).	87
4.6 Examples of checkerboard candidate region detection.	88
4.7 Internal corners estimation using quadrangles.	89
4.8 Examples of internal corners estimation using quadrangles.	90
4.9 An example of binarizing the image with a fixed threshold and a local threshold. (a) - original checkerboard image, (b) - binary image with local thresholding, and (c) - binary image with fixed value thresholding.	91
4.10 An example of the Douglas-Peucker (DP) method.	92
4.11 An food image with motion and Gaussian blur.	94
4.12 (a) - Motion blurred image with $L = 10$, $\phi = 45^\circ$; (c) - motion blurred image with $L = 30$, $\phi = 45^\circ$; (b) and (d) - frequency response of (a) and (c) from [116].	95
4.13 The effect radius of the blur kernel on matching accuracy [117]	96
4.14 Effect of image blur on SIFT features and the proposed blur metric.	98
4.15 An illustration of the measurement of the width around an edge pixel.	99
4.16 The linear model from RGB to XYZ.	108
4.17 The color model from RGB to LAB.	109
4.18 The image with reference light condition (D65).	111

Figure	Page
4.19 Comparison of the five color correction methods - part1	113
4.20 Comparison of the five color correction methods - part2	114
5.1 The image processing inside a digital camera.	116
5.2 A same scene with different color temperature.	118
5.3 (a) Original image and results obtained by applying (b) GWM, (c) perfect reflector, (d) fuzzy rules [144]	121
5.4 A block diagram of the device to device transformation CCMX (Φ).	123
5.5 A schematic diagram of an imaging unit.	123
5.6 A printed sheet of colored patches used to construct the camera models.	125
5.7 An example of the interface for user input. Image on the left is captured and displayed alongside a grid of common colors (right) for the user to match.	127
5.8 The flow chat of the user interface.	129
5.9 Result of correcting an image with our method assuming known illumination. The columns represent the input image (left), the white corrected image (center), and the image taken under reference conditions (right).	130
5.10 An illustration of user-specified color matching for white synthesis. The reference image on the right is only provided for comparison.	131
5.11 Two examples of images color corrected with our method and user input. The columns represent (i) input image, (ii) image acquired with automatic settings, (iii) input image corrected with our method, and (iv) image under reference illumination.	132
6.1 A future 3D reconstruction system.	137

ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
API	Application Programming Interface
AR	Augmented Reality
CCD	Charge Couple Device
CPBD	Cumulative Probability of Blur Detection
DLT	Direct Linear Transformation
DLW	Doubly Labeled Water
DOF	Degree of Freedom
DSC	Digital Still Color Cameras
FFQ	Food Frequency Questionnaire
FNDDS	Food and Nutrient Database for Dietary Studies
FOV	Field of View
FR	Food Record
GUI	Graphical User Interface
HMM	Hidden Markov Models
HSV	Human Visual System
IQ	Image Quality
LUT	Look-up Tables
mpFR	Mobile Telephone Food Record
RANSAC	RANdom SAmple Consensus
ROI	Region of Interest
SDK	Software Development Kit
SfM	Shape from Motion

SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
SDK	Software Development Kit
TADA	Technology Assisted Dietary Assessment

ABSTRACT

Xu, Chang. Ph.D., Purdue University, May 2014. Volume Estimation and Image Quality Assessment with Application in Dietary Assessment and Evaluation. Major Professor: Edward J. Delp.

Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields.

Our team at Purdue University and the University of Hawaii Cancer Center have been developing an image analysis system to automatically estimate energy and nutrient intake from food images acquired by mobile devices for the past six years with support from the National Institutes of Health. This system known as the Technology Assisted Dietary Assessment System (TADA) has developed a mobile telephone food record (mpFR) application and deployed it on iOS and Android devices. The TADA system can automatically identify and quantify foods and beverages consumed based on analyzing meal images captured with a mobile device. After food items are segmented and identified, accurately reconstructing the volume of the food in the image is important for determining the nutrient content of the food. Once food portion size is estimated using volume and density information of the food items, the energy and nutrient information of the meal are obtained.

In this thesis, we investigate the improvement of several aspects of the TADA system. We describe methods for food volume estimation, image quality assessment, and color correction. We propose a novel food portion size estimation method with the use of the 3D reconstruction and pose estimation methods based on a single image or multiple images. The single-view method estimates food volume by using prior information - segmentation and food labels generated from food identification methods in TADA. A 3D object model is reconstructed for each food item on the meal image using the prior shape information. Then, we determine the pose of the 3D model by projecting it onto the meal

image. Subsequently, the food volume is estimated by matching the projection image of the transformed 3D model with the segment of the food item. We also implemented a multi-view shape recovery method using “Shape from Silhouettes” methods. We evaluated our single-view volume estimation models using food datasets from a 24 hour controlled eating occasion study and a free-living study with 56 food types. Apart from food volume estimation, we also investigate how to refine the volume estimate based on user adjustment from TADA/mpFR system. With the user feedback, corrected labels and hand segments, we obtained better food segmentation using active contours and consequentially improve the volume estimation.

Food identification is a difficult problem since foods can dramatically vary in appearance. Such variations may arise not only from non-rigid deformations and intra-class variability in shape, texture, color and other visual properties, but also from changes in illumination and viewpoint. Therefore, it is very important to assist the user in requiring a good quality image by providing immediate feedback about the image quality. Low complexity image quality measures which are deployed on mpFR are also investigated. Furthermore, to address the color consistency problem, three color correction methods are proposed for illumination quality assessment.

1. INTRODUCTION

The last several years have seen a growing interest in preventing and/or managing chronic diseases related to diet, including obesity, cancer, diabetes, and heart disease. Among ten major causes of death in the US, six of them are related to diet. Accurate food intake and dietary record provides helpful information in preventing the occurrences of chronic diseases. Traditional dietary assessment methods [1, 2] have been developed to measure dietary intake. Unfortunately, fine grain assessment and proactive health management of diet does not currently exist. The accuracy of these methods is debatable, especially in adolescents, because the consistent under-reporting is found in these dietary measurement methods. Therefore, accurate dietary assessment is still considered to be an open research problem, and dietary assessment methods that are less burdensome and less time-consuming are highly demanded in the nutrition and health research community. A convenient and portable dietary assessment method which could lead to tailored dietary goal setting by adults in tandem with their healthcare providers would be a transformation tool in assisting healthcare providers in this effort. The rapidly developed mobile technologies (e.g. a mobile telephone or PDA-like device) in recent years have emerged in healthcare field to provide unique mechanisms to monitor eating habits of users and improve the accuracy of dietary and food intake assessment [3–12].

1.1 Traditional Methods for Dietary Assessment

Traditional dietary assessment is comprised of written and orally reported methods that are time consuming and tedious, often require a nutrition professional to complete, and are not widely acceptable or feasible for everyday monitoring [13]. The error inherent in these methods is a barrier in developing care plans that actually match what people eat. Error is introduced in several ways: underreporting, estimated to be as high as 50% [14]; the

human inability to estimate food portion size accurately [3]; and as such, consumed energy and nutrients are incorrectly estimated. Existing technology approaches are either digital versions of traditional recording methods [15] [16] [17] or just provide ways to record food images [18] [19]. The steps needed to identify the foods and determine the portion and serving sizes were described as limited and lacking flexibility [20].

Real-time personal health monitoring is becoming accessible due to advances in technology. There has been an explosion of health-related applications developed on mobile platforms. The use of meal images using mobile telephones with embedded cameras designed to identify foods and beverages addresses the barriers indicated by individuals engaged in diabetes self-management [21, 22].

1.2 Technology Assistant Dietary Assessment System

Our team at Purdue University and the University of Hawaii Cancer Center have been developing an image analysis system to automatically estimate energy and nutrient intake from food images acquired by mobile devices for the past six years with support from the National Institutes of Health. This system is known as the Technology Assisted Dietary Assessment System (TADA) (www.tadaproject.org). Our team has developed a mobile telephone food record (mpFR) application and deployed it on iOS and Android devices [3–9, 12, 23–25]. A mobile telephone with a built-in camera is used to acquire meal images at each eating occasion to record dietary intake. Images of food taken before and after eating allow for automatic labeling and volume estimation of consumed food using image analysis methods [8, 25–27].

The TADA system is currently being used by dietitians and nutritionists in various departments at Purdue University, the University of Hawaii Cancer Center, and the Curtin University of Technology in Australia. This system has been tested with more than 300 users and we have collected more than 20,000 eating occasion images containing foods and beverages. Figure 1.1 shows the TADA system.

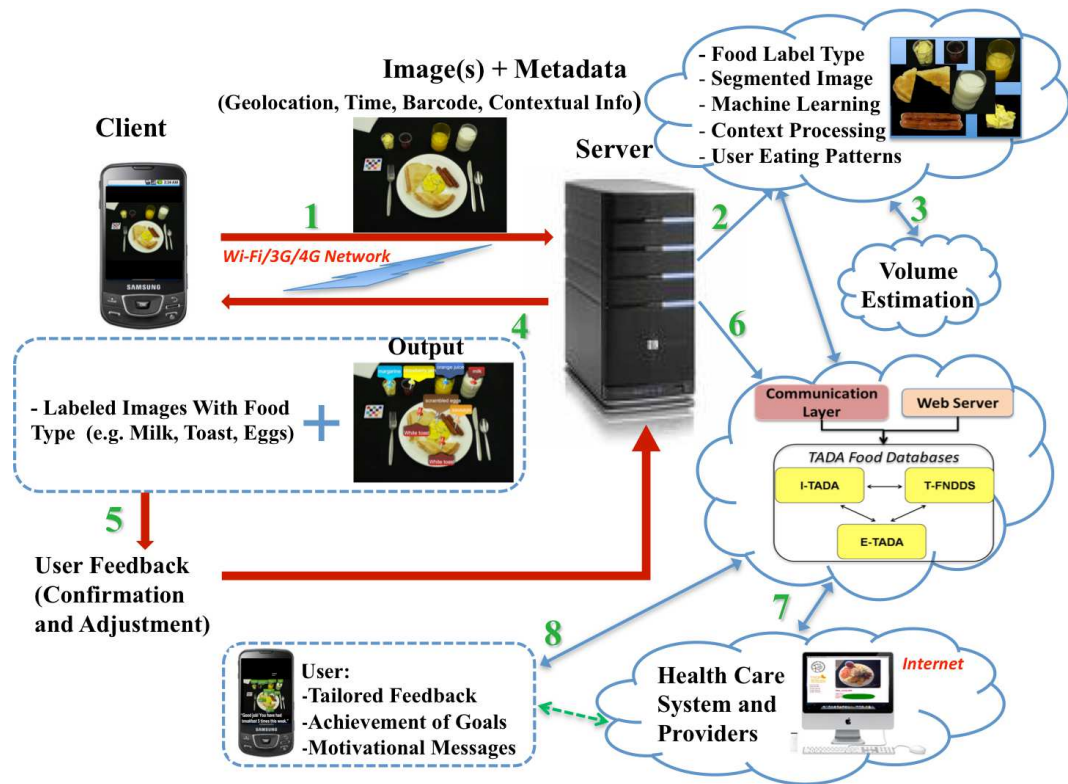


Fig. 1.1. The architecture of the TADA system.

The TADA system process starts with the user acquiring food images with the mpFR and sending the images and various types of contextual data (e.g. date, time, and geolocation) to the server (step 1). Automatic analysis (e.g. image segmentation, food identification, context processing and weight estimation) is done on the server (step 2 and step 3). The labeled images with food types are sent back to the user to confirm and/or adjust the image analysis results, if necessary (step 4 and step 5). Based on the feedback information from users, the server refines image analysis results of food identification and weight estimation in previous steps. An extended version of the USDA Food and Nutrient Database for Dietary Studies (FNDDS) [5, 28, 29] is used for estimating the energy and nutrient information given a food label and weight (step 6). The database contains the most common foods in the US, their weights, nutrient values, and food densities [30,31]. Finally,

these results are sent to the user and the healthcare community for presentation of dietary recommendations and planning (step 7 and 8).

Since we are interested in knowing how much food is consumed we need to have a 3D calibrated imaging system. In the current version of the TADA system, a user takes only one image of the food and 3D models are used to construct the 3D object. Whatever approach is used, we still need a calibrated imaging system that is calibrated both spatially and with respect to the colors represented in the scene. We have chosen to use a checkerboard-like design as a particular type of “fiducial marker” for our calibration information. The fiducial marker is included in every image to provide a reference for the scale and pose of the objects in the scene. After exploring and testing several designs, we decided to use a compact checkerboard pattern. This is described in more detail in [4, 32]. The current version of the checkerboard is a small ($7 \times 6 \text{ cm}^2$) asymmetric color checkerboard (see Figure 1.2) with 5×4 square dimensions, rigid mounting on foam board, and a high contrast pattern. The colors of the checkerboard are chosen as uniform in the color space as possible. Consequentially, this checkerboard is used to calibrate both geometry and the color representation in the meal scene. Automatic detection of the checkerboard and color correction using the checkerboard will be introduced in this thesis in Section 4.2.

Spatial calibration could also be accomplished by having a known object in the scene, such as a coin, credit card, or a plate [33]. However, due to the variations in the size of some of these objects, they may not be reliable. Several studies we conducted in the Department of Nutrition Science at Purdue University indicate our credit-card-sized checkerboard is convenient to carry around and incorporate into the participants’ lifestyle [34].

Color features play a crucial role in food identification and many food items have closely related colors [5, 25, 35] while there is a wide range of illumination conditions for different eating occasions. Thus, color correction plays a crucial role in dietary assessment methods [8, 36, 37]. Most of these methods use some fiducial markers for estimating unknown illumination conditions and subsequently color correction methods for mapping the test image back to the reference illumination conditions. Similarly, the fiducial marker

plays a key role in 3D reconstruction from 2D images for estimating the volumes of different food items [27, 36].

One of our goals is to detect the “quality” of the image as the user acquires it with the mpFR. For example, if we detect a poor quality image we can then ask the user to re-take the image. Apart from the general notion of image quality such as sharpness, in the context of dietary assessment, good quality images need to satisfy specific requirements such as the presence of a fiducial marker and appropriate camera angle. Based on our user studies the most significant reasons for poor quality images are: non-detectable fiducial marker (forgetting to use the fiducial marker or overexposed/blurred images), spectral reflections, shadows, insufficient illumination, and blur. Doing image quality assessment on the TADA backend server and sending back the response to the user is not feasible in most circumstances due to associated network and computation delays. Mobile devices have limited computation power and the image quality assessment needs to be performed before the image is sent to the server for further processing. Therefore, these methods must use low computation and memory resources. The methods described in this thesis have been deployed on the mpFR without adding any perceptible delay in the “image capture step.”



Fig. 1.2. An example of the color fiducial marker used in the TADA system.

This thesis will first introduces several volume estimation methods for food portion size measurement. After the food images are segmented and classified (the food item is identified) the volume of the food is estimated. Several single-view and multi-view volume estimation methods are described in this thesis and some traditional methods for 3D reconstruction are introduced in the following sections.

Then, we will describe image color quality assessment methods using mobile devices and post processing for color correction. While some of the correction steps can be done on the mobile device, computationally intensive steps including color correction are done on the server. We have found that a fast illumination check can be implemented on the mobile device without adding any perceptible delay in the image capture process.

1.3 Segmentation and Identification in the TADA System

Full utilization of the side information provided by the use of a checkerboard and other contextual information is very important for dealing with the challenges involved in food classification and volume estimation from a single image. Previous work on the TADA system has investigated methods for food classification and volume estimation. Food identification is a difficult problem since foods can dramatically vary in appearance [4, 5, 11, 12, 25, 35, 38, 39]. Such variations can be caused by changes in food arrangement, inter-class variability, background clutter, cooking/food preparation variations, and changes in illumination and viewpoint. Some of these problems have been addressed by the TADA research team [11, 12, 40, 41].

After image acquisition, image segmentation techniques are used to locate the object boundaries for food items. The segmented regions are then used for food labeling/identification and volume estimation. Earlier work in the TADA system has developed a joint iterative segmentation and classification system, where the classifier’s feedback (i.e. class label and confidence score) is used to obtain a final segmentation. We call this approach Multiple Hypothesis Segmentation and Classification (MHSC) [4, 8, 25]. Salient regions are first detected (plates, bowls and glasses in the image). After salient region detection, multi-scale segmentation [4, 8, 25] is done. As a result of this operation, we obtain a pool of segmented regions for each image. Each segmented region is classified using a multichannel feature classifier. We use a combination of global and local features including color feature, texture feature, and local feature [5, 25, 29, 42]. The final segmentation is obtained by an iterative process for joint segmentation and classification [25, 29, 42]. At each

iteration, each salient region is partitioned into a different number of segmented regions that are automatically classified. Every pixel is assigned with the class label that has the highest cumulative confidence score from the classifier up to the current iteration.

1.4 Traditional Methods for 3D Volume Reconstruction

3D reconstruction, also referred to as *3D modeling* is the process of recovering the shape and structure information of scenes or objects. Most of the recent image based 3D reconstruction techniques can be mainly classified into two categories according to the data acquisition devices: active sensors or passive method. The active methods rely on active sensors (e.g. moving light sources, coded or structured light, time-of-flight lasers to microwaves or ultrasound) and directly interfere with the reconstructed objects [43]. A specific example of the active method is the Microsoft Kinect sensor [43] which uses an infrared laser projector and scanner to provide full-body 3D motion capture. The active method can provide a highly accurate and complete 3D representation of small and medium scale objects.

Passive methods of 3D reconstruction only use an image sensor to obtain the radiance reflected from the object's surface and the output of the image sensor. A set of images (one, two, three or more) or a video clip, will need further processing to infer the 3D structure. Figure 1.3 illustrates an overview of various types of approaches designed for 3D reconstruction: single view reconstruction and multi-view/video reconstruction [44–48]. However, many of these methods are targeted for large scale geometric scenes where the details of the small scale object reconstruction are not considered.

1.5 Contributions Of This Thesis

In this thesis, we focus on developing methods for image quality assessment, color correction and 3D volume reconstruction for use in the TADA system. The main contributions in this thesis are as follows:

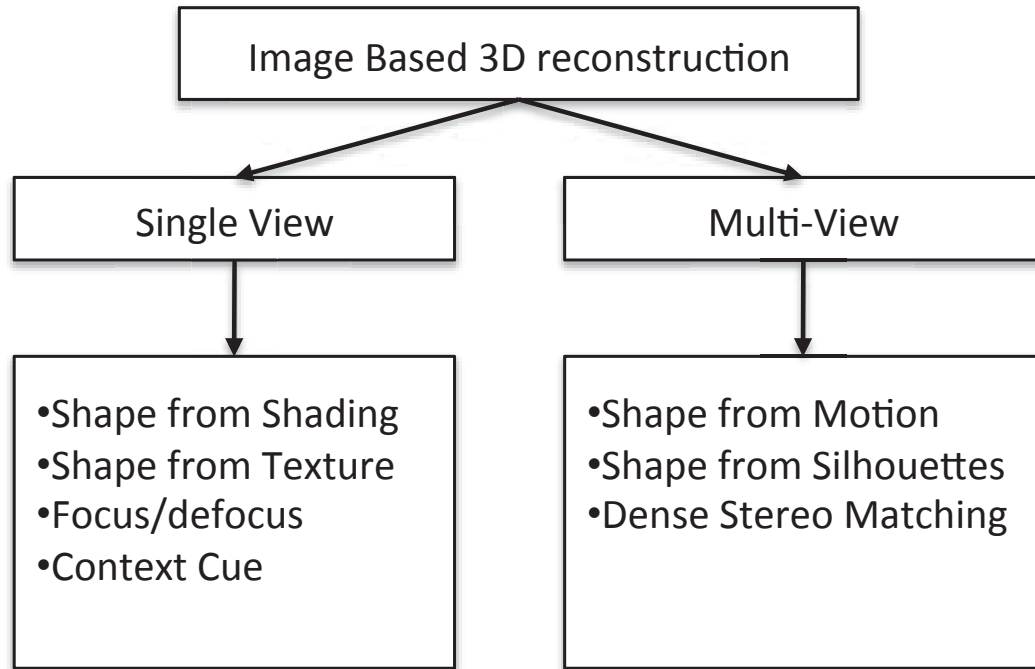


Fig. 1.3. Overview of approaches used for image based 3D reconstruction.

- We propose a novel food portion size estimation method using a single image. The single view volume estimate is implemented using a model-driven system for creating 3D reconstructions of specific food items. A pre-built or defined 3D model of a food item is projected back to the image plane. Subsequently, the portion size and degrees-of-freedom (DOFs) for the final pose is estimated by an image similarity measure.
- We describe a multi-view volume estimation method to automatically estimate the food portion size. We demonstrate the use of stereo vision in order to improve the accuracy of food segmentation and volume estimation. Also, the multi-view shape recovery method is implemented using a combination of shape from silhouettes and shape from correspondence techniques.
- We propose an approach to automatically detect the presence of the fiducial marker. The method is based on region search, which is less sensitive to illumination changes

and noise than the corner or line based methods. We compare this method with these traditional methods. The proposed approach reduces the computational time for locating the corners, speeds up the image pre-processing, and adapts to automatic image quality assessment on mobile devices.

- We developed a low complexity blur metric by suitably modifying a well known method known as cumulative probability of blur detection (CPBD) which utilizes probability distribution (CPBD) of edge widths. The average computational runtime of the original CPBD method is reduced from 9 seconds to 1 second on a mobile device for 2048×1536 sized images.
- We propose three chromatic adaptation models that use more perceptually uniform color space models: a linear RGB to RGB transform, a nonlinear RGB to RGB transform, and a linear model in CIELAB color space. From experimental results, our proposed methods offer better color consistency in various illumination tests than the other well known techniques.
- We devised an intuitive method for a user to specify veridical color descriptors for color patches in a scene. This is facilitated by the availability of graphical interfaces on many mobile devices such as smartphones. A scheme to synthesize the white point descriptor using a weighted combination of the colors provided by the user is also introduced.

1.6 Publications Resulting From This Work

Journal Papers

1. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Volume Estimation with Application in Dietary Assessment," *IEEE Transactions on Information Technology in Biomedicine*, in preparation.

2. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Image Classification with Contextual Dietary Information," *IEEE Transactions on Multimedia*, in preparation.
3. **Chang Xu**, Satyam Srivastava, and Edward J. Delp, "User Assisted White Synthesis on Mobile Device," *Journal of Imaging Science and Technology*, in preparation.

Conference Papers

1. **Chang Xu**, Ye He, Albert Parra Pozo, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Image-Based Food Volume Estimation," *Proceedings of ACM International Conference on Multimedia*, Barcelona, Spain, October 2013, pp.75-80.
2. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Model-based food volume estimation using 3D pose," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2534-2538.
3. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Context based food image analysis," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2748-2752.
4. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of the IEEE International Conference on Multimedia and Expo*, San Jose, CA, July 2013, pp.1-6.
5. **Chang Xu**, Fengqing Zhu, Nitin Khanna, Carol J. Boushey, Edward 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. 82960Q110, San Francisco Airport, California, January, 2012.
6. **Chang Xu**, Nitin Khanna, Carol J. Boushey, Edward J. Delp, "Low Complexity Image Quality Measures for Dietary Assessment Using Mobile Devices," *Proceedings*

of the IEEE International Symposium on Multimedia, Dana Point, California, December, 2012, pp. 351–356.

7. Satyam Srivastava, **Chang Xu**, Edward J. Delp, “White synthesis with user input for color balancing on mobile camera system,” *Proceedings of the IS&T/SPIE Conference on Multimedia on Mobile Devices 2012*, Vol. 8304, pp. 83 040F110, San Francisco Airport, California, January, 2012.

2. SINGLE VIEW AND MULTI-VIEW VOLUME ESTIMATION

2.1 Introduction

Accurately measuring dietary intake is considered to be an open research problem in the nutrition and health fields. Traditional dietary assessment is composed of written and orally reported methods that are time-consuming and tedious, which makes them not widely acceptable or feasible for everyday monitoring. Besides the TADA system [8, 24, 34, 39, 49], a number of other dietary assessment systems utilizing images/videos of eating occasions have been proposed [50], [51]. These systems provide unique mechanisms for improving the accuracy and reliability of dietary assessment. Most of these approaches involve manual or automatic food identification. Portion size of the food items is then estimated through volume estimation. Once food portion size is estimated, the energy and nutrient information of food eaten can be obtained using the methods discussed in Section 3.3.

Portion size estimation is extremely difficult since many foods have large variations in shape and appearance due to eating or food preparation conditions [34, 52]. Most image-based dietary assessment systems use a single image [26, 53], multiple images [54], video [55], or 3D rangefinding [56]. For example, “DietCam” [51] is a mobile application where food intake assessment is based on images acquired from multiple views. It requires users to acquire three images separated by about 120° which increases user burden.

A mobile structured light system (SLS) to measure daily food intake is being developed by Sheng et al. [56]. A laser device which attaches to a mobile telephone is used to capture depth images of the food objects. This system seems burdensome and requires extra hardware which is not suitable for daily use. Jia et al. [33] developed a wearable camera device to collect eating occasion information. It makes use of a known-size plate as the geometric reference. They define several simple geometric shapes to model food shapes and manual adjustment is required. Chen et al. [53] proposed a 3D/2D model-based image registration

method for quantitative food intake assessment. The method utilizes a global contour to solve the position, orientation and scale of the user-selected 3D shape model. It obtains reliable food volume estimation for many simple food items. However, it does not have a solution for foods that do not fit a simple model (e.g. banana, pear) or complex structured food items (e.g. fries, salad). In addition, it only uses the outline of the object and discards the internal structure (lines, curves, and ridges) of the segments, which could lead to low accuracy in pose registration.

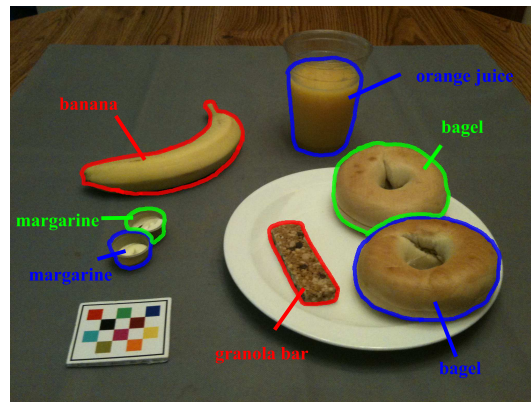


Fig. 2.1. An example of a food image with the fiducial marker in the scene. Each food item is segmented and identified using the TADA system.

Food volume estimation is a critical step in accurate dietary intake estimation in that it is used to estimate the nutrients in the food items. In the early work in the TADA system, a shape template method for 3D reconstruction of some particular types of food objects was used [27]. It utilized the feature/corner points from the segmented image to compute the geometric information of the shape template. However, this method is highly dependent on the accuracy of the segmentation mask and feature points detection. Moreover, it fails when the food item has a complex or amorphous shape. The objective of our work is to automatically estimate food portion size through 3D volume reconstruction from a single image of an eating occasion, multi-view images, or a video. We formulate food portion size estimation in this thesis as a 3D volume reconstruction problem. We also evaluate

the accuracy of our approach using the images from one of the TADA free living user studies [11, 40].

We describe novel and reliable volume estimation methods based on geometric constraints and a contextual 3D model. We first obtain a 3D graphical model of the food object from multiple training images. We then compute a segmentation mask and a food label using the TADA image segmentation and food identification techniques described in [25, 39, 41, 42, 57]. The segmentation mask provides the location of a food item and the food label indicates the food identity as shown in Figure 2.1. We assume the fiducial marker is included in every image as a geometric reference for the scale of the world coordinates and to provide color calibration information [58].

We then estimate the camera pose from the checkerboard and establish the world coordinates. We utilize several geometric constraints and the food placement regularities to solve the pose registration problem. The degrees-of-freedom of the pose for different foods are obtained based on the food identification. After the pose of a food item is determined, we are able to estimate the volume of the food through other information of the pose such as the size factors. Once the volume is estimated, the nutrient content of the food is obtained using the density for that particular category of food [30, 31] and the method is also discussed in Section 3.3.

The rest of this chapter is organized as follows: Section 2.2 presents the fundamental concepts of 3D geometry and the pinhole camera model. Section 2.3 introduces the volume reconstruction using a single view, which is implemented in a template-driven system for 3D reconstructions of specific-shaped objects. In Section 2.6, we describe a multi-view shape reconstruction method using the combination of shape from carving and shape from correspondences techniques. Experimental results are presented in Section 2.7. This work was also published in [10, 59].

2.2 Fundamental Concepts

Most 3D reconstruction techniques are based on geometric modeling of the imaging device, exploiting the mathematical and contextual relationships between the 3D world coordinates and the 2D image plane. The pinhole camera model is a good approximation of the perspective projection of a 3D world scene onto a 2D image plane [60].

2.2.1 2D Projective Geometry and Transformation

In basic geometry, a 2D point in the plane is represented by (x, y) in \mathbb{R}^2 space. However, this representation is limited when the problem incorporates points at infinity. Therefore, homogeneous coordinates [61] were introduced as a better representation than Cartesian coordinates in Euclidean geometry. The point $x = (x, y)^T$ in \mathbb{R}^2 , can be converted into a homogeneous space by adding a third coordinate of 1. And for an infinite point $x = (\infty, \infty)$, it can simply be defined as $x = (1, 1, 0)$ in the homogeneous representation. Using homogeneous coordinates, most of 2D transformation such as translation, rotation, scaling, and perspective projection can be implemented as a linear matrix operation without considering the infinite points.

2D linear transformations such as translation, rotation, scaling, and shear are categorized as Affine transformations. The matrix representation [61] is shown in Equation 2.1

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2.1)$$

where $(x, y, 1)$ and $(x', y', 1)$ are the homogeneous coordinates of the before and after transformation. It also has a block form:

$$x' = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} x, \quad (2.2)$$

where A is a 2×2 non-singular matrix. Most of 2D geometric transformations are a derivation of Affine transforms [61] and they can be represented using a 2×2 transformation matrix.

For example, the translation transformation is in the form where $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $t = [a \ b]^T$.

In the field of computer vision, projective transformation or homography is often used to model the transformation of a 2D planar object projected from two camera views [61]. The matrix representation of this is:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ b_1 & b_2 & b \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2.3)$$

The block form is:

$$x' = \begin{bmatrix} A & t \\ B^T & b \end{bmatrix} x, \quad (2.4)$$

where A is a non-singular 2×2 matrix, t is a translation vector in dimension of 2, and $B = (b_1, b_2)^T$.

Projective transformation is a generalized form of Affine transformation with a non-singular linear transformation of inhomogeneous coordinates and a translation vector [61]. That is to say, it is a non-linear transformation in Cartesian coordinates. Since it describes the transformation of projection a 2D planar from two distinct camera views, it is often used in image rectification, stereo vision, and image stitching problems [61, 62]. In Figure 2.2, the effect of three different types of 2D transformations of a small checkerboard is shown.

2.2.2 Pinhole Camera Model

In this work, we use the pinhole camera model to represent the imaging geometry [61]. In particular, Figure 2.3 shows the notation used to model a 3D point projected on a 2D

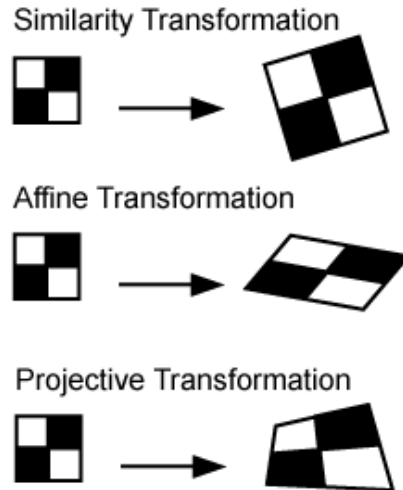


Fig. 2.2. An illustration of 2D transformations.

point by the camera. $M = (X, Y, Z)$ is a 3D point from the scene in the camera coordinates system, which is projected into $m = (u, v)$ that is a 2D point of the image plane I . C is the optical center. Note that the image plane is virtually between C and world point M for making modeling easier, with f the focal length. The camera center is denoted by C and the principle point is denoted by P , which is the camera center projected onto the image plane. The axis which crosses the principle point P and camera center C is known as the principle axis. The image plane is oriented perpendicular to the z -axis so that it intersects the axis at $z = f$, where f is known as the focal length of the camera. The relationship between the 3D point M and 2D image pixel m will be determined by the camera parameters that will be introduced in the following section.

2.2.3 Camera Calibration

Camera calibration is usually the first step in many computer vision applications. It consists of estimating the intrinsic and extrinsic parameters of the camera [61]. The intrinsic parameters describe the focal length and principal point of the camera. Methods to estimate the focal length, in general, require multiple images (usually more than five im-

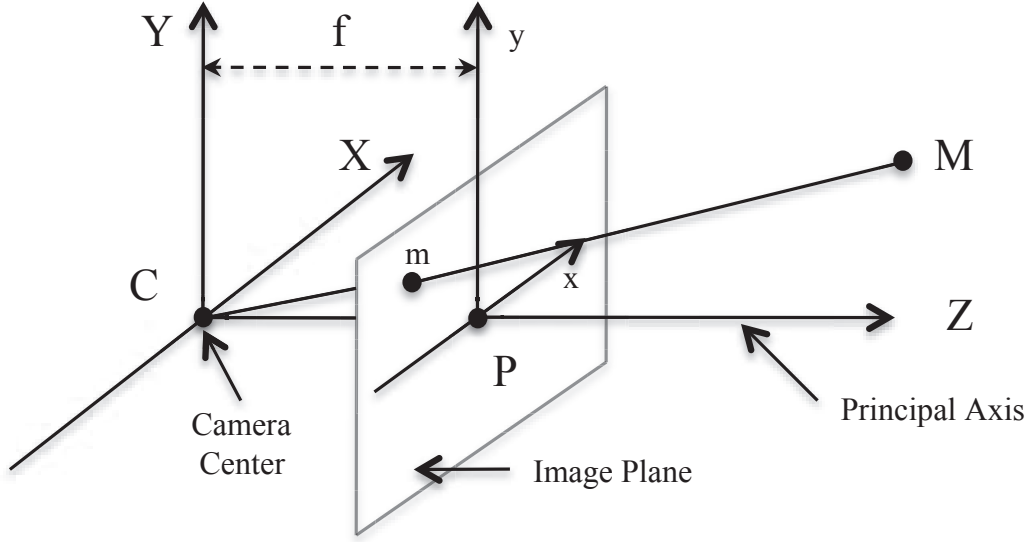


Fig. 2.3. Imaging geometry of the pinhole camera.

ages) or specific constructed grids [60], [63]. Currently, many camera manufacturers add the focal length information in the EXIF (Exchangeable Image File Format) header of the image file [64]. We assume that the principal point is located at the center of the image plane. Estimating the principle point using a small calibrated target is an ill-posed problem and may lead to a large error in 3D reconstruction.

The model that represents the camera parameters is given by [60]:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.5)$$

where (X, Y, Z) are the coordinates of a 3D point M in the world coordinate, and (u, v) are the coordinates of the 2D projection point m in pixels in image coordinates. C_x and C_y are the coordinates of the principal point (we locate it at the center of the image), and f_x and f_y are the focal lengths in the unit of pixels. r_i and t_i are the rotation and translation external parameters that relate the world coordinate system to the camera image

coordinates. As mentioned earlier, in order to establish the 2D-3D relationship from this model, we also need to estimate the extrinsic parameters. The extrinsic parameters contain the rotation matrix R , described by the rotation parameter r_i and the translation vector t , given by the parameters t_i . Rotation and translation parameters can be estimated from a checkerboard pattern present in the image. After finding the locations of all the intersection points (corners) in the checkerboard pattern in the image, the maximum likelihood inference [60] can be used to map these corner points onto the 3D world coordinates, and thus, we find the rotation and translation parameters. This is described by:

$$(R, t) = \arg \min_{R, t} \sum_{i=1}^{12} \|m_i - \hat{m}(A, R, t, M_i)\|^2, \quad (2.6)$$

where $\hat{m}(A, R, t, M_i)$ is the projection of 3D point M into the 2D image, according to Equation 2.28. Here A is the intrinsic matrix which consists of focal length and principle point.

2.2.4 3D Volume Reconstruction

3D modeling of an object from a single image is a difficult task and is still an open problem in the computer vision [43, 65, 66]. Single view 3D reconstruction is considered to be the most difficult 3D problem since a huge amount of information is lost in the 3D to 2D projection. This is an ill-posed inverse problem and requires the use of regularization techniques or side information (contextual information) [67–71]. Contextual information of the object is data that is not directly produced by the visual appearance of an object in the image, but yields semantic information that is not intuitively obtained from the image, such as shadows of the object or the object category. Contextual information is often used to simplify (or regularize [69, 70, 72]) the problem, e.g. use of the underlying information of shading [47] or texture of the object surface [73] has been proposed as contextual information. Context-based single view reconstruction highly depends on the specific properties of the object surface. Objects with homogeneous texture properties may not provide useful context in order to aid in the reconstruction. In [66], the authors describe a 3D affine recon-

struction which only uses minimal geometric information, which is typically the vanishing line of a reference plane and a vanishing point for a direction not parallel to the plane. A 3D depth estimate is derived in [74] that utilizes a supervised learning approach. However, most of the studies in this field do not discuss the reconstruction of occluded parts of the scene. Therefore, the volume information of the object cannot be retrieved.

In [65], a single view curved 3D surfaces modeling method is proposed. In this method, where the constructed surface of an object is computed by minimizing a smoothness objective function:

$$E(r) = \int_0^1 \int_0^1 \|r_{uu}\|^2 + \|r_{uv}\|^2 + \|r_{vv}\|^2 dudv. \quad (2.7)$$

Two different types of constraints will apply to the surface energy namely: terminology constraints (imaging rays are tangent to the surface), inflation constraints (the surface should pass near certain 3D point). However, the method is based on an initial contour drawn by the user and therefore is not practical for our system.

A considerable amount of work has been done in stereo vision [75, 76]. In stereo, the depth information can be retrieved by estimating the correspondent pixel in the left and right image [75]. If only sparse feature points are considered and locating their correspondent points in the second view are determined, then it is categorized as a sparse stereo matching problem. In contrast, finding the correspondent pixel in another view for each pixel in the current view is defined as a dense stereo matching problem. Stereo calibration and rectification is one of the essential step for dense stereo matching. The purpose of stereo calibration is to reduce the radius distortion in left and right images [60]. Rectification is used in order to force epipolar lines to match with the horizontal scanned lines in the images and the two image planes to be aligned coplanar. As shown in Figure 2.4.(a) is a pair of images from the left and right camera. After calibrating each camera using the technique in Section 2.2.3, the images will be distorted as shown in 2.4.(b). The rotation matrix R_l and the translation vector T_l of the left camera with respect to the world coordinate, and the rotation matrix R_r and the translation vector T_r of the right camera with respect to the world coordinate are also obtained for each calibrated image. Then, the relative position (R, T) of the right camera with respect to the left camera is determined based on Equation

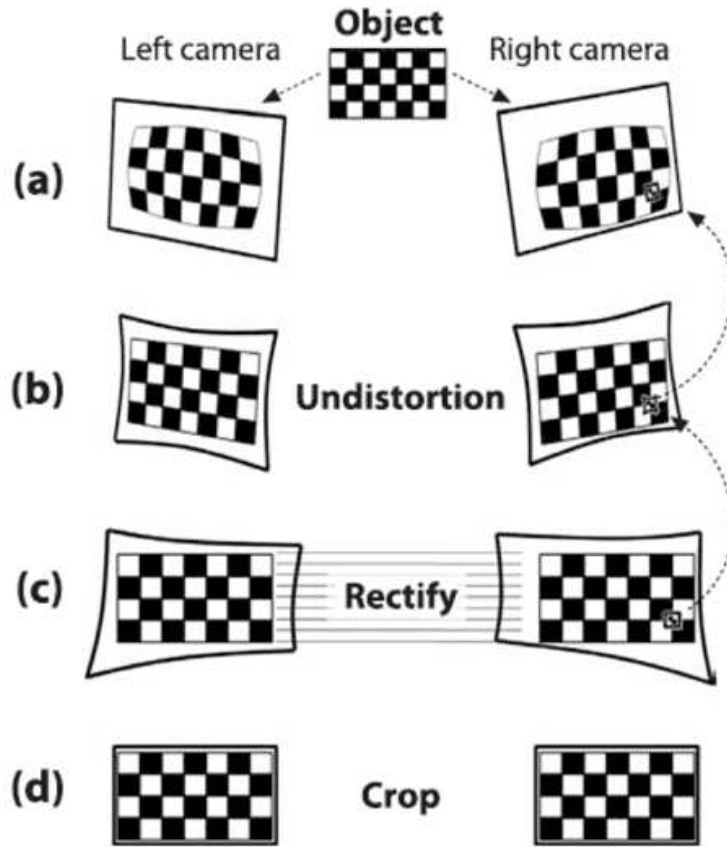


Fig. 2.4. The stereo camera calibration process.

2.8. This relative location will be used to rectify the left and right images. As a result, the projection pixel of a 3D point M in the right image will only have a horizontal direction shift with the correspondent pixel in the left image. The dense points matching problem converted from a 2D matching problem to a 1D matching problem after the calibration and rectification process.

$$\begin{bmatrix} R_r & T_r^T \end{bmatrix} = \begin{bmatrix} R_l & T_l^T \end{bmatrix} * R + T \quad (2.8)$$

Usually the stereo matching problem is formulated as a labeling problem [77] [78]. In a general labeling problem, a cost volume C is constructed which indicates the costs at displacement d at pixel i . In the stereo matching problem, it is the grayscale different

between pixel $i = (x, y)$ in the left image to pixel $i' = (x + d, y)$ in the right image. The cost volume C will be filtered by a smoothing filter W :

$$C_{i,d} = \sum_j W_{i,j}(I) C_{j,d} \quad (2.9)$$

The problem of recovering 3D shape and structure information from more than two images (multi-view) from different view points has been widely explored [48, 79]. Shape from motion (SfM) was introduced in [80]. In [81] a simultaneous reconstruction of a unknown 3D scene and the camera positions and orientation is described using a set of feature correspondences. The feature correspondences property is shown in Figure 2.5, M is the 3D point in the world coordinate; A, B, C are the camera centers; and a, b, c are the image pixels that correspond to M when camera center is located at A, B, C , respectively.

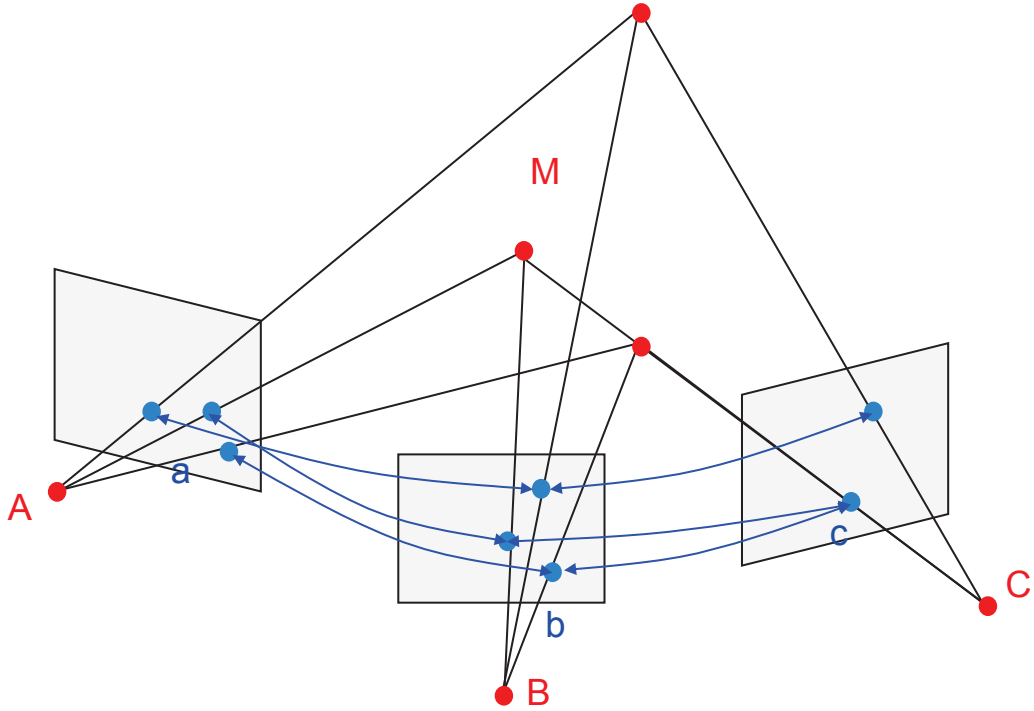


Fig. 2.5. An example of three world points M seen in three images.

Single image reconstruction has a fundamental advantage over other methods and that is user burden. With multiple views users are required to acquire multiple images from dif-

ferent viewpoints, in single view, users are only required to acquire one image of the foods. We explore both single view and the use of the multi-view image-base 3D reconstruction techniques. Single view volume estimation method highly depends on the contextual information about the food object.

2.3 Template-Based Volume Estimation Using A Single View

The TADA system initially used an volume estimation method based on a single image [26, 27, 82]. This was implemented as a template-based system for 3D reconstructions of specific shaped food objects. This volume estimation method is greatly influenced by the camera angle (viewing angle). If the view angle is too large, then it will be difficult to retrieve the height information. In contrast, if the view angle is too small, the checkerboard may be hard to detect and some region information of the food may be lost. In the experiments conducted by Chae et al. [27], the range of the appropriate camera angle is 30 – 45 degrees. We developed the user interface of the TADA mobile application (mpFR) as shown in Figure 2.6 to assist the user in taking an image at proper angles utilizing the gyroscope in the mobile phone [9].

Our goal here is to extend the earlier TADA methods and obtain an accurate 3D volume estimation for food items from a single image. As shown in Figure 2.7, the intrinsic and extrinsic camera parameters are first estimated using the fiducial marker. Then the segmentation mask of the food object must be obtained and used as a prior information for the next step - corner detection. The local corner points of the segments are detected and their correspondent 3D locations in the world coordinate are computed for volume estimation.

After camera calibration and obtaining the camera pose information, we use a food-specific shape template to reconstruct the 3D model of the food item. Two inputs are used, the segmentation mask that indicates the location of each food item and the food label obtained from the image segmentation and classification. Based on the food type, a food template shape (e.g., cylinder, and square box) is chosen for this specific food type. We then detect the corners from the segments to size the food template shape.

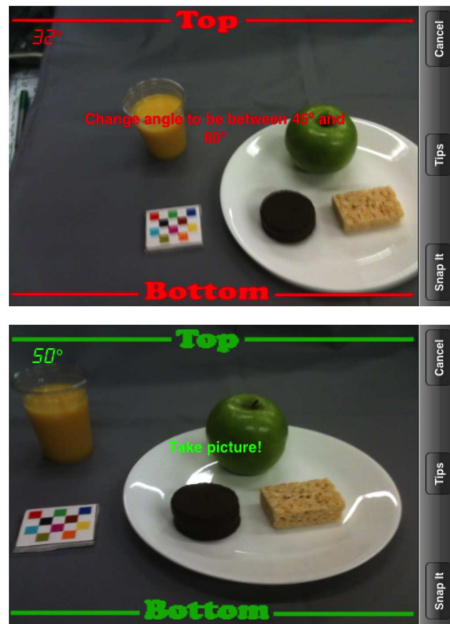


Fig. 2.6. Guide colors along with prompts in the mpFR used to assist the user in taking an image at preferred angles.

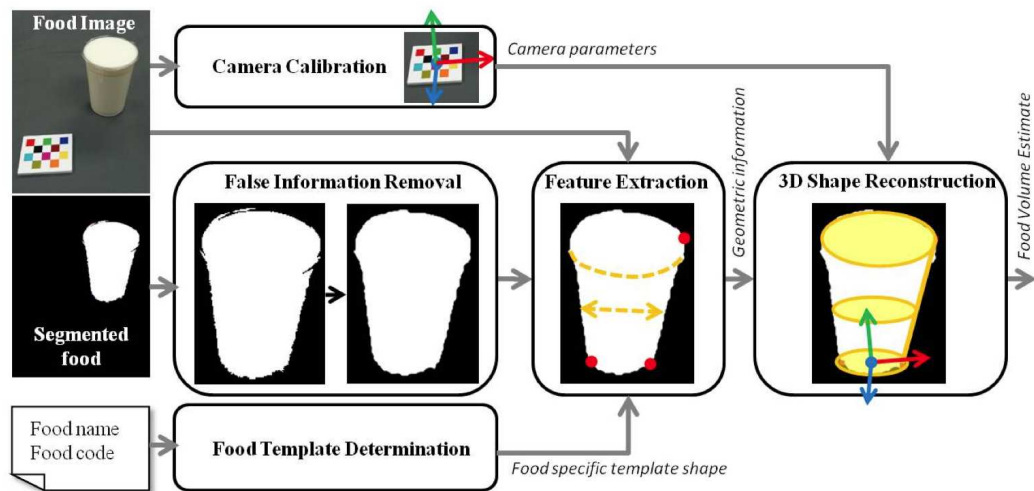


Fig. 2.7. Food volume estimation using food specific shape templates (from [27]).

Since the template features are highly dependent on the accuracy of the segmentation mask, the mask is “smoothed” using mathematical binary erosion to reduce the noise as

shown in Figure 2.7. We then detect feature points from the smoothed segmented region. We define these feature points as the dominant points or corner points which can be used to estimate the size of the shape template. For instance, two bottom corner points are used to estimate the radius of the bottom area and one top corner point to estimate the height of the cylinder.

In Equation 2.10, the curvature function C_i of all the points $p_i = (x_i, y_i)$ on the contour of a segmentation mask is obtained. We select the points with the three maximum curvature C_i as the feature points of this mask.

$$C_i = \frac{S_{yi} - \lambda S_{xi}}{\sqrt{\frac{1}{K} \sum_{k=1}^K (y_i^k - \bar{y}_i^1)^2} - \lambda \sqrt{\frac{1}{K} \sum_{k=1}^K (x_i^k - \bar{x}_i^1)^2}} \quad (2.10)$$

where K is the total number of contour points, \bar{y}_i^1 and \bar{x}_i^1 is the average value of x_i and y_i , and λ is a weight value depending on the specific shape template.

As mentioned before, the feature points of a cylinder shaped template are the top-right corner TR , bottom-left corner BL , bottom-right corner BR on the segmentation mask, and the 3D back-projection points of the feature corners are defined as TR' , BL' , and BR' correspondingly. These points can be used to estimate the top radius, bottom radius and the height of the cylinder shape. The feature points are then back projected onto the 3D world coordinate using Equation 2.28 using the assumption that the bottom points are on the same plane as the checkerboard. Then, we use the location of the feature points in the 3D world coordinate to obtain the dimension of the 3D cylinder food object. The distance between the 3D points BL' and BR' is used to estimate the radius of the cylinder r , and the distance between the 3D points TR' and BR' are used to compute the height of the cylinder h . Finally, the volume of this cylinder shaped food object is computed by using $volume = \pi \times r^2 \times h$.

Two geometrical shape templates namely cylinder and square box were investigated with the template-driven system we described above. In addition, for irregular and arbitrary shaped food object, a prismatic approximation model was proposed in [26] in a semi-automatic manner. The assumption of this model is the segmentation mask of this food

item has the same area with the physical region that the food adjacent to the table or plane surface with. And the area of cross section of the food is approximately the same along the vertical direction.

As shown in Figure 2.8.(c), the segmented region of scrambled egg is extracted manually by a user or automatically using the food image segmentation method in [8, 25]. This region will be used as the base of prismatic approximation model. The physical dimension of this region is computed by first using Delaunay triangulation [83] to partition the planar region. Then, the area of the planar polygon is computed by adding all the areas of the triangles. Afterwards, the 3D prismatic approximation model is reconstructed by manually extruding the planar area in the vertical direction by the user as shown in Figure 2.8.(d). The volume of the food item is estimated by computing the volume of the 3D prismatic model which is generated by the user.

2.4 Model-Based Food Volume Estimation using 3D Pose Registration for A Single View

In this section we will extend some of the earlier done in the TADA system to increase the accuracy of the volume estimation. 3D shape recovery involves a one-to-many mapping from a 2D image to a 3D space. Therefore, single-view 3D reconstruction in general is an ill-posed inverse problem. However, if the location and identification of the food is known and a 3D model of this food item is provided, we can simplify the 3D reconstruction problem to a 3D-to-2D pose registration problem. 3D models of food items need to be trained and stored in the database prior to volume estimation.

The single-view based method estimates food volume by using prior information - segmentation and food labels generated from food identification methods [12, 25, 41, 42, 57]. In Section 2.3, we used a shape template method for 3D reconstruction of some food objects [27]. We have utilized the corner points from the segmented image to compute the geometric information for the shape template. However, this method is highly dependent

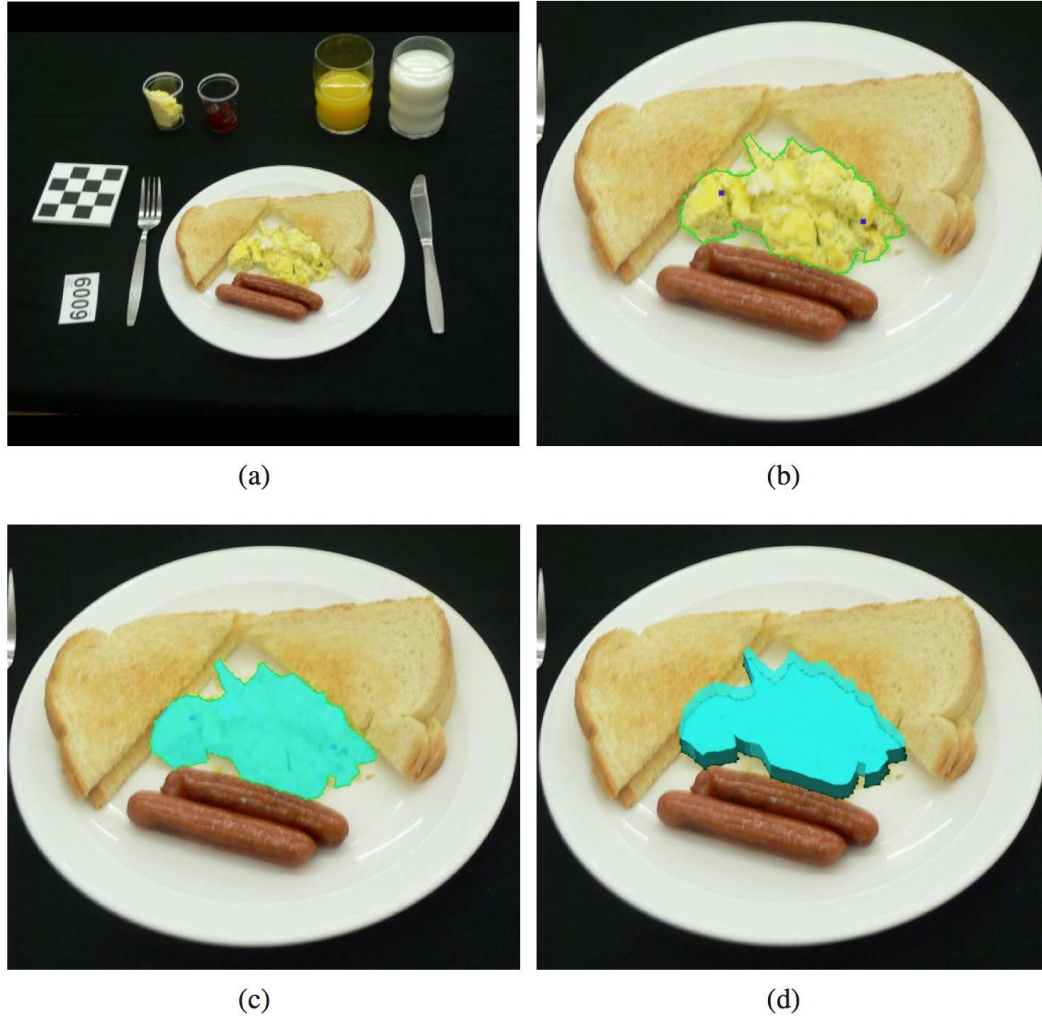


Fig. 2.8. Volume estimation using prism shape template approximation; (a) the original image, (b) contour of the food item (scrambled eggs) is segmented (c) the contour is used with the prismatic approximation model (d) the reconstructed volume model of the food item (from [26]).

on the accuracy of the segmentation method and the corner points detection is not robust. Moreover, it fails when the food item has a complex and amorphous shape.

In this section, we will propose a volume estimation method [59] that utilizes a 3D graphical model of the food object from multiple training images. First, we create a 3D graphical model during the training step using 3D reconstruction from multiple views.

Then, for each food image, we determine the pose and scale of each of the food items according to the camera matrix and food segment.






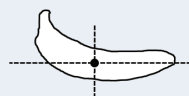
Shape	Example Food Type	Dimension Parameters	Locator
Cylinder	Orange juice, Milk	Radius, Height	
Sphere	Apple, Orange	Radius	
Square Box	Chococolate Cake, Brownie	Width, Length, Height, Rotation Angle	
Slice of Cone/ Slice of Sphere	Spaghetti, Ice Cream	Top Radius, Bottom Radius, Height	
Prism	Bread, Scrambled Eggs	Area, Height	
Irregular Shape	Banana, Pear	Scale X, Scale Y, Scale Z, (Rotation Angle)	

Fig. 2.9. Shape dictionary: food items with their corresponding 3D shapes.

We will also extend our method in [59] by pre-defining some conventional geometrical 3D models. Instead of reconstructing the 3D models in the training step, we generate several conventional 3D shape models. Some food items can be represented by these conventional shape models (e.g. cylinder, sphere, and cone). Using this prior knowledge, we can greatly improve the accuracy of the volume estimation for food objects. For other food items whose shape cannot be approximated to a regular shape model, we use a prismatic model to approximate the shape.

For our model-based method (Figure 2.12), we designed a shape dictionary consisting of pre-defined or pre-built 3D models. Examples of food items along with their 3D models are shown in Figure 2.9. There are conventional shapes in the shape dictionary such as

sphere, cylinder, box, and slice of cone. Nevertheless, the food shapes can be quite complex. Therefore we pre-build the 3D model for some food shapes (bananas and pears) in the dictionary by using the method described in [10].

The conventional shape models can be generated without training images. For the sphere, the dimensions (x, y, z) for each voxel can be obtained by Equation 2.11:

$$x^2 + y^2 + z^2 \leq radius^2 \quad (2.11)$$

A cylinder can be represented by Equation 2.12:

$$\begin{cases} x^2 + y^2 \leq radius^2 \\ z \leq height \end{cases} \quad (2.12)$$

A square box is defined by Equation 2.13:

$$\begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ z_{min} \leq z \leq z_{max} \end{cases} \quad (2.13)$$

A slice of a cone is represented by Equation 2.14:

$$x^2 + y^2 \leq \left(bottom_r + \frac{z}{height * (top_r - bottom_r)} \right)^2 \quad (2.14)$$

In our system, we first need to calibrate the camera. A credit-card-sized colored checkerboard is used as the fiducial marker as described in Section 1.2, which is included in every image as a geometric reference for the scale of the world coordinates and to provide color calibration information [58]. We can then estimate the camera pose from the checkerboard and establish the world coordinates. We assume that the food image has been correctly segmented and identified, and segmentation masks and food labels are generated accordingly [12, 25, 41, 42, 57]. The segmentation mask provides the location of a food item and the food label indicates the food identification. A locator M on the segmentation mask is also used to define the 3D coordinate (x, y, z) of the food item. The locator is chosen in a manner that it is easy to locate on the 2D image and back project to 3D space. For all the conventional shape models, the locator is defined to be the lowest point on the segment food

image. To make the position of the locator more robust, we draw a vertical line through the centroid of the mask, and choose the lowest point on the line as the locator M . The locator M on the food segment is shown in Figure 2.9. The locator M for the prismatic model or irregular shape model is simply defined as the centroid of the segmentation mask.

In most of the circumstances, the locator M on the food object will be close to the table cloth or eating surface. Therefore, the 3D coordinate of this point is easy to obtain by using Equation 2.15:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K * [R|T] \begin{bmatrix} X \\ Y \\ Z = 0 \\ 1 \end{bmatrix} \quad (2.15)$$

where s is a scale factor, u, v are the location of pixel on the image, K, R, T are respectively the intrinsic camera matrix, the rotation vector, and translation vector.

The identification of the food is used to indicate the shape model we shall use. Since the food model is either pre-built or pre-defined, we can convert the 3D reconstruction problem into an optimization problem. The object has 9 degrees of freedom (DOF):

$$W = (X, Y, Z, \Theta_X, \Theta_Y, \Theta_Z, s_x, s_y, s_z)^T d \quad (2.16)$$

Equation 2.16 consists of the object translation along three coordinate axes, three rotation angles to the axes, and three relative scale parameters. The best match pose and scale parameters of the food item can be found by using Equation 2.17:

$$W = \arg \max_W (similarity_measure(I_{seg}, I_{project}^W)) \quad (2.17)$$

where I_{seg} is the segment of the food item, $I_{project}^W$ is the projected image using a set of pose or scale parameters. The *similarity_measure* is a function indicating the similarity score between two images. For conventional shapes, we will use the XOR binary operation to find the similarity since we only have the silhouette of the projected object. For reconstructed model [59], we use a normalized cross correlation function since the textured projection image is obtained in this method. The 3D locator M can be used to determine X, Y, Z . The degrees-of-freedom of the pose for different foods is dependent on the 3D shape model. We

utilize several geometric constraints and food placement constraints (e.g. a banana will not be standing on its long axis) to solve the pose registration problem [59]. After the pose of a food item is determined, we are able to estimate the volume of the food by computing the volume of the shape model.

2.4.1 Prism Model Volume Estimation

For the prism technique, the base area is computed by rectifying the segmentation mask using the 2D homography matrix obtained from the checkerboard corners [61]. The homography model is widely used for rectification of a projective camera image [61]. It describes the relationship between two camera projective areas of the same planar object in 3D space projects from two views. Since the twelve corner points can be precisely located in the image and the geometry layout is known, we can rectify the image using this information. Corner points are detected from the checkerboard and labeled as $p_{src}^1, p_{src}^2, \dots, p_{src}^{12}$ in the following order: top to bottom, left to right. We define the 3×3 projective transformation matrix H as a mapping from the corner points in the original image to the rectified image.

$$p_{dst} = Hp_{src} \quad (2.18)$$

where p_{dst} is the corresponding corner points in the rectified image. The homography matrix H has nine elements with only their ratio being significant, therefore the transformation is specified by 8 parameters. The problem then becomes given the twelve 2D to 2D point correspondences $p_{src} \mapsto p_{dst}$, determine the 2D homography matrix H such that $p_{dst} = Hp_{src}$.

The typical solutions to this well-known over-determined problem are Direct Linear Transformation (DLT) [61], RANdom SAmple Consensus (RANSAC) [84], or the Gold Standard [61]. DLT is in general used for initialization and the two other methods are more robust to outliers. Since the checkerboard corner detection is robust and precise, we use the DLT method [61] to estimate H as follows. Given a set of four 2D to 2D corresponding

corners $p_{src} \mapsto p_{dst}$, $p_{dst} = Hp_{src}$ can be written as the vector cross product p_{dst} and Hp_{src} and have the same direction.

Therefore, the equation can also be expressed as:

$$\begin{bmatrix} 0^T & -w'p_{src}^T & y'_ip_{src}^T \\ w'_ix_i^T & 0^T & -x'_ip_{src}^T \\ -y'_ip_{src}^T & x'_ip_{src}^T & 0^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0 \quad (2.19)$$

where $p_{dst} = (x'_i, y'_i, w'_i)$ and $H = \begin{pmatrix} h1 & h2 & h3 \end{pmatrix}^T$. H will be found using Equation 2.19 with the original images.

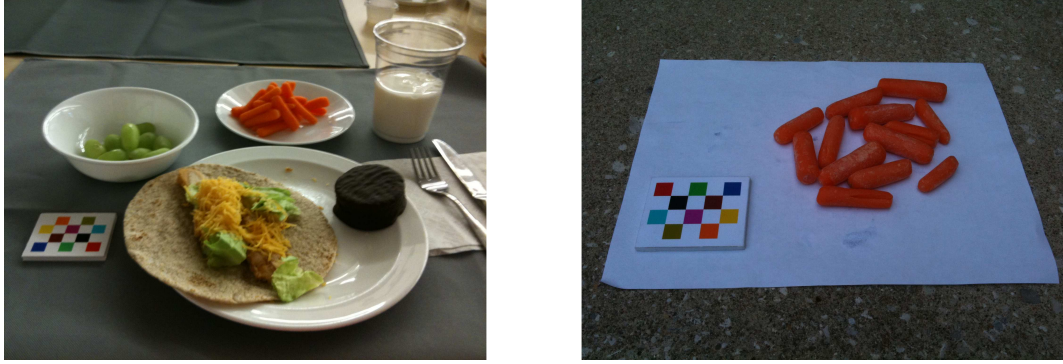


Fig. 2.10. Examples of the same food type (carrot) with different heights.

By using Equation 2.19, we can transform the image into a front view image. And then, the approximate area of the food item is computed using the segmentation of the food item after the transformation.

The area of the rectified segment is then used as the base of prismatic shape and we then need to estimate the height of the food object. Figure 2.10 illustrates that two different displacements for the same food item (carrot) can have different heights. The height information is hard to determine for arbitrary shaped food objects. Therefore, we choose an average height for each food type and multiply the area of the base to estimate the food volume. The height of a food item varies from scene to scene. However, we can use a predicted height using training images. Several images of broccoli are captured and the area of the food is computed for each image as shown in Figure 2.11. The x axis is the estimated

height and the y axis is the estimated weight with respect to the height. The height with minimum error will be chosen as the average height for this type of food.

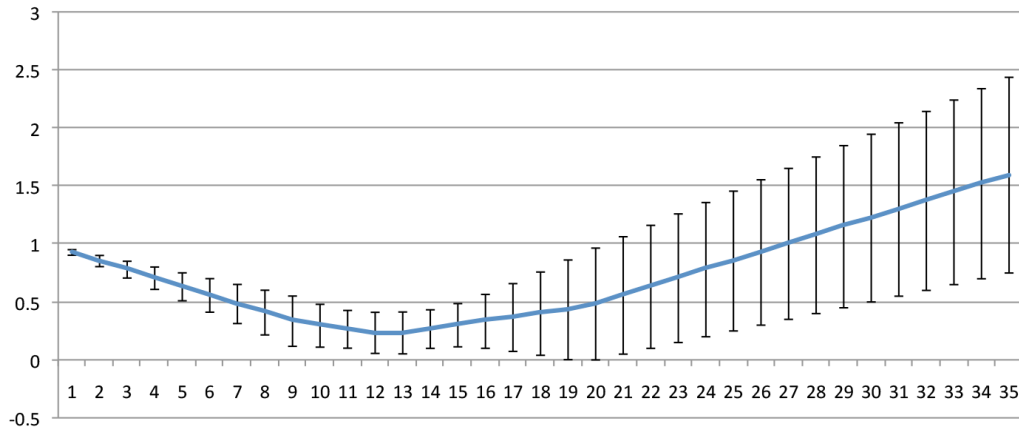


Fig. 2.11. The percentage weight estimation of broccoli with respect to the height.

Once the volume is estimated, the nutrient content of the food is obtained using the density for that particular category of food [31]. This is described in more detail in Section 3.3.

2.4.2 3D Model Generation

In order to obtain the 3D models for the shape dictionary, we first need to reconstruct the model of the food from multiple images or a video sequence (step 1 in Figure 2.12). After taking multiple images or a video of the food items, we use a variation of a multi-view shape recovery method - *Shape from Silhouettes* [46], also known as *Backprojection Reconstruction* [85]. This method reconstructs a 3D model of an object from the set of contours that outlines the projection of the object onto a sequence of 2D image planes. The ideal image acquisition step for shape from carving is to acquire images of the object from different view angles such as a turn-table.

The typical number of images required for most food items is 15 to 20 images. These images can be obtained from video frames or by capturing multiple still images. The se-

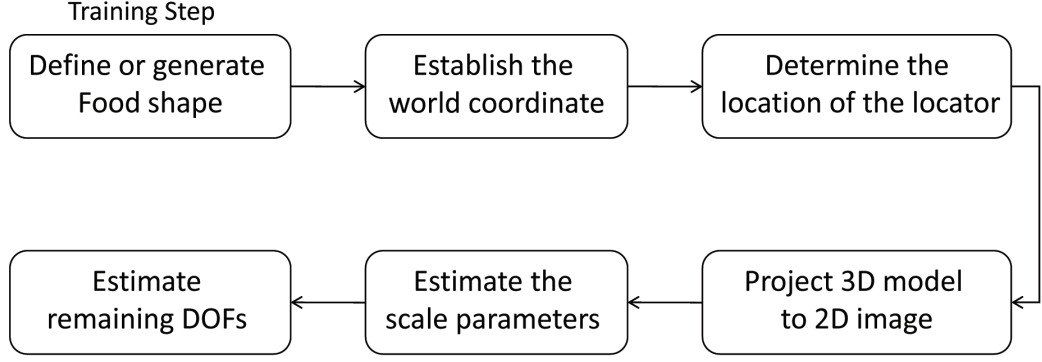


Fig. 2.12. Our single-view volume estimation system.

lection of frames can be automatically done by checkerboard detection and camera pose detection. Then, the intrinsic and extrinsic camera parameters need to be determined for each image. In our case, in order to calibrate the images, each image needs to include the checkerboard in the scene. After computing the camera calibration matrix, each camera image is converted to a binary mask which indicates the object silhouette using “1” for object pixels and “0” for background and other content. Shadow, blur and specular reflection effects could decrease the segmentation accuracy. To make our method more robust to segmentation noise, we use morphological operators to clean up the boundary and avoid small holes (less than 1% of the segmented area of a food item) in the object mask.

Next, the bounding box of the object masks from each image is back-projected onto 3D world coordinates using the camera projection matrix. Based on this 3D bounding box, we fill it with a 3D grid of volume voxels, V , for “carving.” The next crucial step is to repeatedly project every $v \in Surf(V)$ onto all the camera images c_1, c_2, \dots, c_n , where $Surf(V)$ is the surface of the volume formed by V . Any voxel that lies outside the object mask in c_i needs to be removed or carved away. As the number of projection and carving steps increases, the object 3D boundary becomes tighter. The iteration is terminated if no non-photo-consistent voxel is found. After carving away every voxel that does not belong to the 3D object model, we obtain the 3D voxel-based model for this food item. We also estimate the volume of the food object by counting how many voxels are left and the size of the voxels from the world coordinate. As shown in Figure 2.14, we project the

reconstructed 3D banana model onto the 2D image plane with various rotation angles. This is a training step and it is done prior to volume estimation.

2.4.3 Pose Registration

Once the model of the food has been reconstructed, we use it to estimate the geometrical state of a food object in world coordinates. In general, an indoor and small scale object has nine degrees of freedom (DOFs) as in Equation (2.20),

$$W = (X, Y, Z, \Theta_X, \Theta_Y, \Theta_Z, s_x, s_y, s_z)^T, \quad (2.20)$$

where the parameters consist of the object translation along three coordinate axes, three rotation angles to the axes, and three relative scale parameters (see Figure 2.13).

When estimating the pose of a food item on a table, we place geometric constraints on the pose. The first constraint is that we assume the food object is adjacent to the table plane. Next (step 2 in Figure 2.12), we define the world coordinates with the checkerboard: one corner of the checkerboard pattern is assigned as the world origin O_w as shown in Figure 2.13. The x-axis and y-axis are aligned with the lines on the checkerboard thus the x-y plane approximates the table plane. As a result, the 3D point P on the object bottom surface has $Z_w = 0$. Most food objects have only one placement position on the table. For example, a banana usually lies on the table on its side instead of standing up on the table. Accordingly, two remaining rotation angles of the food object on the table are represented by the azimuth angle ϕ and the elevation angle θ . The azimuth ϕ , is the horizontal rotation about the z-axis from the negative y-axis. The elevation θ is the vertical elevation of the viewpoint moving above the x-y plane (Figure 2.13).

We also observe that most non-rigid foods and beverages have isotropic shape (e.g. an orange) or symmetric and balanced shape (e.g. orange juice and donuts). For these food items there is not much variation in the ratio of the width to the length of the foods. Therefore, we can safely fix the ratio of s_x to s_y with respect to the ratio we obtained from the 3D model for these foods. For food where the ratio of s_x to s_y has considerable variation from sample to sample (e.g. a banana), we currently use the average ratio of two dimensions

s_x, s_y to approximate the 3D model of a banana. For other foods, we obtain the possible range of the s_x to s_y ratio for the specific food and then project the 3D reconstructed model by varying the ratio along with the azimuth angle ϕ as well.

After determining the three DOF, the coordinate representation of the object can be written as in Equation (2.21)

$$E = (x, y, \phi, \theta, s_x, s_z)^T, \quad (2.21)$$

where x, y are the locator, ϕ, θ are the azimuth and elevation angle, and s_x, s_z are the size factors.

First, we use the lowest 2D point p within the object contour on the image to define the object location (step 3 in Figure 2.12). We find two displacement parameters x, y by back-projecting the 2D point $p(p_x, p_y)$ to its corresponding 3D surface point $P(x, y, 0)$ using Equation (2.22) [61]

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}, \quad (2.22)$$

where f_i, C_i, r_i, t_i are respectively the focus length parameter, principle point location, the rotation matrix, and the translation vector obtained during the camera calibration step using the checkerboard. Elevation angle θ will be determined by the right triangle consisting of the camera center C , the object point P , and the projection point C' that C projects onto the x-y plane as shown in Equation (2.23):

$$\theta = \arcsin\left(\frac{C_z}{\|(C_x, C_y, C_z) - (P_x, P_y, 0)\|}\right). \quad (2.23)$$

Two scale parameters s_x, s_z are given by the length and height of the bounding box on the segmented food.

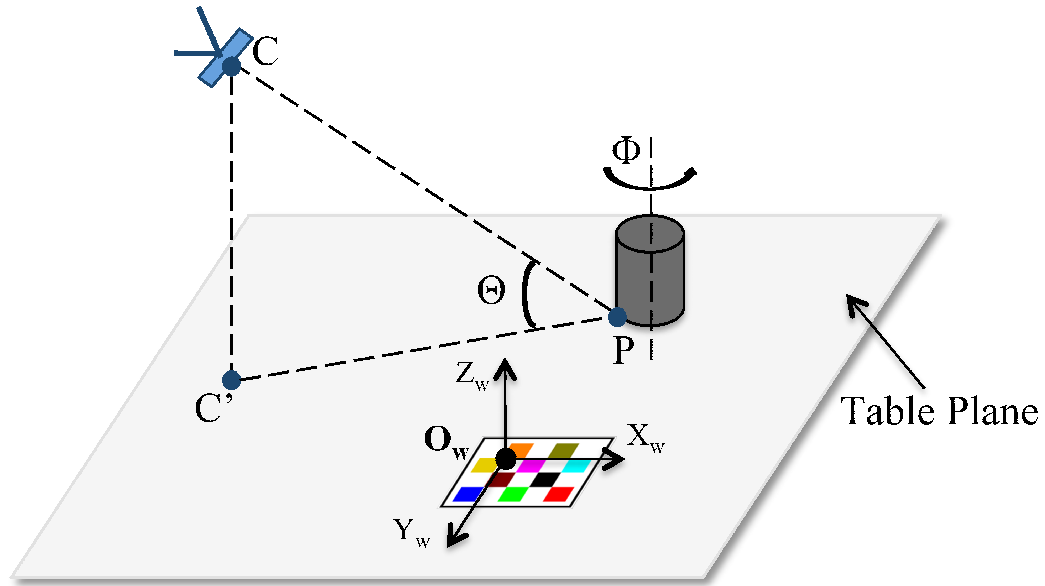


Fig. 2.13. The geometric relationships between the camera center and the food object.

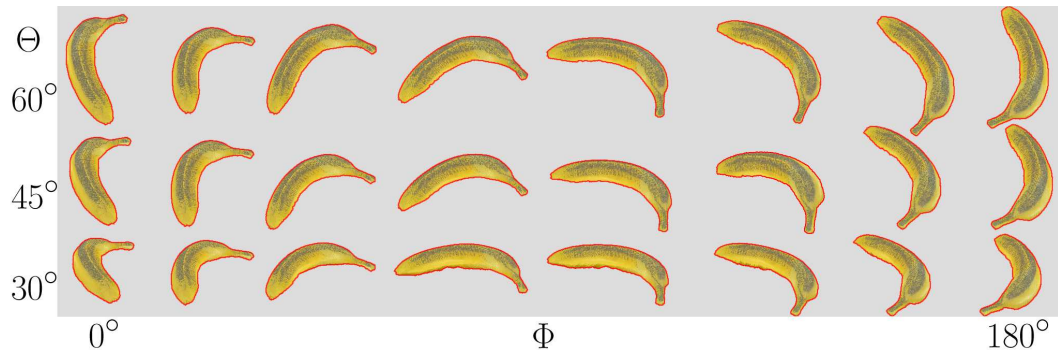


Fig. 2.14. An example of projecting a 3D banana model to a 2D image plane with two pose angles. The elevation angle ϕ is varied from 0° to 180° , and the azimuth angle θ is varied from 30° to 60° .

2.4.4 Volume Model Finalization

For rotation symmetric food items (e.g. bagel and orange juice), the object pose is invariant to the azimuth angle ϕ . In other cases, the self-rotation angle ϕ can be estimated by image patch matching. We have trained the food 3D model and determined the elevation angle θ . We then project the 3D food model with various size factors s_x, s_z and azimuth

angle ϕ (step 4 in Figure 2.12). Based on the fixed θ, x, y , we project the 2D projection images by varying the self-rotation angle ϕ as shown in Figure 2.14. The DOFs are sampled at 10° intervals for the self-rotation angle ϕ and $1mm$ for the size factors s_x, s_z . The last DOFs ϕ, s_x , and s_z are chosen by measuring the similarity between each projected image G_ϕ with the segmented food image F as shown in Equation (2.24).

$$\phi = \arg \max_{\phi} \left(\frac{\sum_{i,j} [F(x_i, y_j) - \bar{F}] [G_\phi(x_i, y_j) - \bar{G}_\phi]}{\sigma_F \sigma_G} \right), \quad (2.24)$$

where $F(x_i, y_i)$ is the gray scale value of a point (x_i, y_i) in the segmented food image patch. \bar{F} is the average intensity value of the image F . $G_\phi(x_i, y_i)$ is the gray scale value of the point (x_i, y_i) on the sampled image after normalization in reference to ϕ . \bar{G}_ϕ is the average intensity value of the image G_ϕ . i, j represents the pixel index. σ_F, σ_G are the standard deviation of $F(x, y)$ and $G_\phi(x, y)$, respectively.

After the object pose is finalized, we render the predefined 3D model of the food into the world coordinate in the eating occasion. Food volume is estimated based on the volume of the 3D rendering model.

2.5 Stereo Vision/Two View

Stereo vision and stereo cameras have many applications in scene understanding, depth computing, gaming, and 3D reconstruction. The mobile industry is also looking into involving 3D technology with their devices. Several mobile telephone models with 3D stereo camera are on the market. With this evolving technology in mind we investigated how to utilize stereo vision techniques in our system.

In the stereo/two view reconstruction problem, two common geometry models are used: the epipolar geometry in Figure 2.15 and the fundamental matrix F . When two cameras view a 3D scene from different positions, there are geometric relationships between the 3D points and their projections onto the 2D images that constrain the projected image points. These can be derived using epipolar geometry. Epipolar geometry is used to represent the intrinsic projective geometry between two views. It does not depend on any information

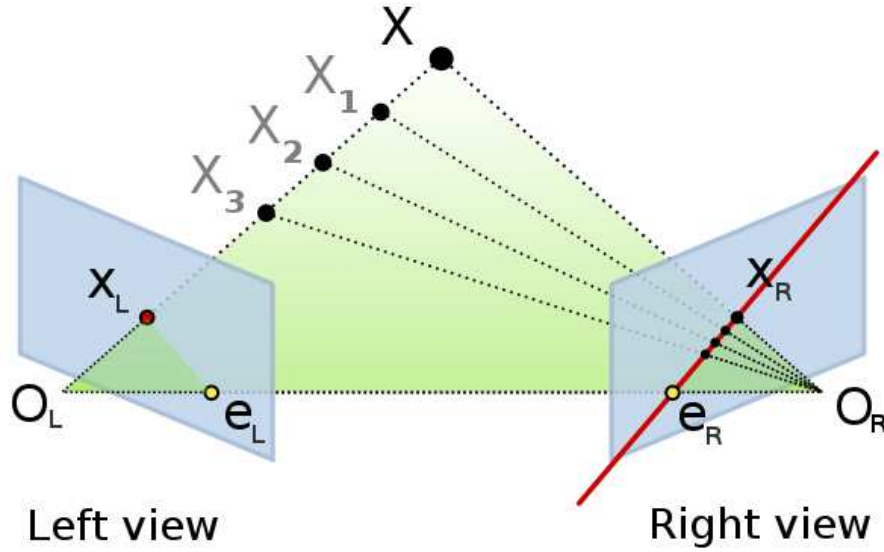


Fig. 2.15. Epipolar geometry for toe-in views.

from scene structure, but only depends on the two cameras' internal matrix and relative pose between them. As shown in Figure 2.15, O_L and O_R are two cameras capturing the same scene and X is a 3D point in the scene. Three points X , O_R , and O_L are defining an epipolar plane as shown the green area in the image. e_L is the projection point of O_R on left view O_L and e_R is the projection point of O_L on left view O_R . The line which is connecting X_R with e_R is defined as epipolar line.

This geometric relationships can also be defined by using the fundamental matrix F which maps a point on view 1 to a line on view 2. The fundamental matrix F is a 3x3 matrix of rank 2. It can be obtained from 7 pairs of homogeneous points in 2 images or two camera matrices P and P' from two views as in Equation 2.25:

$$F = [e'] \times P'P^+ \quad (2.25)$$

where P^+ is the pseudo-inverse of P , and $e' = P'C$ where $PC = 0$. Figure 2.16 demonstrate this correspondence between a point x with its epipolar line l' as in Equation 2.26.

$$l' = Fx \quad (2.26)$$

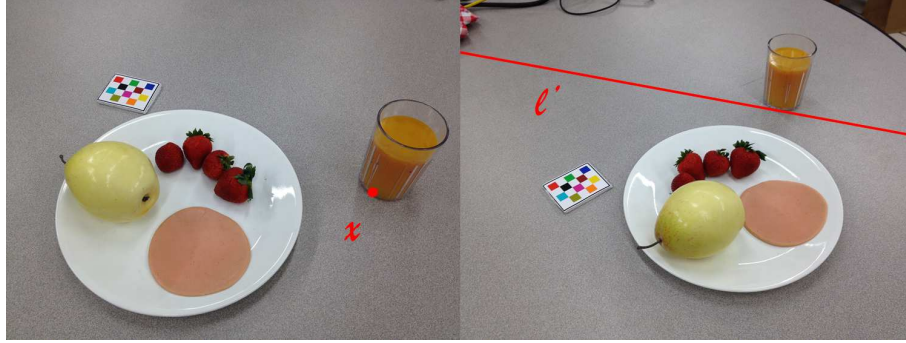
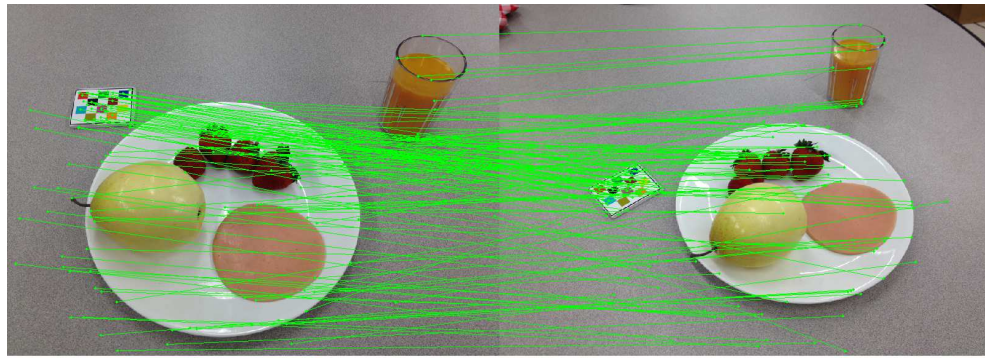


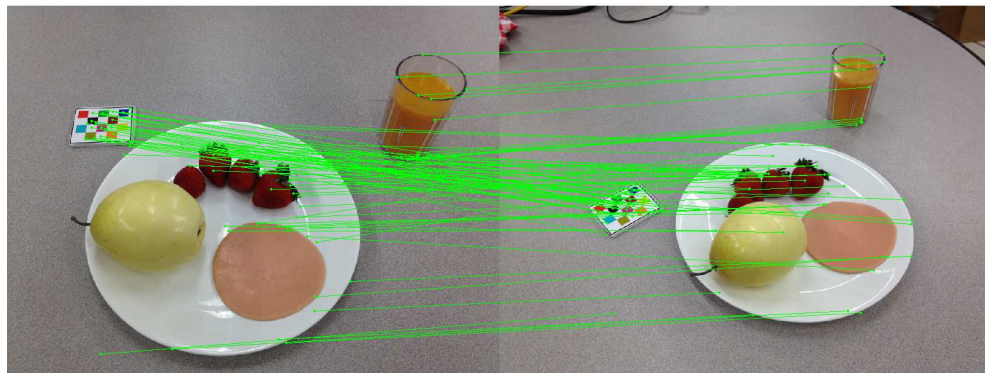
Fig. 2.16. An illustration of correspondence between a point on view 1 with a line on view two.

In Figure 2.17, we demonstrate how this epipolar constraint improves the accuracy of feature matching. The Scale-Invariant Feature Transform (SIFT) [86] is a widely used in computer vision for feature detection or description. We use SIFT keypoints for feature matching of two images from View1 and View2 of the scene for various view angles with the following approach: First, two sets of SIFT keypoints will be detected from View1 and View2. Then, the first set of SIFT keypoints from View1 will be individually compared with each keypoint in View2 based on Euclidean distance of their feature vectors. Finally, two nearest neighbor features from two sets of keypoints will be identified as a match.

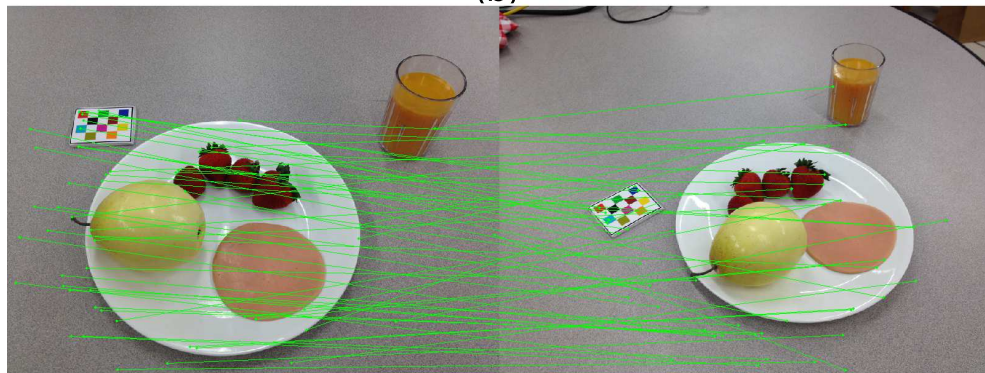
Figure 2.17(a) shows the feature matching result directly from SIFT. As shown, the correspondent features of the two images of the same scene from different view angles are connected with green lines. Many features are mismatched because of the limitation of SIFT: As the view angle is changed, the number of mismatches will increase [86]. Figure 2.17(b) demonstrates the feature matching result after using the epipolar constraint. The point matching accuracy can be improved by removing the matching points that fail the epipolar constraint in Equation 2.26. The matches which are been removed with the epipolar constraint are shown in Figure 2.17(c).



(a)



(b)



(c)

Fig. 2.17. An example of feature matching utilizing the epipolar constraint. The correspondent points in left and right images are connected by the green line. (a) is showing all the correspondent points using SIFT matching, (b) shows the point matches fulfill the epipolar constraint, (c) shows the point matches fail the epipolar constraint.

2.6 Multi-View/Video Reconstruction

Although the proposed single view shape model-based method provides a consistent method for supporting various types of regular and complex shape food, it is insufficient for foods with undefined or unknown shape. Many foods have a large variation of shapes due to eating conditions, i.e. cooking preparation and lighting condition. The single view approach highly depends on the accuracy of the segmentation mask in order to detect the feature points. Thus, single-view food volume reconstruction is a difficult task and it contains limitations. Additionally, single-view methods require the shape context information related to the food items, which may not always be available to us. Thus, it is valuable to investigate the use of multi-view approaches in our system.

The previous section on stereo illustrates the geometric relationship between a 2D point on the image plane and the reprojected 3D points in the world coordinate. However, if we want to explicitly and accurately find the one-to-one correspondence of the 3D-2D point mapping, at least two images from different viewpoints are needed as shown in Figure 2.5.

The target of our volume estimation system is to reconstruct the geometric model of the food scenes. So here we consider using “shape from silhouettes” [46, 85] for multi-view 3D reconstruction problem.

2.6.1 Shape From Silhouettes

We have developed a variation of a multi-view shape recovery method - *Shape from Carving* [46], also known as *Shape from Silhouettes* [85]. This method attempts to reconstruct a 3D model from a set of contours that outline the projection of an object onto a sequence of 2D image planes.

The ideal image acquisition for using shape from carving are images of an object from various view angles such as images acquired from a turn-table device (see Fig. 2.18). Video capturing begins by acquiring a video of the food objects. Empirical evidence indicates that we obtain the best results using 14 to 20 frames from the video sequence.

$$x = K[R|T]X \quad (2.27)$$

As shown in Equation 2.27, x and X are the 2D and 3D locations of the same point on the image coordinate and the 3D world coordinates respectively, K represents the intrinsic matrix of a camera, R is the rotation matrix of the camera, and T is the translation matrix of the camera in the world coordinate. The intrinsic matrix K and the extrinsic camera parameters R, T are determined for each image. In our case, in order to calibrate the images, each image needs to include the checkerboard in the scene.

Since the same camera is used for capturing the sequence of images for multiple-view method, the intrinsic matrix K is identical for each view and can be determined using the camera calibration procedure presented in [61]. After detecting the corners on the checkerboard, the pose of the camera, the rotation vector R and the translation vector T , can be found by minimizing the reprojection error. This reprojection error corresponds to the image distance between the projected 3D corners using the intrinsic matrix K and the extrinsic camera parameters R, T to the location of the detected corners on the image [61].

After determining the camera parameters, each camera image is converted to a binary image using the segmentation mask which indicates the object silhouette using “1” for object pixels and “0” for background.

This method requires accurate segmentation. We use morphological operators to clean up the boundary and remove small holes on the segmentation mask.

There are mainly two types of 3D representation: volumetric models and 3D surface grids [87, 88]. We use the volumetric representation of a 3D object as shown in Figure 2.19. The bounding box of the cleaned object masks obtained from the segmentation masks is then back-projected onto 3D world coordinates using the camera parameters (camera matrix) and Equation 2.28:

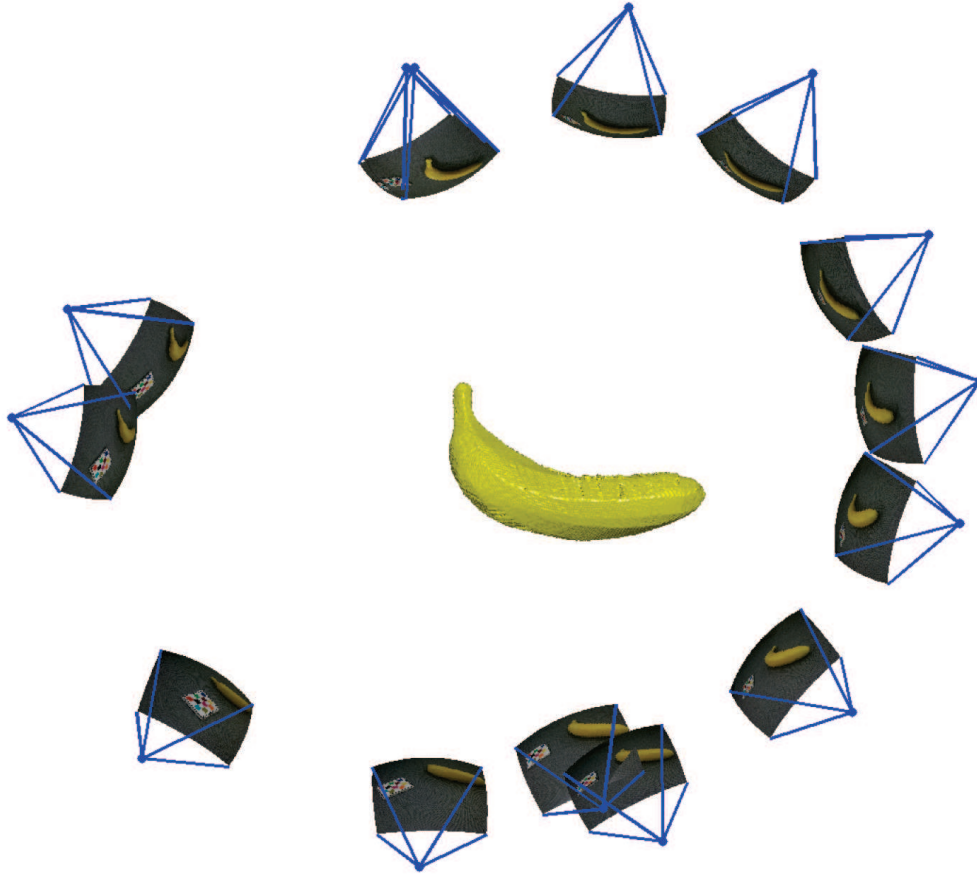


Fig. 2.18. Training angles for a food object and the reconstructed 3D model.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.28)$$

where (X, Y, Z) are the coordinates of a 3D point M in world coordinates, (u, v) are the coordinates of the 2D projection point m in pixels of the image coordinate. C_x, C_y are the coordinates of the principal point (we locate it at the center of the image), f_x, f_y are the focal lengths in the unit of pixels. r_i, t_i are the the rotation and translation external

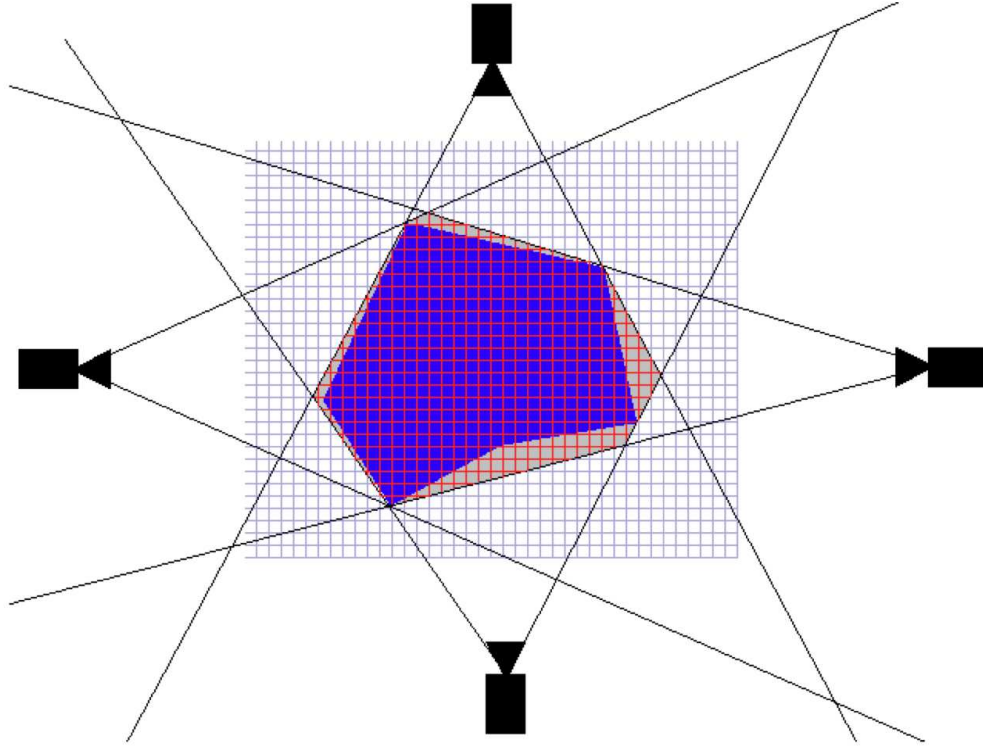


Fig. 2.19. An example of voxel based marching intersection.

parameters that relate the world coordinate system to the camera image coordinates. This equation is used to back-project 2D image pixel to its correspondence 3D world point.

Based on the 3D bounding box, we fill a 3D grid of volume voxels for carving. The next step is to back-project the silhouettes onto the 3D volume voxels one by one. These silhouettes are used to bound the object and carve away any voxels outside the reprojected mask. As the number of the silhouettes increases, the object 3D boundary becomes tighter. After carving out every voxels that does not belong to the 3D object model, we estimate the volume of object by counting how many voxels are left and the size of voxels obtained from the world coordinate.

In the following Section 2.7, we evaluate both methods, i.e. single view-based reconstruction and multi-view/video-based reconstruction, and compare them with ground truth information.

2.7 Experimental Results and Conclusion

2.7.1 Model-Based vs. Template-Based

The initial performance comparison of our proposed single-view volume estimation method with the previous template based method was done by conducting an experiment with three plastic food items (orange juice, a bagel, and a banana) and one real food - a Rice Krispy treat. We are interested in comparing template-based methods [27] and our new proposed 3D-model-based method. Orange juice and a rice krispy treat can be reconstructed using a single view in an efficient manner since they have very regular shapes (cylinder and square box). A bagel could also be considered as a regular shaped object, but due to the ambiguity of its color homogeneity, height, and depth information, it cannot be clearly distinguished. Moreover, its “textureless” uniform color composition does not allow us to use shape information to distinguish the height from depth. A banana has a complex shape and there is no regular 3D geometrical template that can be used from the 2D segmentation mask. Orange juice and Rice Krispy Treat are examples of foods where the regular shaped model can be used, whereas bagels and bananas require a pre-built model for their volume reconstruction.

The images are captured using the integrated camera available on the iPhone 3GS. We obtained 15 to 20 images for training as discussed in Section 2.4.2 and acquired 35 images per food from various view angles and estimated their corresponding volume using our technique. The results of the estimated volume (mean and standard deviation) for these three plastic food items (banana, bagel, and orange juice) and one real food - Rice Krispy Treat are shown in Table 2.1 in terms of milliliters and compared with the ground truth volume obtained from water displacement measurement.

The results of the banana and bagel for the template-based method [27] are not available because this method cannot be used for foods without regular shapes. The results of orange juice and the Rice Krispy Treat using the template-based method are satisfactory but our method further improves the volume estimation accuracy. In addition, we observe the template-based approach is sensitive to segmentation noise since it is based on feature ex-

Table 2.1

Comparison of a template-based method and our model-based method for three plastic food items (banana, bagel, and orange juice) and one real food - Rice Krispy Treat. The mean and standard deviation are shown as $\mu(\sigma)$.

Food Item	Template Method(ml)	Model Method(ml)	Ground Truth(ml)
Banana	N/A	182.6(15.9)	170
Bagel	N/A	151.2(14.3)	145
Orange Juice	179.9(26.6)	193.9(15.1)	200
Rice Krispy Treat	78.8(13.6)	72.5(5.3)	70

traction. Overall, our new 3D model based method achieves an average volume estimation error of 10%. Given that portion size estimation errors of more than 50% from human observation have been reported in the use of traditional dietary assessment methods [22, 89], our results are reasonable and exceed traditional approaches.

Another evaluation of our single-view volume estimation methods was done by conducting an experiment with 15 Adolescent participants under controlled conditions [8, 22, 49, 90]. Images of their eating occasions over a 24 hour period were captured with three different types of cameras: Cannon PowerShot S3, Cannon PowerShot SD200, HTC p4351 mobile telephones. A total of 19 types of foods and beverages were weighed, and the ground truth weight collected. The ground truth segmentation (i.e. manual segmentation) is used in this experiment because the segmentation noise is relatively large in this study since each eating occasion image contains about 7 food items.

After the automated volume estimation is completed, we convert it into weight (g) using the food density based on the method derived from [31] (also see Section 3.3).

The volume and weight results for the 19 foods using automated volume analysis by food from images taken by 15 adolescents (11-18 y) during meals over a 24-hr period

Table 2.2
Estimated weight for 19 foods items using the estimated volume and apparent density compared with the ground truth weight. The mean and standard deviation of each value is also shown. (n = number of food images that contains a particular food item)

Food name	n	Apparent Density (g/cc)	Estimated volume (cc ± SD)	Estimated weight (g ± SD)	Ground truth weight (g ± SD)	Weight percentage error
2% Milk (C)	54	0.973	226.3 ± 19.1	220.2 ± 18.5	220 ± 0.0	0.9
Sausage links (P)	22	0.863	49.9 ± 14.8	43.1 ± 12.8	46.5 ± 1.0	7.3
Scrambled eggs (P)	22	1.123	69.8 ± 30.9	78.4 ± 34.7	61.5 ± 0.7	27.5
Toast (P)	22	0.276	185.2 ± 82.8	51.1 ± 22.9	47.4 ± 3.4	7.8
Garlic bread (P)	15	0.564	98.6 ± 12.8	55.6 ± 7.2	41.1 ± 3.0	35.3
Chocolate cake (SB)	15	0.683	128.5 ± 26.0	87.8 ± 17.8	81.5 ± 12.5	7.7
Sugar cookie (P)	17	0.860	35.9 ± 6.0	30.8 ± 5.2	27.8 ± 1.9	10.8
Spaghetti w/ sauce, cheese (P)	15	0.670	385.4 ± 62.5	258.2 ± 41.9	240.3 ± 2.6	7.4
Orange juice (C)	22	1.011	124.6 ± 9.1	125.9 ± 9.2	124.0 ± 0.0	1.5

Table 2.3

Estimated weight for 19 foods items using the estimated volume and apparent density compared with the ground truth weight. The mean and standard deviation of each value is also shown. (n = number of food images that contains a particular food item)

Food name	n	Apparent Density (g/cc)	Estimated volume (cc ± SD)	Estimated weight (g ± SD)	Ground truth weight (g ± SD)	Weight per- centage error
Peach slices (P)	17	0.953	94.8 ± 31.7	90.4 ± 30.2	69.3 ± 9.9	30.4
Pear, canned halves (P)	15	1.047	80.8 ± 21.4	84.6 ± 22.4	75.6 ± 4.9	11.9
French fries (P)	17	0.241	230.0 ± 76.6	55.4 ± 18.5	70.5 ± 4.3	21.4
Ketchup (C)	17	1.141	12.7 ± 2.5	14.5 ± 2.8	15.5 ± 0.4	14.7
Lettuce (salad) (C)	15	0.316	259.9 ± 33.7	82.1 ± 10.7	48.3 ± 4.8	70.0
Margarine (C)	22	0.957	29.4 ± 6.5	28.1 ± 6.2	27.8 ± 0.6	10.8
French dressing (C)	15	1.108	32.6 ± 3.9	36.1 ± 4.3	35.7 ± 1.0	1.1
Strawberry jam (C)	22	1.307	25.4 ± 6.5	33.2 ± 8.5	21.1 ± 1.1	57.3
Coke (SC)	32	1.027	223.9 ± 13.7	229.9 ± 14.1	227.2 ± 2.3	1.2
Cheeseburger sandwich (P)	17	0.598	380.2 ± 99.0	227.3 ± 59.2	198.8 ± 11.5	14.3

[22, 49, 90] are presented in Table 2.2 and Table 2.3. Figure 2.20 shows some images acquired in this study by participants under controlled conditions. The ratio of the estimated weight to the ground truth weight is also visualized in Figure 2.22. In the tables, the apparent density is “the density of a particle including all pores (porosity) remaining in the material” [31]. The weight percentage error is obtained by $|W_e - W_g|/W_g$, where W_e is the estimated weight and W_g is the ground truth weight.

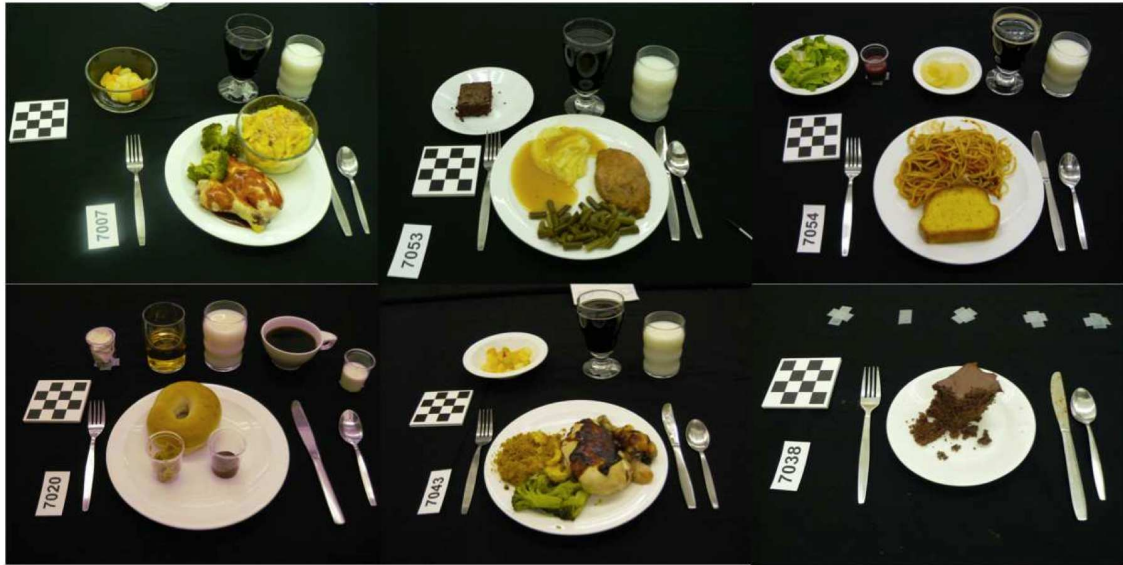


Fig. 2.20. Examples of meal images from the 24-hr controlled user study.

As shown in Table 2.2 and Table 2.3, most of the volume estimation errors for the foods and beverages which can be modeled as a conventional model (e.g. not an irregular shape) are small (less than 10%). However, the error for foods using a prismatic model method is relatively high (from 7.4% to 70%).

The reason for this is that using conventional shape models is more precise for foods with regular shapes than foods with arbitrary shape. The average error for all type of foods is 17.9%. The error is also affected by errors in the type of food density used. The food density we used is “apparent density.” However, for foods such as lettuce (salad), it will be more precise to use a density “when particles are packed or stacked in bulk including

void spaces (void fraction)” [31] namely bulk density. Overall, our result is a significant improvement when compared with our previous experimental result described in [52] as shown in Figure 2.21. The previous method using a shape template based volume estimation method is described in Section 2.3. As shown in Figure 2.21, both methods perform well on the regular shaped food items (e.g milk, orange juice, and sausage links) which can be approximated with a cylinder or square box. Our model based volume estimation method has improved the weight percentage error on all the food types except for strawberry jam and cheeseburger sandwich. Overall, our new method reduced the average weight percentage error from 63.8% to 16.8%.

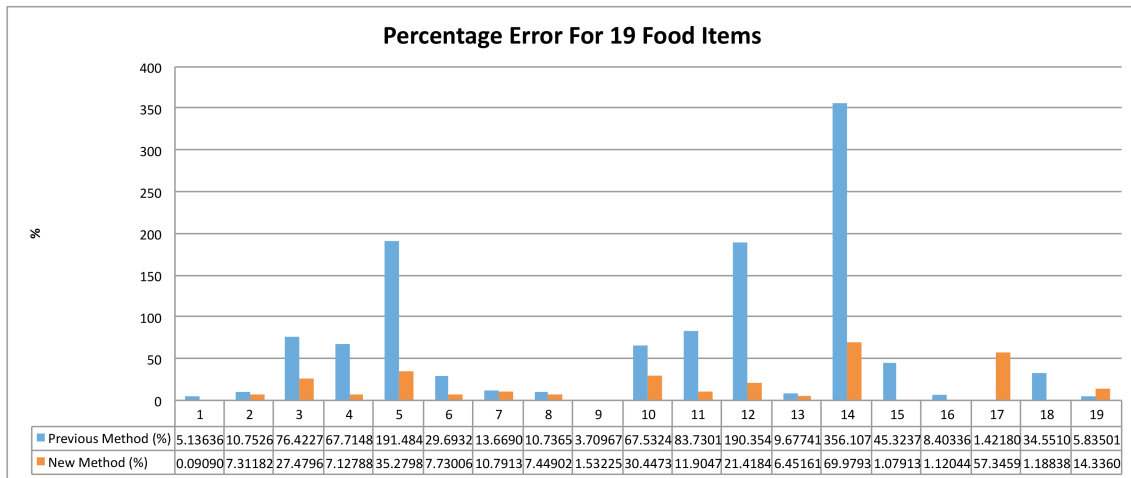


Fig. 2.21. The comparison result of our new model based method with previous template based method.

Figure 2.22 illustrates the weight error with standard deviations as the error bars. Scrambled eggs, toast, and peach slices are the food types with large standard deviation and we used the prismatic model for most of them. The reason behind this is because the prismatic model is an approximation model which is based on the assumption that the cross section of the object remains similar along the vertical direction. The area of the model is computed using the segmentation mask of the food item and the height of the model for each food item is estimated using the average height of the foods from the training images.

Therefore, the average error of the volume estimation with the prismatic model tends to be small with the standard deviation being large since the height food varies from sec to scene. Alternatively, the height of the prismatic model in the previous approached we used is adjusted with each food image by the user. Therefore the error in the height estimation is relatively lower than our automatically estimated height.

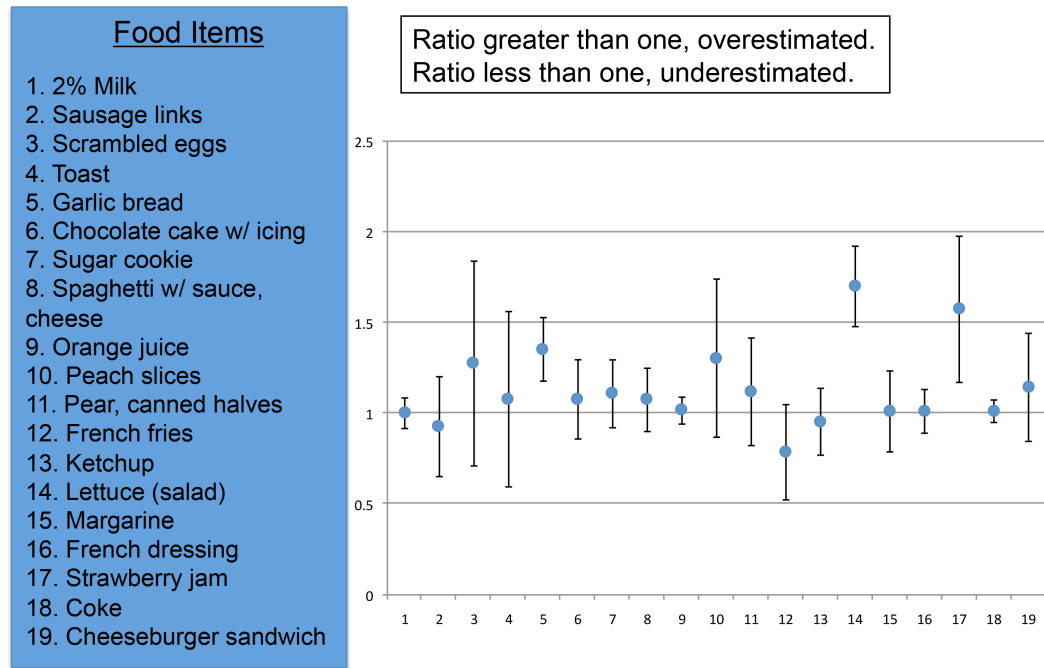


Fig. 2.22. Food weight error using automated volume analysis by food type from images taken by 15 adolescents (11-18 y) during meals over a 24-hr period. The error bars indicate the standard deviation.

2.7.2 Single-View vs. Multi-View

We also compared the single-view method with the multi-view method by conducting an experiment with five plastic food items (banana, bagel, orange juice, spaghetti, and ground beef) and one real food - Rice Krispy Treat. In this experiment automatic segmentation is used. The accuracy of the segmentation is relatively high since each meal image

contains only one food item. We are interested in comparing the model-based single-view and the multi-view volume estimation methods.

As we discussed above orange juice and Rice Krispy Treat can be reconstructed using a pre-defined shape model since they have very regular shapes (cylinder and square box). A bagel could also be considered as a regular shaped object, but due to the ambiguity of its color homogeneity, height, and depth information, it cannot be clearly distinguished. A banana has a complex shape and there is no regular 3D geometrical template that can be used from the 2D segmentation mask. Orange juice and Rice Krispy Treat are examples of foods where our geometrical model based approach can be used, whereas bagels and bananas require a more complex model for their volume reconstruction which require a pre-built 3D model. For spaghetti and ground beef, the prism model is used for single-view volume estimation due to their flat and irregular shapes.

Based on the experiments, at least 6 images taken from different view angles are needed for multi-view volume estimation. We obtained 14 to 20 images for multi-view volume estimation and acquired 35 images per food from various view angles and estimated their corresponding volume using the single-view method. The results of the estimated volume and estimation error for the food items are shown in Table 2.4 in terms of milliliters. The results are compared with a ground truth volume obtained from water displacement measurement. The estimation error is determined by $|V_e - V_g|/V_g$, where V_e is the estimated volume and V_g is the ground truth volume.

The volume estimation results for a bagel using the prior-based method are satisfactory, but our multi-view method needs further improvement. We also observed that the single-view method performs better for Rice Krispy Treat, spaghetti, and ground beef since the performance of the multi-view approach depends on the manner of image capture. If the number of side view images are not sufficient enough and cover all the angles, some voxels will not be carved away. Nevertheless, the multi-view volume estimation approach does not require any prior information (e.g. food identification and pre-defined food shape model) which single-view volume estimation relies on. Thus, the multi-view volume estimation

has the advantage over single-view volume estimation when the food type is unknown or pre-defined food shape is incorrect.

Table 2.4

Comparison of our multi-view volume estimation method in [59] and our single-view volume estimation methods for three plastic food items (banana, bagel, and orange juice) and one real food - Rice Krispy Treat. The estimation error is shown in ().

Food Item	Multi-View Method(ml)	Single-View Method(ml)	Ground Truth(ml)
Banana	180.6(5.0%)	183.9(6.9%)	172
Bagel	161.2(7.3%)	157.2(9.7%)	174
Orange Juice	179.9(10.0%)	221.3(10.7%)	200
Rice Krispy Treat	82.8(18.2%)	77.5(10.1%)	70
Spaghetti	275.2(25.1%)	242.9(10.4%)	220
Ground Beef	101.8(15.7%)	79.3(9.19%)	88

2.7.3 Single-View Volume Estimation on Free-Living Images

Our proposed single-view volume estimation method was also tested using images from a “free-living” study we conducted in 2011 [49, 90]. Examples of eating occasion images acquired in this study are shown in Figure 2.23. Sample images of each of the 56 food items are shown in Figure 2.24. Tables 2.5, 2.6, 2.7, 2.8, and 2.9 show the weight estimation results from this study. Unlike the other controlled studies we did, the food images acquired in this study were taken under natural eating conditions (also referred to “free-living” or “community-dwelling” such as home and on the go). The dataset contains a number of 315 meal images from 11 participants and 56 types of food items are observed in the study. The segmentation results used in this study are ground truth segmentation (manually drawn by a human).

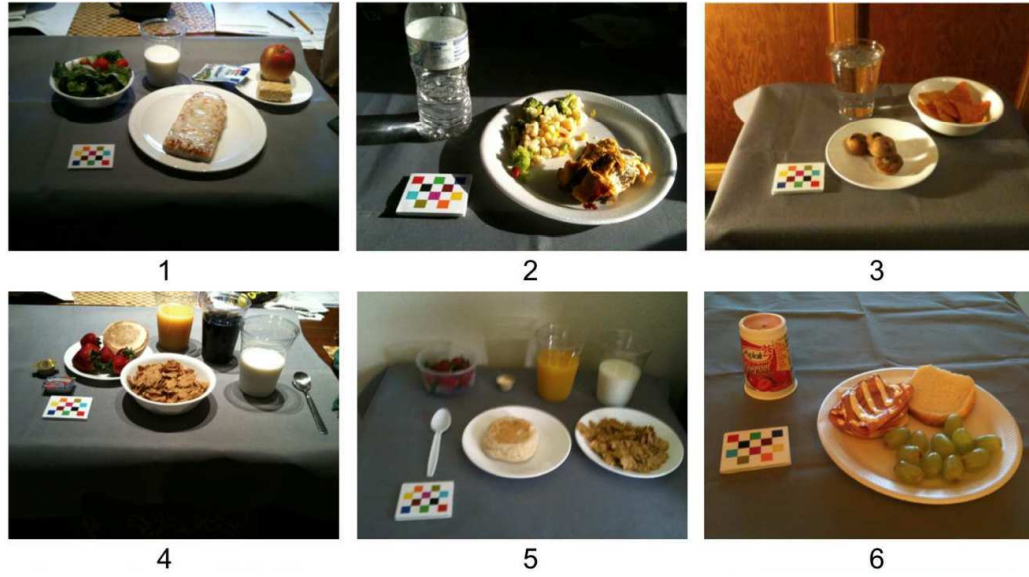


Fig. 2.23. Examples of meal images from the freelifing user study [49, 90].

The average percentage error of the weight in the experiment is define in Equation 2.29:

$$err = \frac{1}{N} \sum_n \frac{|W_E(n) - W_{GT}|}{W_{GT}}, \quad (2.29)$$

where N is the number of one type of food items, and for each food item, we divided the ground truth weight W_{GT} by the absolute difference between the estimated weight $W_E(n)$ and W_{GT} . The ground truth weight is obtained using a digital kitchen scale. Figure 2.25 shows the average percentage error and its standard derivation. The x axis is the food type, namely: 1 - apple, 2 - bagel, 3 - banana, 4 - broccoli, 5 - carrots, 6 - celery, 7 - chicken wrap, 8 - chocolate chip, 9 - clementine, 10 - cream cheese, 11 - ding dong, 12 - doritos, 13 - english muffin, 14 - frozen meal meatloaf, 15 - frozen meal turkey, 16 - fruit cocktail, 17 - garlic toast, 18 - goldfish, 19 - granola bar, 20 - grapes, 21 - ham sandwich, 22 - ice cream, 23 - jelly, 24 - lasagna, 25 - margarine, 26 - mashed potato, 27 - mayonnaise, 28 - milk, 29 - muffin, 30 - mustard, 31 - no fat dressing, 32 - noodle soup, 33 - orange, 34 - orange juice, 35 - pancake, 36 - peanut butter, 37 - pears, 38 - peas, 39 - pizza, 40 - potato chips, 41 - pretzel, 42 - pudding, 43 - ranch dressing, 44 - rice krispy bar, 45 - salad mix,

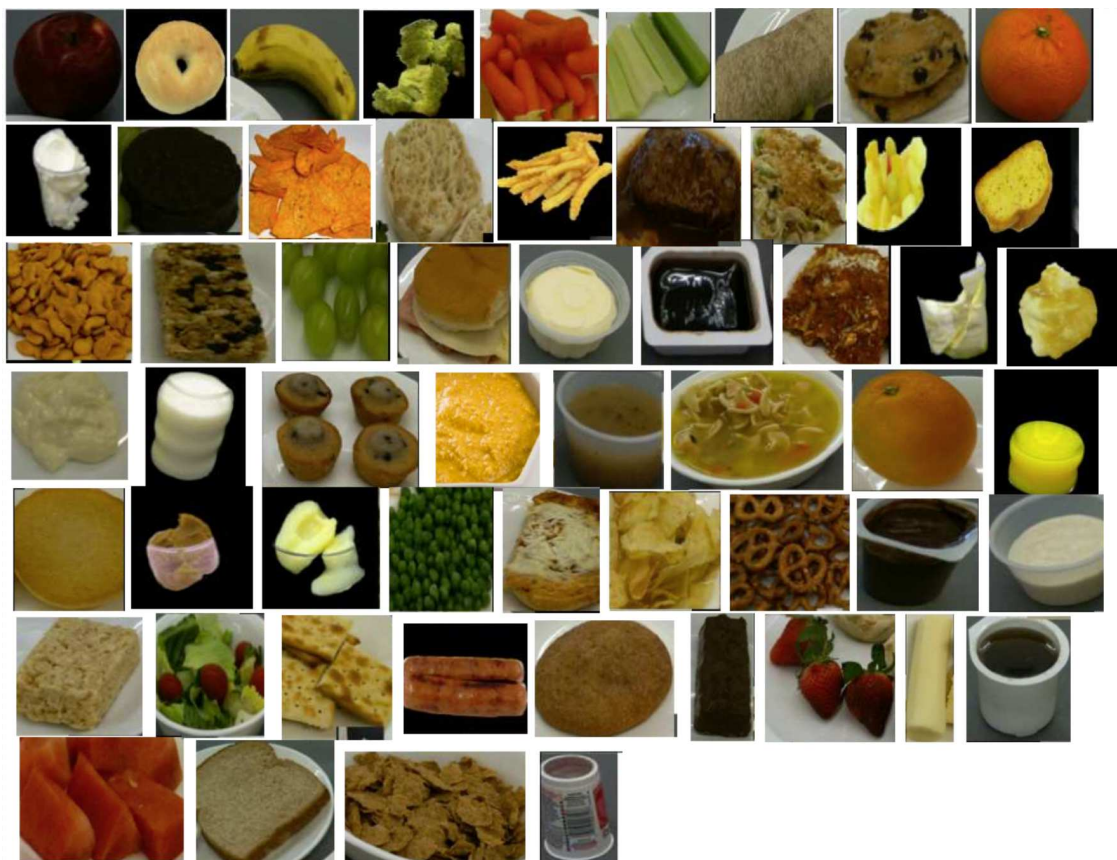
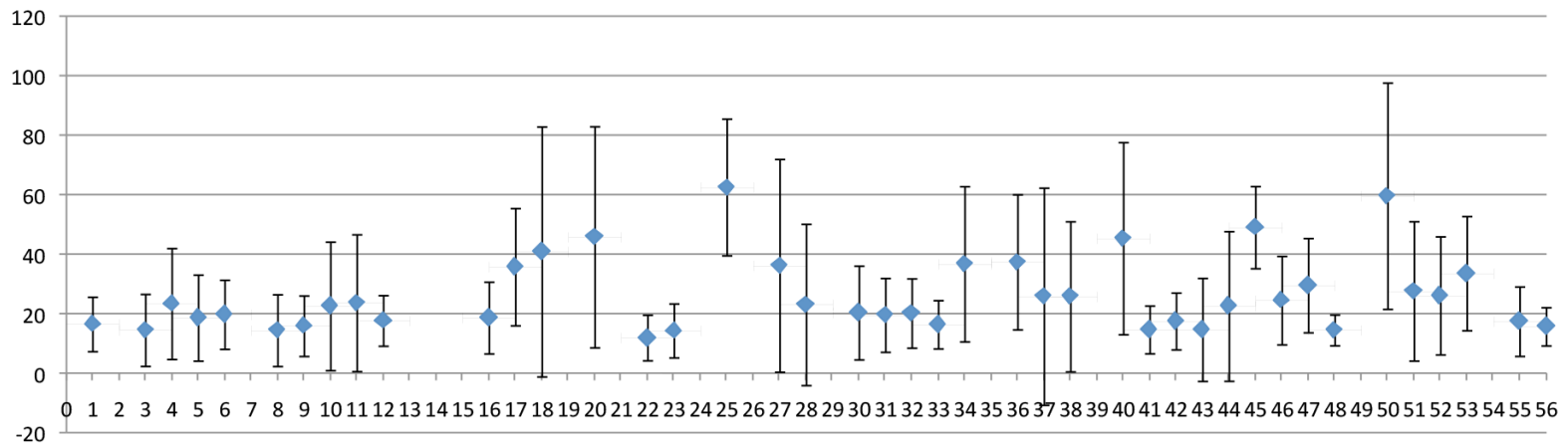
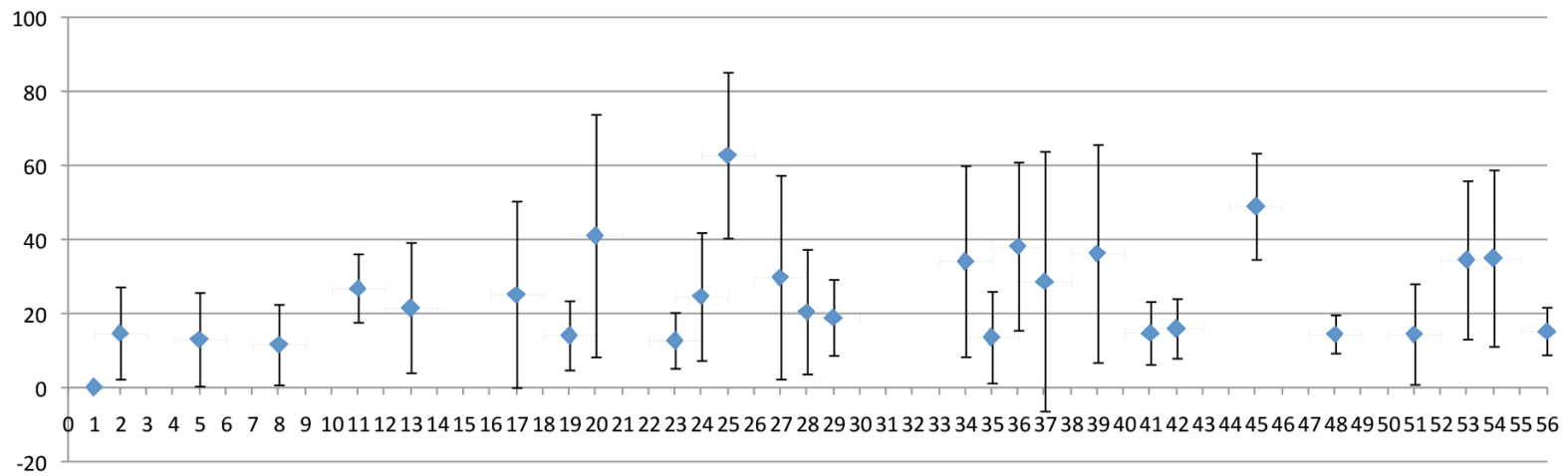


Fig. 2.24. Examples of 56 food items from the freelifving user study. (From left to right and top to bottom: apple, bagel, banana, broccoli, carrots, celery, chicken wrap, chocolate chip, clementine, cream cheese, ding dong, doritos, english muffin, frozen meal meatloaf, frozen meal turkey, fruit cocktail, garlic toast, goldfish, granola bar, grapes, ham sandwich, ice cream, jelly, lasagna, margarine, mashed potato, mayonnaise, milk, muffin, mustard, no fat dressing, noodle soup, orange, orange juice, pancake, peanut butter, pears, peas, pizza, potato chips, pretzel, pudding, ranch dressing, rice krispy bar, salad mix, saltines, sausage, snickerdoodle, snickers, strawberry, string cheese, syrup, watermelon, wheat bread, wheaties, yogurt).

46 - saltines, 47 - sausage, 48 - snickerdoodle, 49 - snickers, 50 - strawberry, 51 - string cheese, 52 - syrup, 53 - watermelon, 54 - wheat bread, 55 - wheaties, 56 - yogurt. The y axis is the percentage weight error. The average percentage error of all the food items for FNDDS density is 26.02% and for Okos density is 22.93%.



(a)



(b)

Fig. 2.25. Average percentage of weight error using automated volume analysis with FNDDS density and Okos density.

Similarly to the result in Figure 2.22, we also observed that the weight estimation with some foods (e.g. strawberry and pears) contains a small average error and a large standard deviation. The reason is the same with the test result for 19 food items experiment: For the food items which we modeled with the prismatic model (e.g. scrambled eggs, toast, and peach slices), we assumed that the cross section of the object remains similar along the vertical direction. Also, another assumption is the height for each prismatic modeled food item is fixed. Therefore, the estimated volume will have a small average error since the height is chosen as the average height. But it will have a large standard derivation for the volume estimation error, as the shape variation is quiet significant for this type of food.

2.7.4 Conclusion

We have conducted four experiments to validate our volume estimation methods, namely template vs. model based study, multi-view vs. single view, using images from a 24-hr controlled study, and using images from the free-living 2011 study. Overall, the 3D model based methods achieve an average volume estimation error of 10%. Given that portion size estimation errors of more than 50% from human observation have been reported in traditional dietary assessment methods [22, 89], our results are promising.

In summary, we proposed a single view volume estimation method to automatically estimate food portion size. In Table 2.10, the percentage weight error results from four food items using two different single-view volume estimation models for “free-living” images are shown. Table 2.10 illustrates that, for the same food item, the volume estimates can be very different with respect which pre-defined food model is used. For regular shaped and pre-built shaped food objects (e.g. milk and banana) the precise model-based method achieves a better accuracy over the prism-model method. However, for broccoli, the prism model obtains better volume estimation results since this food type has a non-rigid shape and also large variations in shape due to eating or food preparation conditions. For chocolate chips, one chocolate chip is sometimes placed on the top of another chocolate chip in

Table 2.5
Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)

food name	n	food code	FNDDS Density	Okos Density	Ave. Weight Error(F)(%)	Ave. Weight Error(O)(%)
apple (S)	14	63101000	0.6001989	NaN	26.33	NaN
bagel (C)	15	51180010	NaN	0.3013	NaN	34.58
banana (M)	12	63107010	0.7069244	NaN	14.32	NaN
broccoli (P)	11	72201232	0.9636996	NaN	24.85	NaN
carrots (P)	26	73101010	0.5663849	0.4649	18.47	12.89
celery (P)	4	75109000	0.5959722	NaN	19.58	NaN
chicken wrap (C)	10	27540300	NaN	NaN	NaN	NaN
chocolate chip (P)	23	53206000	0.5586843	0.6387	14.25	11.45
clementine (S)	5	61125010	0.7502486	NaN	15.74	NaN
cream cheese (H)	14	14301010	0.904994	NaN	22.4	NaN
ding dong (C)	14	53108200	0.4820876	0.651	23.47	26.73
doritos (P)	12	54401080	0.2324714	NaN	24.69	NaN
english muffin (C)	11	51186010	NaN	0.38	NaN	21.44

Table 2.6
Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)

food name	n	food code	FNDDS Density	Okos Density	Ave. Weight Error(F)(%)	Ave. Weight Error(O)(%)
frozen meatloaf (P)	7	28160310	NaN	NaN	NaN	NaN
frozen turkey (P)	11	28145710	NaN	NaN	NaN	NaN
fruit cocktail (C)	10	63311140	1.0228742	NaN	18.47	NaN
garlic toast (P)	37	51121040	0.2379926	0.564	51.37	14.29
goldfish (P)	6	54304000	0.2525485	NaN	32.89	NaN
granola bar (B)	5	53542100	NaN	0.671	NaN	13.94
grapes (P)	31	63123010	0.6931875	0.638	45.62	40.88
ham sandwich (C)	26	27520300	NaN	0.4204	NaN	NaN
ice cream (C)	43	13110100	0.5617883	NaN	11.78	NaN
jelly (C)	8	91401000	1.2764794	1.469	14.15	12.6
lasagna (P)	30	58301080	NaN	0.9005	NaN	24.43
margarine (P)	31	81103080	0.9675037	0.9135	30.2	26.34

Table 2.7

Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)

food name	n	food code	FNDDS Density	Okos Density	Ave. Weight Error(F)(%)	Ave. Weight Error(O)(%)
mashed potato(P)	7	71501000	0.8876181	NaN	NaN	NaN
mayonnaise (P)	10	83107000	0.9315763	1.085	36.03	29.65
milk (C)	96	11112110	1.0313277	1.01	22.89	20.35
muffin (C)	57	52302010	NaN	0.4929	NaN	18.79
mustard (P)	9	75506010	1.0355544	NaN	17.43	NaN
no fat dressing (C)	8	83205500	0.9615863	NaN	19.38	NaN
noodle soup (C)	12	28340510	1.0144207	NaN	20.01	NaN
orange (S)	40	61119010	0.735455	NaN	16.22	NaN
orange juice (C)	27	61210220	1.0520388	1.011	35.84	34.49
pancake (P)	24	55101000	NaN	0.4269	NaN	13.45
peanut butter (P)	13	42202000	1.0820487	1.166	24.59	22.83
pears(M)	4	63137140	1.0524615	1.011	12.7	12.38
peas (P)	7	75224022	0.6762805	NaN	29.32	NaN

Table 2.8
Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)

food name	n	food code	FNDDS Density	Okos Density	Ave. Weight Error(F)(%)	Ave. Weight Error(O)(%)
pizza (C)	14	58106230	NaN	0.7452	NaN	31.01
potato chips (P)	26	71201010	0.145823	NaN	36.34	NaN
pretzel (P)	4	54408010	0.1852727	0.19	14.48	14.59
pudding (C)	8	13230130	1.1031825	1.118	17.34	15.83
ranch dress- ing(C)	27	83112500	0.9937096	NaN	24.5	NaN
rice krispy bar (B)	9	53226500	0.4149615	NaN	21.08	NaN
salad mix (C)	34	75114000	0.2324714	0.2651	63.74	61.89
saltines (P)	11	54325000	0.2430383	NaN	24.34	NaN
sausage (C)	22	25221860	0.5410244	NaN	29.86	NaN
snickerdoodle (P)	10	53241500	0.359274	0.3593	14.33	14.32
snickers (B)	8	91715100	NaN	NaN	NaN	NaN
strawberry (P)	13	63223020	0.7481353	NaN	59.43	NaN

Table 2.9
Estimated percentage weight error for 56 foods items using the estimated volume and apparent density compared with the ground truth weight. (n = number of food images that contains a particular food item) (NaN - Information Not Available)

food name	n	food code	FNDDS Density	Okos Density	Ave. Weight Error(F)(%)	Ave. Weight Error(O)(%)
string cheese (C)	9	14107030	0.6467989	0.4762	27.45	14.29
syrup (C)	15	91301020	1.3863749	NaN	25.94	NaN
watermelon (P)	21	63149010	0.6838886	0.643	40.73	42.45
wheat bread (P)	16	51201010	NaN	0.2122	NaN	19.42
wheaties (P)	11	57418000	0.1268026	NaN	33.48	NaN
yogurt (C)	33	11432000	1.0355544	1.039	15.52	15.12

the meal images. Therefore, the prism-model performs badly for this type of food as the placement of the chocolate chips is not considered in prism model.

Table 2.10
Percentage weight error from different volume estimation models

	Model	Prism
Milk	22.89 + 27.1	49.49+38.49
Banana	14.32+12.1	17.31+13.79
Broccoli	32.16+15.65	23.2+18.64
chocolate chip	14.25+12.05	66.08+40.56

In this chapter, we also extended our previously reported single view volume estimation method and proposed a multi-view volume estimation method using “Shape from Silhouettes” to estimate the food portion size automatically.

Based on the experimental results, we observed that our single-view volume estimation technique not only improves the volume estimation accuracy for foods with simple shapes, but also provides a quantitative approach to estimate the volume for foods with irregular shapes.

For the multi-view volume estimation, the image sequence must be taken from different viewing angles. The intrinsic and extrinsic camera parameters need to be determined for each image. However, compared with other methods, it appears to be robust to segmentation noise. Furthermore, this approach does not require any prior shape information from the food identification and it may work for many arbitrary or unknown shaped food objects (e.g. scrambled eggs, cut carrots). In most of the experiments segmented regions of the food objects are obtained by manually drawing the outline of each food item. All of our volume estimation experiments used manual segmentation. Future work will include investigation of our volume estimation methods using automatic segmentation.

3. SEGMENTATION REFINEMENT AND VOLUME ESTIMATION

3.1 Volume Estimation In the TADA System

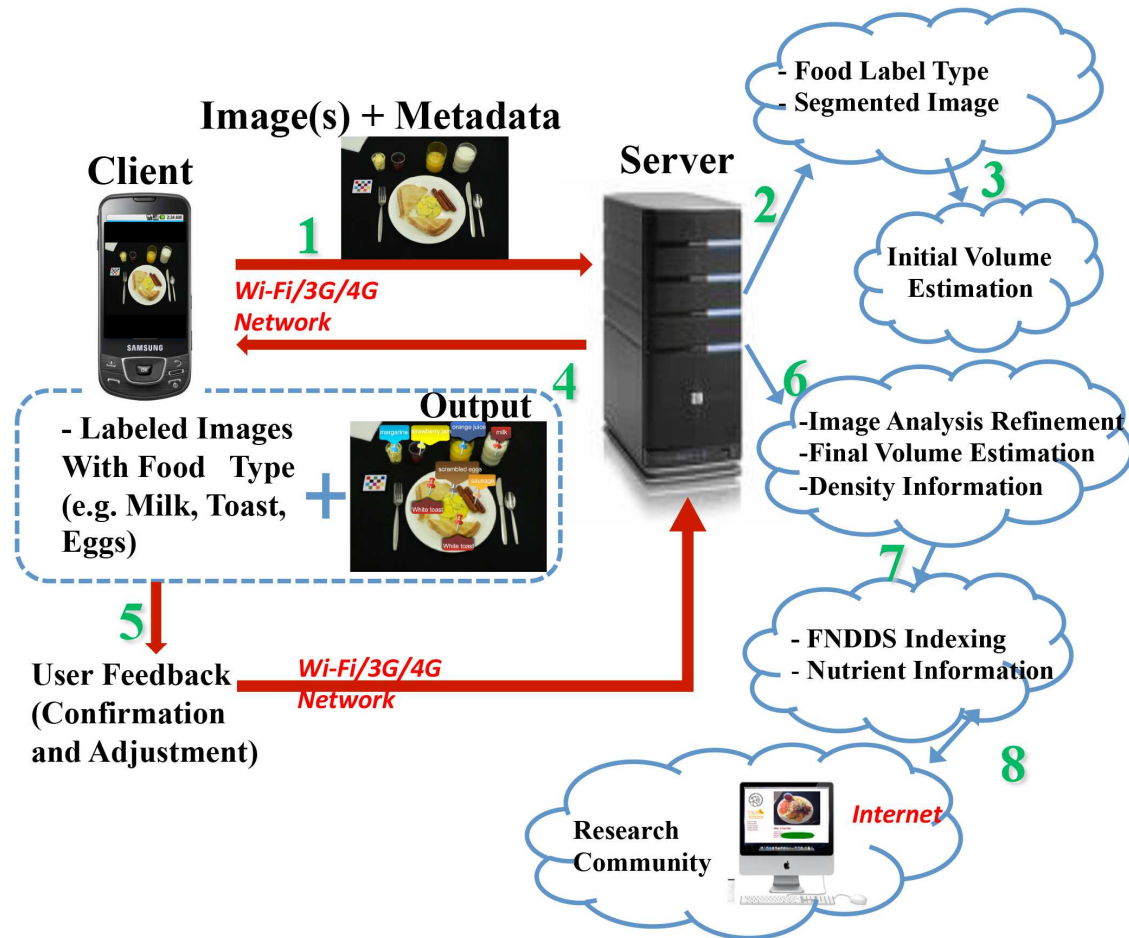


Fig. 3.1. The architecture of the TADA system.

In Chapter 2, we described our fully automatic single view and multi-view volume estimation methods. As we discussed earlier in Section 1.2 relative to Figure 1.1, the segmentation and food labels [4, 5, 9] are sent back to the user where the user confirms and/or adjusts this information (step 4). In step 5, based on the users feedback, refinements

are done only to the image segmentation and food labeling. The user's feedback is not used for volume and nutrition analysis in the previous system. In this chapter, we investigate how to adjust the volume estimation and further refine the nutrition analysis based on user feedback.

Figure 3.1 shows the modified overall architecture of our TADA system. Steps 1, 2, 3, 4, and 5 remain the same with Figure 1.1. In step 6, after the server receives the confirmed information from the user, a segmentation refinement and a food label update will be done based on the user feedback. Then, the adjustment to the volume estimation will be taken placed and weight and nutrient information will be finalized sequentially as shown in step 7. Finally the new results will be stored in the database for further analysis (step 8).

Figure 3.2 shows two major steps that are included in our volume estimation system explicitly: Step 1 is to estimate the volume using the labels and masks from the automatic segmentation and identification techniques described in [12,25,41,42,57]. This is an initial volume estimation step. Step 2 is to use the user feedback to refine the segmentation and the identification results, and this is the final volume estimation step. And as shown in Figure 3.2, the weight and nutrition information of a meal will be calculated after the final volume estimation.

An example of the user feedback is shown in Figure 3.3 [9]. The left half image in Figure 3.3 is a screen shot of the food label display and adjustment [9]. The right half image in Figure 3.3 demonstrates how a user can circle a food object on the touchscreen of a mobile device [9]. Once the server receives feedback relative to the food identification, the food labels will be updated and a segmentation refinement will be performed based on the contour drawn by the user. The weight and nutrition information will then be computed and collected for further nutrition analysis.

3.2 Segmentation Refinement with User Feedback

Automatic object segmentation and identification are difficult tasks in computer vision. In the TADA system the "review process" [9] will allow a user to fix the analysis when the

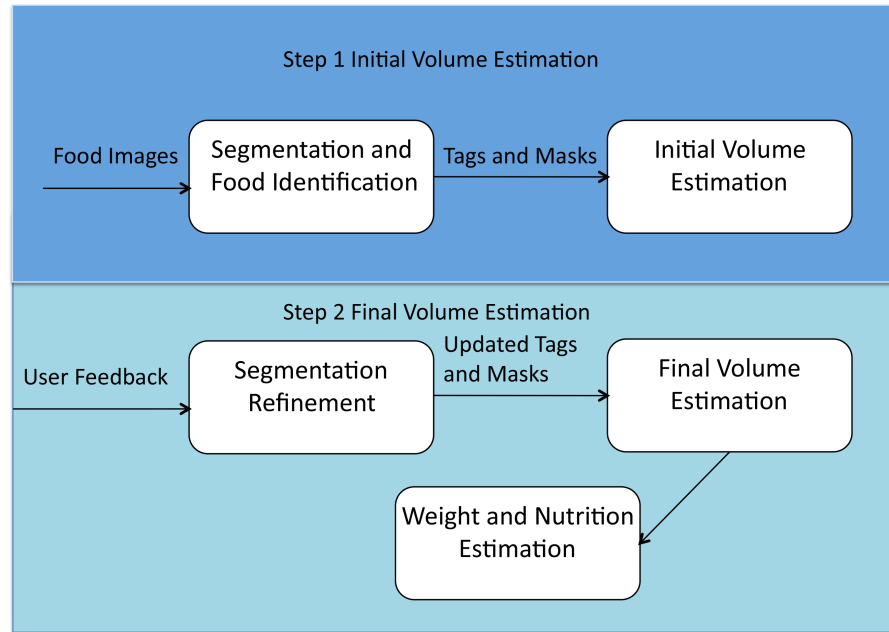


Fig. 3.2. Block diagram of the proposed volume estimation in the TADA system.

food segmentation fails to locate some food items or the identification incorrectly identifies food items. The user will be provided with a food labeling user interface - a pen tool on the review step of the application [9]. As shown in Figure 3.3, the user can provide information to assist the image analysis. The user can either change the food labeling or outline the area of the food item by circling the food region on the touch screen.

The feedback from the user consists of two things: one is an updated list of food labels and another is the center locations of the food labels from the meal image. A sequence of disconnected 2D points which are used to bound the area of a food object is also sent to the server. A contour will be generated by connecting the 2D sequence points. The initial contour is suitable to use as an input to semi-automatic segmentation.

User assisted/semi-automatic/interactive segmentation is a well known research area [91–93]. In this type of segmentation, some amount of user interaction is provided, such as outlines of the objects are drawn by the user or a seed pixel selected by the user in the image. This information is used for refinements and constraints applied to a contour that best fits the region of interest. Many applications which utilize this technique are in

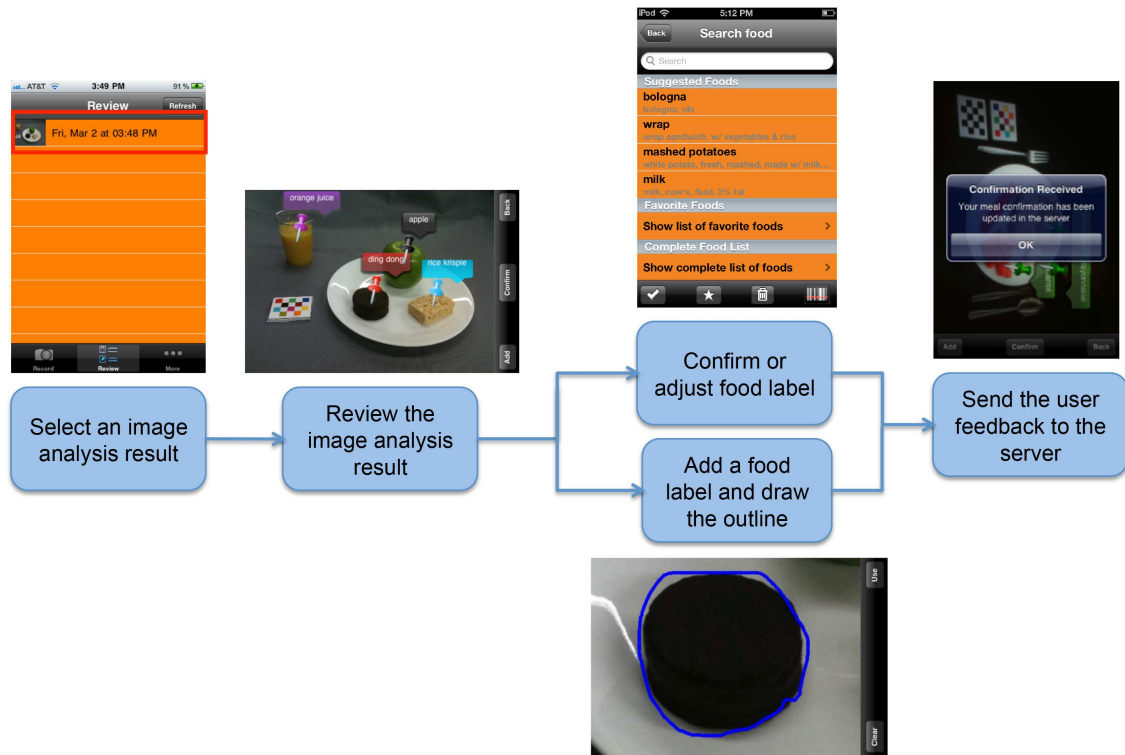


Fig. 3.3. Screen shots of the mpFR app during the review process [9].

the medical imaging area. For the case where a user selects a seed point, several region growing methods [91, 92] have been described. The essential idea of region growing is adding a neighborhood pixel to the seed region if its intensity difference is smaller than a threshold. When the initial contour is given by the user, graph cut segmentation [94] or active contour/snake [93] are two common methods.

We used active contours (snakes) [93] for semi-automatic segmentation. This technique has been used in a variety of applications in the last decade including image segmentation and motion tracking. The basic idea is to deform an initial curve to the boundary of an object under some constraints from the image.

For image segmentation, there are two types of active contour models according to the force evolving the contours, namely edge-based and region-based. For edge-based active contours, an edge detector is used to find boundaries of objects and to bound the contours to these boundaries. Instead of searching for geometrical boundaries, region-based active

contour methods usually use statistical information of the image intensity to attract the contours.

Here we use a region based approach and the basic idea of this method is to match a deformable model to an image by minimizing energy. The energy function is given in Equation 3.1:

$$E_{snake}^* = \int_0^1 E_{snake}(\mathbf{v}(s)) ds = \int_0^1 (E_{internal}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))) ds \quad (3.1)$$

The internal energy function is in Equation 3.2:

$$E_{external} = E_{image} + E_{con}, \quad (3.2)$$

where $E_{internal}$ represents the internal energy of the spline (snake) due to bending, E_{image} denotes the image forces acting on the spline and E_{con} serves as the external constraint forces introduced by the user. The combination of E_{image} and E_{con} can be represented as $E_{external}$, that denotes the external energy acting on the spline.

The internal energy function of the snake is $E_{internal} = E_{cont} + E_{curv}$, where E_{cont} denotes the energy of the snake contour and E_{curv} denotes the energy of the spline curvature. The external image energy can be represented as follows: $E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term}$.

We adopted the approach proposed in [95] by Chan and Vese, which provides a robust and accurate contour refinement when the foreground and background are statistically different and homogeneous. This method is used to detect the boundary of objects by utilizing the zero-level curve of a 3D level set function ϕ , where ϕ is defined by Equation 3.3:

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| F, \phi(0, x, y) = \phi_0(x, y), \quad (3.3)$$

where ϕ_0 defines the initial contour, and F is the curvature of the level-curve of ϕ at (x, y) . We illustrate in Figure 3.4 a zero-level set function representation of a evolving curve C in [95].

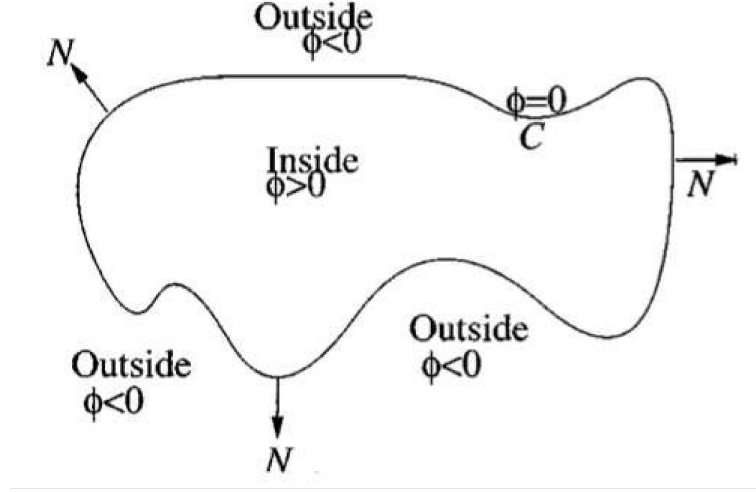


Fig. 3.4. A curve C can be represented by a level set function ϕ by propagating in normal direction (from [95]).

In [95], two constants c_1 and c_2 are also defined as:

$$c_1 = \frac{\int_{\Omega} I(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx}, \quad (3.4)$$

$$c_2 = \frac{\int_{\Omega} I(x, y) (1 - H(\phi(x, y))) dx}{\int_{\Omega} (1 - H(\phi(x, y))) dx}, \quad (3.5)$$

where $I(x, y)$ is the grayscale value at (x, y) in the image, $H(\phi)$ is the Heaviside function of the level set function, which is equal to 1 when the pixel belongs to the object and 0 otherwise.

Then, the final energy function $E(c_1, c_2, \phi)$ can be written as

$$E(c_1, c_2, \phi) = \lambda_1 \int_{\Omega} (I - c_1)^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} (I - c_2)^2 (1 - H(\phi)) dx dy + \mu \int_{\Omega} |\nabla H(\phi)| dx dy, \quad (3.6)$$

where $\lambda_1, \lambda_2 > 0$ and $\mu \geq 0$ are fixed parameters. In our work, we set $\lambda_1, \lambda_2 = 1$ and $\mu = 0.2$.

If ϕ is fixed, then c_1 and c_2 are also fixed. Therefore, the gradient descent equation for ϕ is

$$\frac{\partial \phi}{\partial t} = H'(\phi) [\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (I - c_1)^2 + \lambda (I - c_2)^2] \quad (3.7)$$

Finally, the contour can be found by minimizing the energy function in Equation 3.6 with the following steps:

- Initialize the level set function $\phi^0 = \phi_0$, and set $n = 0$.
- Estimate $c_1(\phi^k)$ and $c_2(\phi^k)$ by Equation 3.4 and 3.5.
- Solve Equation 3.7 to obtain $\phi^{(n+1)}$.
- Check if a stopping criteria for ϕ is obtained. If not, we will iteratively do the entire process and set $n = n + 1$.

The stopping criteria is often defined by checking whether the rate of change of the overall energy is smaller than a threshold or it reaches the maximum steps we defined. In our work, we simply run 300 steps to achieve an accurate contour.

Figure 3.5 shows an example of a mask resulting from the user's feedback where some foods have been added and other pins removed. The pin on the food image indicates the location each food item and a bubble that contains an abbreviated food name will be placed above it as shown in Figure 3.3.

Figure 3.5(a) is an meal image from our freeliving 2011 study. Figure 3.5(b) is the automatic segmentation result. As we can observe from the figure, the garlic bread and lasagna are segmented into several regions (over-segmentation) which can bring large error rate for the food identification and nutrition analysis. Therefore, we refine the food segmentation by utilizing the user input. The initial contours of the over-segmented objects drawn by the user are shown in Figure 3.5(c) and used as the input to Chan and Vese's method. The finalized contours after 300 iterations are shown in Figure 3.5(d) as the green lines.

Compared with traditional active contours, the Chan and Vese's method [95] does not depends on the use of edges, but relates to the specific segments of the image. Therefore,

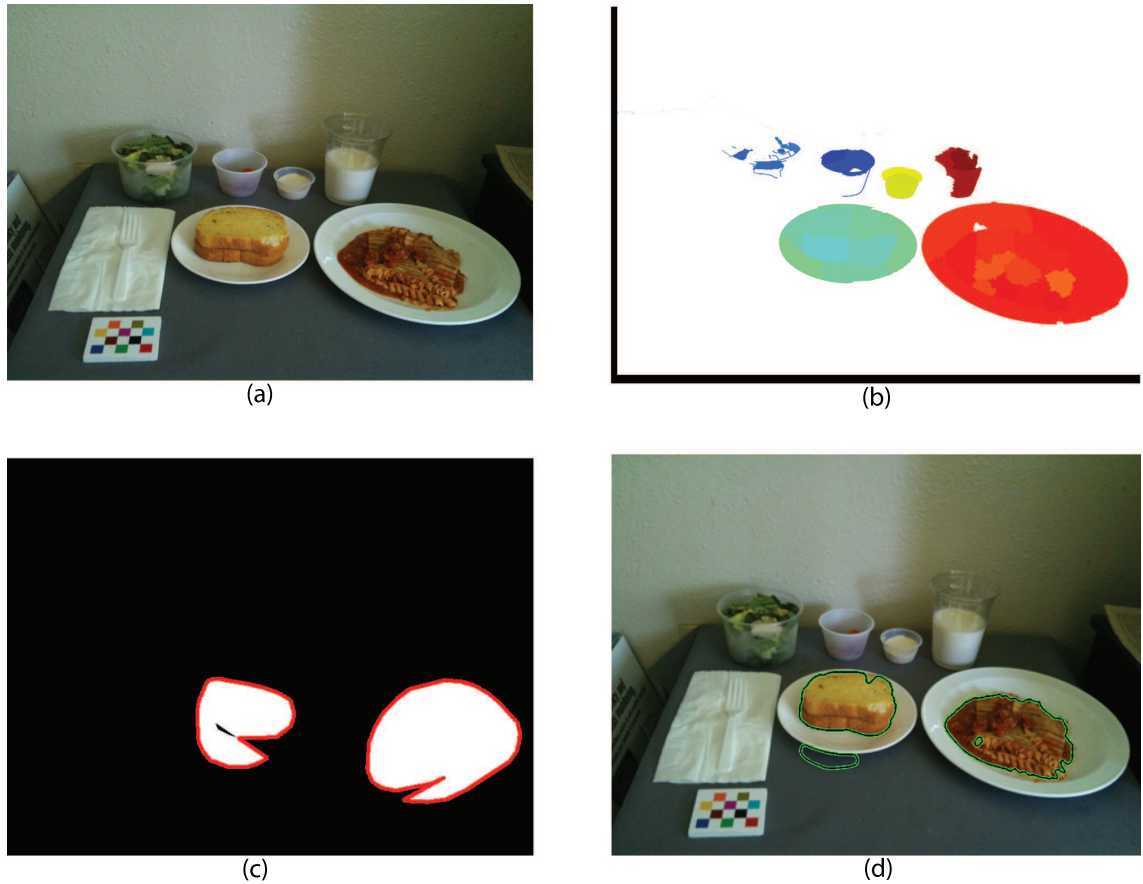


Fig. 3.5. An example of user feedback and segmentation refinement.

this method is robust to noise on the image. However, when two objects with similar colors are next to each other, the active contour method often fails. Two failure cases for active contour are shown in Figure 3.6. Figure 3.6(a) is the original meal image, Figure 3.6(b) is the initial contour, Figure 3.6(c) is the 300 iteration contour plot on the image, and Figure 3.6(d) is the final segmentation mask.

The first row images illustrate that the segmentation refinement will be wrong if the input contour provided by the user is not bounding the food object but the plate. For the second row of the images, they demonstrate that when the side part of the food item (milk) has more similar color to the table cloth than to the top part of itself, the segmentation mask will separate the top part from the side part as shown.

The failure case - over-segmentation is shown in the second row of images in Figure 3.6 as the final mask is much smaller than the initial one. Therefore, further study on segmentation refinement is needed to improve the segmentation accuracy.

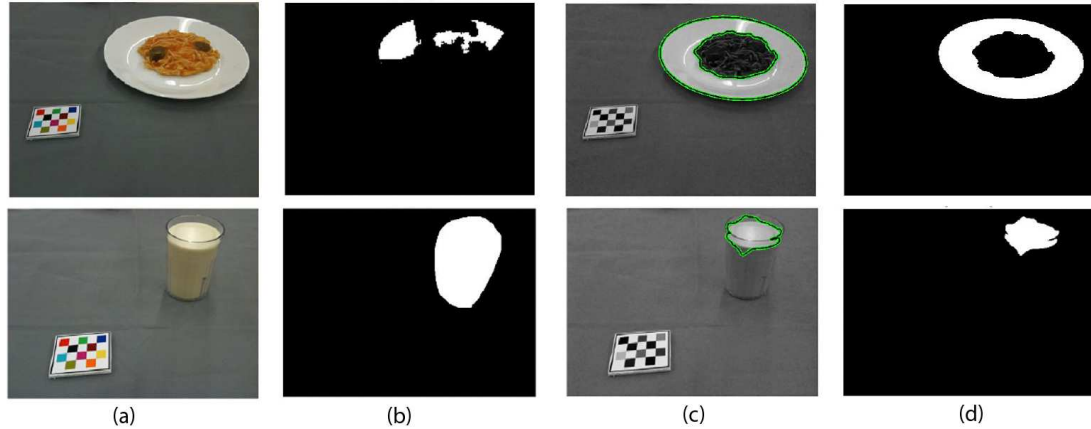


Fig. 3.6. Examples of the failure cases using active contours.

After the segmentation refinement, the final volume will be estimated from the refined segmentation using active contours.

3.3 Food Density and Nutrition

In our TADA system, the food weight can be simply obtained by multiplying the volume estimation we acquired from the meal images with the specific food density as shown in Equation 3.8. The food volume can be estimated using our model-based method introduced in Chapter 2. Food density is not yet available for all foods in the FNDDS [28] database. A group in our TADA team are working on developing several density measurement techniques [30, 31].

Theoretically, food density can be obtained by measuring the volume and the weight of the food item, as shown in Equation 3.8. However, the volume of the food can be measured in different ways, such as Solid Displacement, Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and using a 3-Dimensional (3D) Scanner [31]. Density for the same type of food can be different because of the variation in the food item (e.g. food place-

ment, appearance, or food preparation). Therefore, various ways for density measurement should be used in the TADA system.

$$\text{food weight} = \text{estimated volume} \times \text{food density}, \quad (3.8)$$

Based on the form of the food object materials, there are typically three type of density measures: true density, apparent density and bulk density [96,97]. The three different types of densities are categorized based on different methods used to measure volume.

When the pure material (without considering pores or void spaces) obtained from a food item is used in the density measure, this is known as “true density.” It is obtained using Equation 3.9 [98]:

$$\rho_t = \frac{m_d + m_w}{V_s + V_w}, \quad (3.9)$$

where m_d is dry solid weight, and m_w is water weight. V_s is solid volume, V_w is water volume, V_a .

“Apparent density” is the density of a single piece food including all pores (porosity) remaining in the material. The apparent density can be computed using Equation 3.10:

$$\rho_t = \frac{m_d + m_w}{V_s + V_w + V_i}, \quad (3.10)$$

where m_d is dry solid weight, and m_w is water weight. V_s is solid volume, V_w is water volume, and V_a is air volume of all pores inside the material.

When pieces are packed or stacked in bulk including void spaces between each particle, it is known as “bulk density.” The volume measurement of this type density can be estimated using a container (e.g. measuring cylinder) filled with a mass of particles. Bulk density is shown in Equation 3.11:

$$\rho_t = \frac{m_d + m_w}{V_s + V_w + V_i + V_o}, \quad (3.11)$$

where m_d is dry solid weight, and m_w is water weight. V_s is solid volume, V_w is water volume, V_i is air volume of all pores inside the material, and V_o is the air volume of all the

pores outside individual particles. Figure 3.7 illustrates an example of these three distinct food density measurements.

In [30, 31], six different food volume to density measurement methods are introduced. The “Gas Pycnometer” method is used to measure true density. “Solid Displacement Method-Rapeseed,” “Computed Tomography,” “Magnetic Resonance Imaging” and “Laser Scanner” methods are used to measure apparent density. And “Artificial Neural Network” methods can be used to measure “Apparent density” or “Bulk density.”

In our study, most foods densities are measured by “Apparent density” or “Bulk density,” since they are a more natural status for food items. In addition, a FNDDS density neural network model was generated using existing data in the FNDDS to obtain a total proximate analysis. The computation of these density measurements is done by our TADA team members, a group lead by Professor Martin Okos of Purdue University [31] and further description of this work is beyond the scope of this thesis.



Fig. 3.7. Example of true, apparent and bulk density of the same food (puffed cornflour pellets) (from [31]).

The nutrition information of the foods in a meal image is finalized by using the estimated volume with the food density as indicated in Equation 3.8. After the volume is estimated using the methods described in Chapter 2, it will be inserted into the TADA database system and the relative density information will be associated with it.

Our system generates large amounts of data that needs to be stored and indexed in a database. We have developed three unique databases to address our needs [29, 99]. The data are organized around three key elements: images, foods, and users. Based on these elements our database system is composed of three logically interrelated databases namely

I-TADA, T-FNDDS, and E-TADA as shown in Figure 3.8. The I-TADA database contains information related to the food image. The T-FNDDS database is an extension of the original USDA FNDDS [100] by adding visual descriptions that can be used for image analysis and other information associated with each food item such as barcode data. Finally, the E-TADA database stores information available for each user and data related to our user validation studies. The density, volume, and weight data will be stored in the I-TADA database.

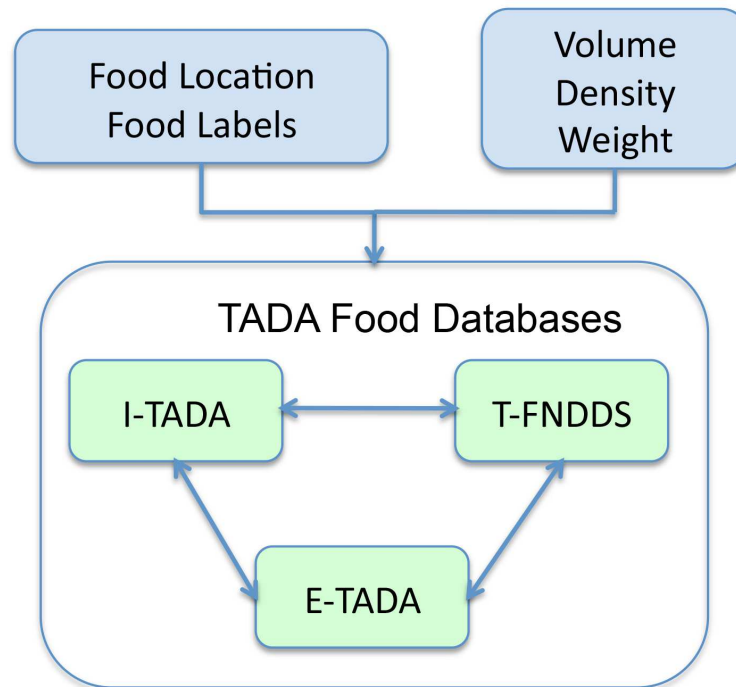


Fig. 3.8. The TADA food databases.

Figure 3.9 illustrates an example of the volume, weight, and density information available in the I-TADA database for a particular image. The first column in Figure 3.9 is the type of the food, and the second column shows which model was used for the volume estimation. The location of the food item is represented by the center of the food segmentation mask. There are two types of density we are using: Okos density [31] and FNDDS density. The weight and energy of the food are estimated by multiplying the estimated volume with one of the density data. The total nutrition value for the Okos density and FNDDS density

is computed by adding up all the nutrition values of the food items as shown in the second table in Figure 3.9. As shown in Figure 3.9, if we click on the weight values presented in red color font, the page contains all the nutrient values for that particular portion of the food will be displayed as shown in Figure 3.10. In this example, all the nutrient values for 185 gram of milk are computed based on Equation 3.12:

$$nvpp = nvpu/100 \times \text{portion weight}, \quad (3.12)$$

where $nvpp$ is the nutrient value per portion, and $nvpu$ is nutrient value per 100 gram.

3.4 Experimental Results and Conclusion

The evaluation of our proposed volume estimation refinement method was done by conducting an experiment with six plastic food items under controlled conditions. The single-view volume estimation method in Section 2.4 is used in this experiment with 10 to 15 food images acquired for each food type. The results are shown in Table 3.1. We compared the refined volume estimation in column 5 and initial volume estimation in column 3 in Table 3.1. The refined volume estimation uses the active contour segmentation mask based on user feedback and the initial volume estimation uses the fully automatic segmentation.

We tested our method on six plastic food items: orange juice, banana, tomato juice, pear, and jello. Here we used water displacement method [31] to find the volume of the plastic foods. The percentage error of volume estimation for orange juice dropped from 23.5% to 8%, for tomato juice it reduced from 10.8% to 5.0%, for spaghetti it is from 48.1% to 10.4%, and for banana it is from 13.9% to 6.4%. These results were improved because the segmentation mask is more accurate with segmentation refinement compared with the initial segmentation mask.

For foods which utilized the prismatic model, e.g. spaghetti, the percentage error is reduced significantly since the estimated volume using the prismatic model is highly dependent on the area of the segmentation. Therefore, if we can provide more accurate seg-

Volume Analysis Using User Feedback

Food Items	Volume Method	Volume(cc)	Segment Center	Okos Density(g/cc)	Weight(g)	Energy	FNDDS Density(g/cc)	Weight(g)	Energy
bologna	prism	73	(1097,997)	N/A	N/A	N/A	N/A	N/A	N/A
pork chop	prism	123	(1539,973)	N/A	N/A	N/A	N/A	N/A	N/A
pear	prism	72	(880,492)	N/A	N/A	N/A	N/A	N/A	N/A
milk	cylinder	183	(1833,389)	1.01	185	92.5	1.031	189	94.5
orange juice	cylinder	172	(1575,545)	1.011	174	73.08	1.052	181	76.02

Nutrient Description	Total Nutrient Value (Okos)	Total Nutrient Value (FNDDS)
Protein	7.1316	7.3049
Total Fat	3.8881	3.9767
Carbohydrate	25.797	26.6737
Energy	165.58	170.52
Alcohol	0.0	0.0
Water	320.1379	329.9418
Caffeine	0.0	0.0
Theobromine	0.0	0.0
Sugars, total	23.977	24.7674
Fiber, total dietary	0.348	0.362
Calcium	230.37	235.61

Fig. 3.9. An example of the volume, weight, and density information available in I-TADA for a particular image.

mentation mask, we can obtain more accurate volume estimation as well, as the volume estimation method greatly depends on the segmentation mask. For pear and jello, the segmentation masks from the refinement are almost the same with the initial segmentation result, therefore, the volume estimation results based on the segmentation didn't change either.

Milk, cow's, fluid, 2% fat

Main Food Description: Milk, cow's, fluid, 2% fat

Food Code: 11112110

Additional Food Description: Hi-Protein milk

Portion Weight: 185 GM

Portion Description: NONE

VCM: Cooming soon

UPC: N/A

Nutrient Description	Nutrient Value per 100g	Nutrient Value per Portion
Protein	3.3	6.105
Total Fat	1.97	3.6445
Carbohydrate	4.68	8.658
Energy	50.0	92.5
Alcohol	0.0	0.0
Water	89.33	165.2605
Caffeine	0.0	0.0
Theobromine	0.0	0.0
Sugars, total	5.06	9.361
Fiber, total dietary	0.0	0.0
Calcium	117.0	216.45
Iron	0.03	0.0555
Magnesium	11.0	20.35
Phosphorus	94.0	173.9
Potassium	150.0	277.5
Sodium	41.0	75.85
Zinc	0.43	0.7955
Copper	0.012	0.0222
Selenium	2.5	4.625
Retinol	55.0	101.75

Fig. 3.10. An example of all the nutrient values for a particular portion of a food item in FNDDS database.

In conclusion, we proposed a refinement for volume estimation that utilizes user input from our mpFR system. Based on the experimental results, we observe that volume esti-

Table 3.1
Estimated volume for six plastic foods items using user feedback

Food Name	Ground Truth Volume (cc)	Initial Volume (cc)	Percentage Error (%)	Refined Volume (cc)	Percentage Error (%)
Orange juice	180	153	23.5	184	8.0
Banana	172	196	13.9	183	6.4
Tomato juice	120	107	10.8	114	5.0
Spaghetti	260	385	48.1	233	10.4
Pear	140	166	18.6	166	18.6
Jello	120	152	26.7	152	26.7

mation refinement increases the accuracy of food identification by utilizing user correction. It also improves the volume estimation by using a refined segmentation mask for most of foods in the experiments.

4. IMAGE QUALITY MEASURES AND COLOR CORRECTION

4.1 Overview of Image Quality Measures

Image quality assessment is important in that one can identify and quantify image degradation in an imaging system in order to maintain consistent visual appearance and user experience.

The loss of quality can be introduced during image acquisition (e.g. out of focus, over exposure, and motion blur), image processing (e.g. compression artifacts, color transform, and changes in resolution) or reproduction/rendering (e.g. limitations of the display device). In our work we will focus on image quality assessment during image acquisition.

The traditional and widely used method for image quality assessment is to have human subjects evaluate the images. Two subjective perceptual methods are “triplet comparison” and “quality ruler” [101]. Subjective methods are known to be cumbersome, time-consuming, and not suitable for real-time application. Many computational models have been proposed to estimate image quality (IQ). A successful model should match close to human performance.

Automatic IQ measures can be classified into: full-reference [102], reduced reference [103], and no-reference methods [104]. A full-reference measure is used to evaluate the similarity between a original/reference image and a image with some quality issues. In [102], an objective method for full-reference image quality is proposed. It is based on the degradation of similarity structures and has demonstrated improvements over other traditional methods since it takes into account human visual system (HSV). This method is known as the Structural Similarity Index (SSIM) and it is also widely used for measuring the similarity between two digital images.

No-reference measures do not use any information about the reference image. A reduced reference measure is used when only partial information related to the reference

image is known. Usually the use of a no-reference measure is a difficult task since the result of image degradation is influenced by the content of the image and the type of distortion. Therefore, most no-reference measures are designed to assess a single defect, i.e. noise, blur, or ringing artifacts. If there is more than one distortion in the image, a summative measure can be deployed to account for all the defects and their interactions. This is the approach we will take here.

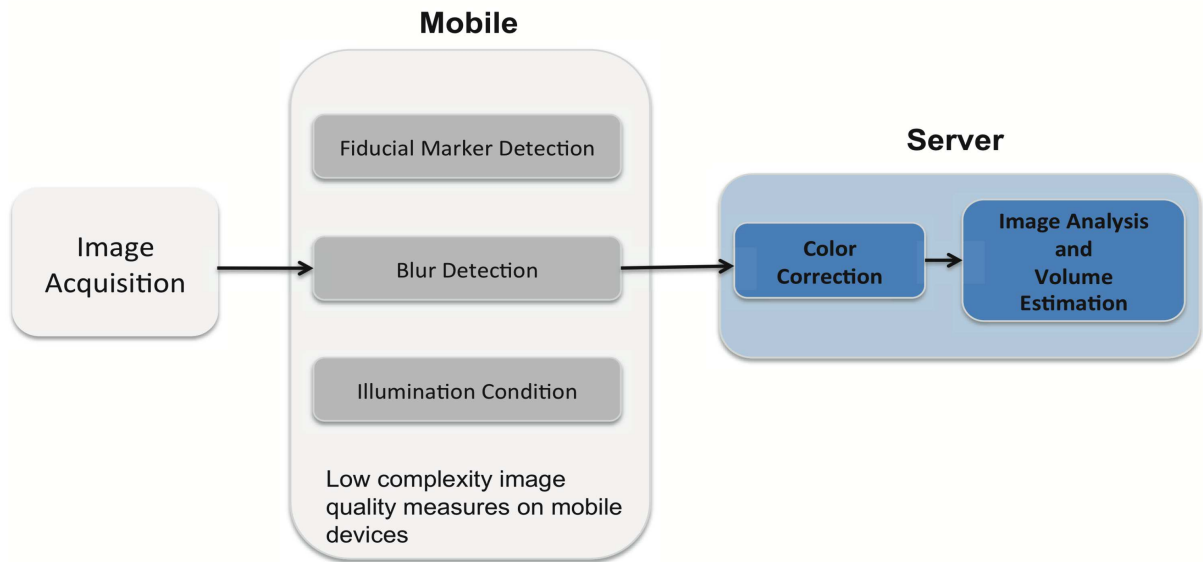


Fig. 4.1. Our proposed quality measure and image enhancement system.

In this thesis, we will first discuss the image distortion in our application and propose several specific image quality measures. As shown in Figure 4.1, our image quality assessment system begins with capturing a food image with the color fiducial marker using a digital still camera or a mobile device camera. The next step is fiducial marker detection that will be described in Section 4.2. After extracting 11 colors from the fiducial marker, we then evaluate the illumination condition by determining if the colors in the fiducial marker fall within a prescribed range of RGB values. The reason for the illumination check is to avoid bad illumination conditions. Specifically conditions that are too dark, too bright or have incorrect color temperature. We can also use this to assist the user in capturing

a better image by providing feedback to them [9]. Since the mobile device has limited computational power and memory resources the color correction step is done on the server.

4.2 Fiducial Marker Detection

The problems encountered in designing the mobile device food record (mdFR) are fundamentally difficult; however it is possible to solve these problems by properly utilizing contextual information. For example, the TADA fiducial marker, shown in Figure 4.2, provides a reference for spatial camera calibration as well as color correction, and thus makes it feasible to use the mdFR in a wider range of eating occasions.

The use of a planar black and white checkerboard pattern with known geometry for camera calibration has been reported in the literature [60]. Its alternating black and white grids produce the basic crossing feature points which make it easy to detect it under varying imaging conditions.

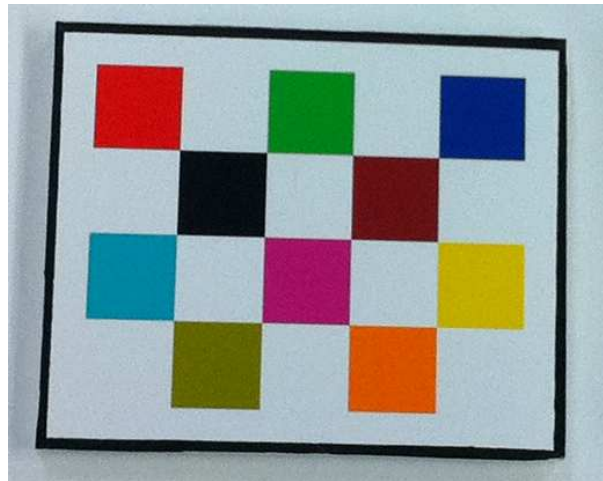


Fig. 4.2. The color fiducial marker used in the TADA system.

In the TADA system [8], we are interested in using the fiducial marker both for camera calibration (geometry reference) and color correction (color reference).

Since this fiducial marker is intended to be used with a mobile application users need to carry these along with their mobile telephones. Therefore, the fiducial marker needs to

be small, preferably credit card size and at the same time, it needs to be large enough to provide the geometry and color information. Currently, we use a fiducial marker of size $7 \times 6 \text{ cm}^2$, which typically corresponds to approximately 280×240 pixels in a 2048×1536 image captured using our application on the iPhone. To provide enough information for good color correction, the fiducial marker should have a wide range of colors. We have chosen all three primaries and some other frequently occurring colors for various squares in the marker (Figure 4.2).

The current version has ten different colors on a white background. It also has a black border for quick identification. These specific characteristics of our fiducial marker provide us more contextual information but they also prevent application of traditional methods for checker board detection [105–107].

In [108, 109] checkerboard detection is done by finding edges and fitting lines to the edges. The corners are detected as the intersection of straight lines fitted to each square. This method works well for high contrast images captured under good illumination. But under insufficient illumination, it may not find the entire checkerboard.

Another approach is to directly detect the corners of the checkerboard. Existing checkerboard corner detection methods [110, 111] generally use a refined Harris or Susan corner finder to locate all possible corner points and then use distance or neighbor constraints to group the corners. This step is followed by matching the checkerboard pattern.

Existing methods for checkerboard detection often fail for our color checkerboard because the color design possesses less contrast (such as yellow on a white background under yellow light) than a black and white checkerboard. In addition, most of these methods use a global adaptive threshold for converting a gray-level version of a color image to a binary image, thus the brightness of the image can affect the ability of these methods to recognize the corners or the lines.

Therefore a robust method for color checkerboard detection in unconstrained environments is required.

For the images captured using our application to be useful for further image analysis we need to automatically detect the presence of the fiducial marker. These methods need to

be implemented on the mobile device and provide immediate feedback to the user before saving the image [9].

Our proposed approach reduces the cost for searching for the corners of the checkerboard and speeds up image pre-processing. To achieve this, we made modifications to existing methods [112] by utilizing the special characteristics of our fiducial marker such as its black border. Our checkerboard pattern detection is based on a region search method which is less sensitive to illumination changes and noise. It consists of two steps as shown in Figure 4.3, the first step is to detect possible candidate regions for the presence of the checkerboard and the second step is to identify the checkerboard pattern and find the corners of each color patch using quadrangles. The detail of each step will be introduced in the following sections.

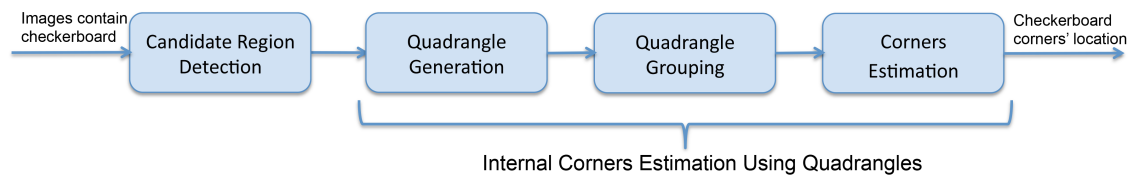


Fig. 4.3. Overview of our proposed checkerboard detection method.

4.2.1 Candidate Region Detection

The purpose of candidate region detection is to reduce the time complexity. Instead of doing an exhaustive search on the entire image for the checkerboard, we use few binary operation to narrow the possible candidate regions for the checkerboard.

One of the distinctive characteristics of our fiducial marker (FM) is that it has a black border on a white background. Thus, any quadrilateral region enclosed by a black border is a possible candidate region. Black borders can be easily identified in a binary image obtained by using a global adaptive threshold to the color image.

Two major steps are involved in this process namely: checkerboard candidate region detection and internal corners estimation using quadrangles. Accurate checkerboard corner

detection is relatively time consuming, therefore, a candidate region detection is used in order to improve checkerboard detection in terms of complexity. The checkerboard candidate region detection process is shown in Figure 4.4. Initially, a fixed global threshold is used on a gray scale version of the color food image. After connected component analysis on black pixels of the binarized image, we need to reject the components with very small areas.

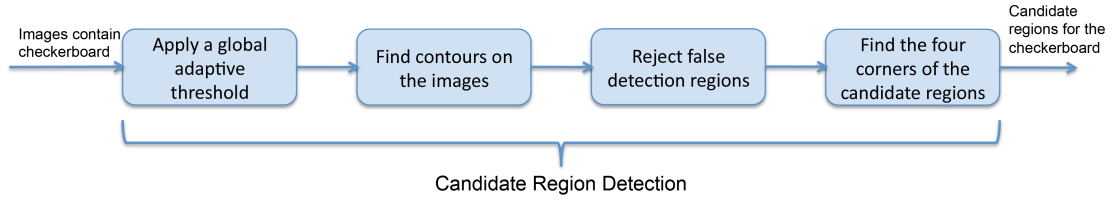


Fig. 4.4. Checkerboard candidate region detection.

For a meal image to have sufficient resolution for automatic dietary assessment the area of the fiducial marker region should be neither too large nor too small. From Equation 4.1:

$$FA(R) = \frac{\text{number of pixels in } R}{\text{Total number of pixels on the image}}, \quad (4.1)$$

where R is a checkerboard candidate region.

$$\text{Candidate regions} = \{R | \alpha < FA(R) < \beta\}, \quad (4.2)$$

The ratio factor FA is computed for each possible checkerboard region. FA is the number of pixels in a candidate checkerboard region scaled by the total number of pixel in the entire image. By our definition, the checkerboard should not be too small in a good food image, as it will be difficult to detect the checkerboard and also a cluttered background could be included in the image if we took the meal image from a very far distance. The checkerboard should not occupy too much space in the image since parts of meal could be occluded. Therefore, we reject the regions which have areas (number of pixels) smaller

than $\alpha = 1\%$ or larger than $\beta = 9\%$ of the total number of pixels in the image as shown in Equation 4.2.

To segment the image into different regions, a connected components labeling technique [113] also referred to as region growing is used. Connected components labeling is used to label a set of pixels that form a connected group. For example, three connected components are extracted in a binary image using 4-connectivity region growing method as shown in Figure 4.5. Then, each region will be labeled with a unique label number. The binary image of our food image can be obtained by converting the grayscale image with a fix threshold of 155 in a 8 bit image.

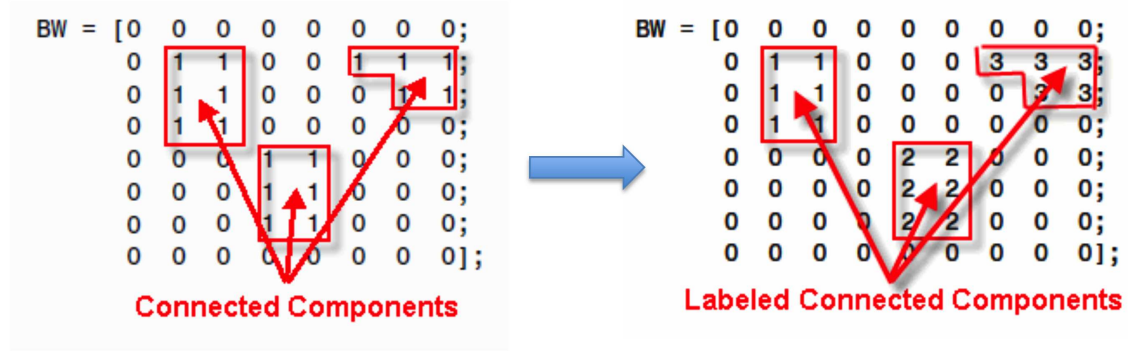


Fig. 4.5. An demonstration of connected components labeling (from [113]).

We demonstrate our candidate region detection method with some examples. In Figure 4.6, two original food images are shown in the first row. The first food image has a relatively clear background compared with the second food image which has a clutter background that even contains some wires and a keyboard. The second row (b) set of images demonstrate the connected components labeling results for those two meal images. As shown in Figure 4.6(b), each connected component is labeled with an unique color.

In Figure 4.6(c), regions R with $FA(R) < \alpha$, where $\alpha = 0.01$, are removed from the candidate list. And in Figure 4.6(c), regions R with $FA(R) > \beta$, where $\beta = 0.09$, are discarded from the list of candidate regions. Finally, the remaining regions in the candidate list will be considered as possible checkerboard location and checkerboard corners extraction will be performed in these regions. By first doing candidate region detection, we are

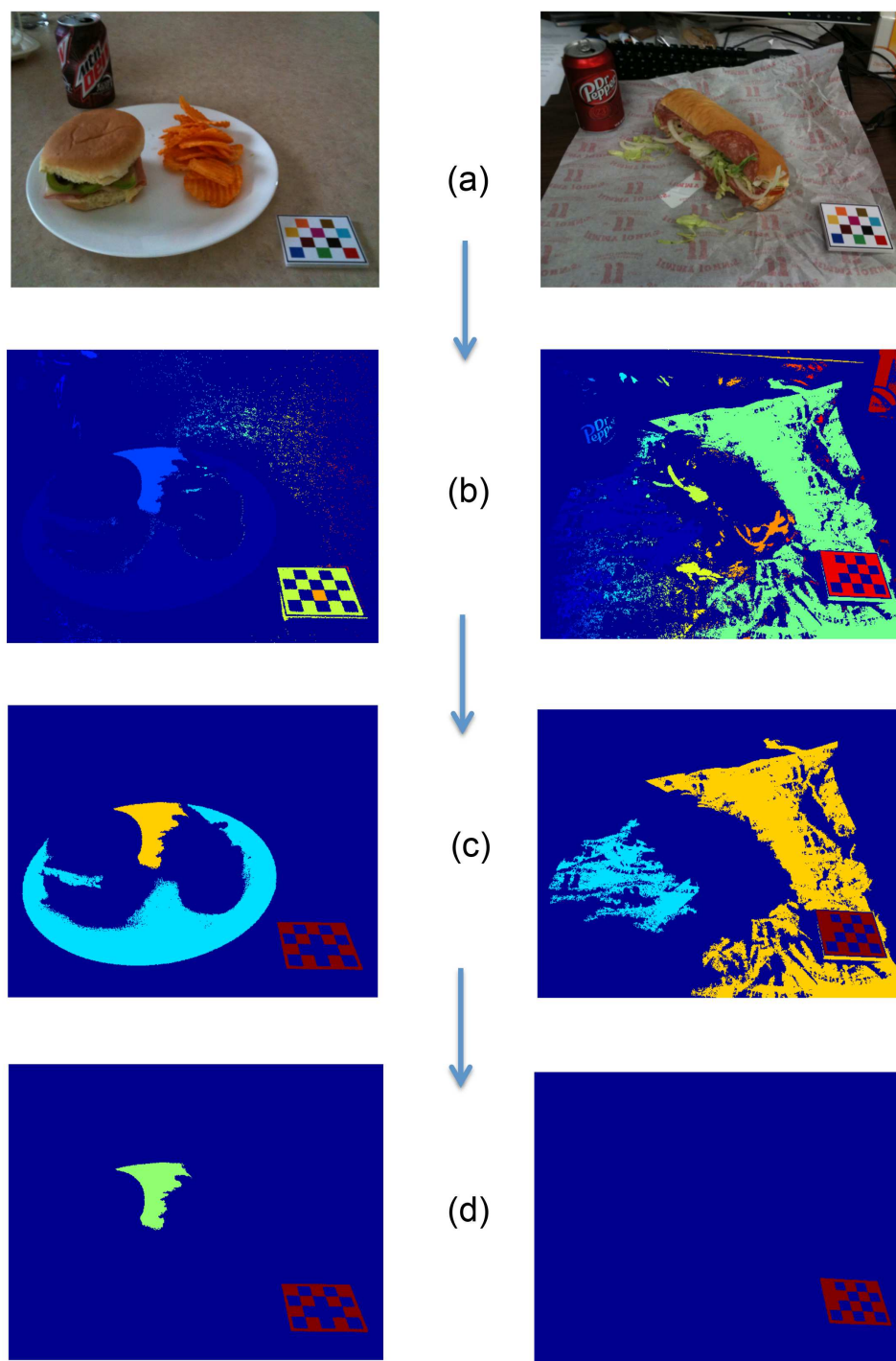


Fig. 4.6. Examples of checkerboard candidate region detection.

able to reduce the checkerboard detection process from 10 seconds to 1-2 seconds for a 2048×1536 sized image on an iPhone 3GS.

4.2.2 Internal Corners Estimation Using Quadrangles

After finding the candidate regions, the next step is checkerboard matching by detecting the internal corners. This is a four steps process consisting of image dilation, quadrangle generation, quadrangle grouping and corner estimation as shown in Figure 4.7. We will introduce each step in detail in the following sections. It should be noted that this series of operations is independently performed on each of the candidate regions of interest.

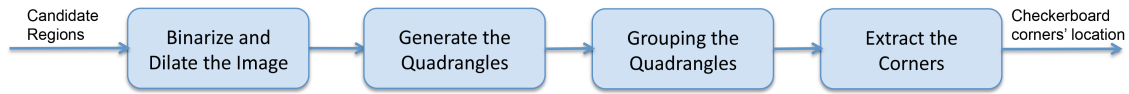


Fig. 4.7. Internal corners estimation using quadrangles.

Quadrangle Generation

First we obtain a binary version of the colored checkerboard candidate region as shown in the second image of Figure 4.8. The regions are detected using the method discussed in Section 4.2.1.

This cannot be done by using the same threshold $T = 155$ used in candidate region detection, because our fiducial marker contains low contrast boundaries such as a yellow square on a white background. And if we use the threshold $T = 155$, the color patches in the checkerboard will not be separated from the white background in some circumstances such as low or high lighting environment. Therefore, a local threshold for this candidate region should be used. We computed the local threshold using Equation 4.3:

$$T_l = \frac{\max + \min}{2}, \quad (4.3)$$

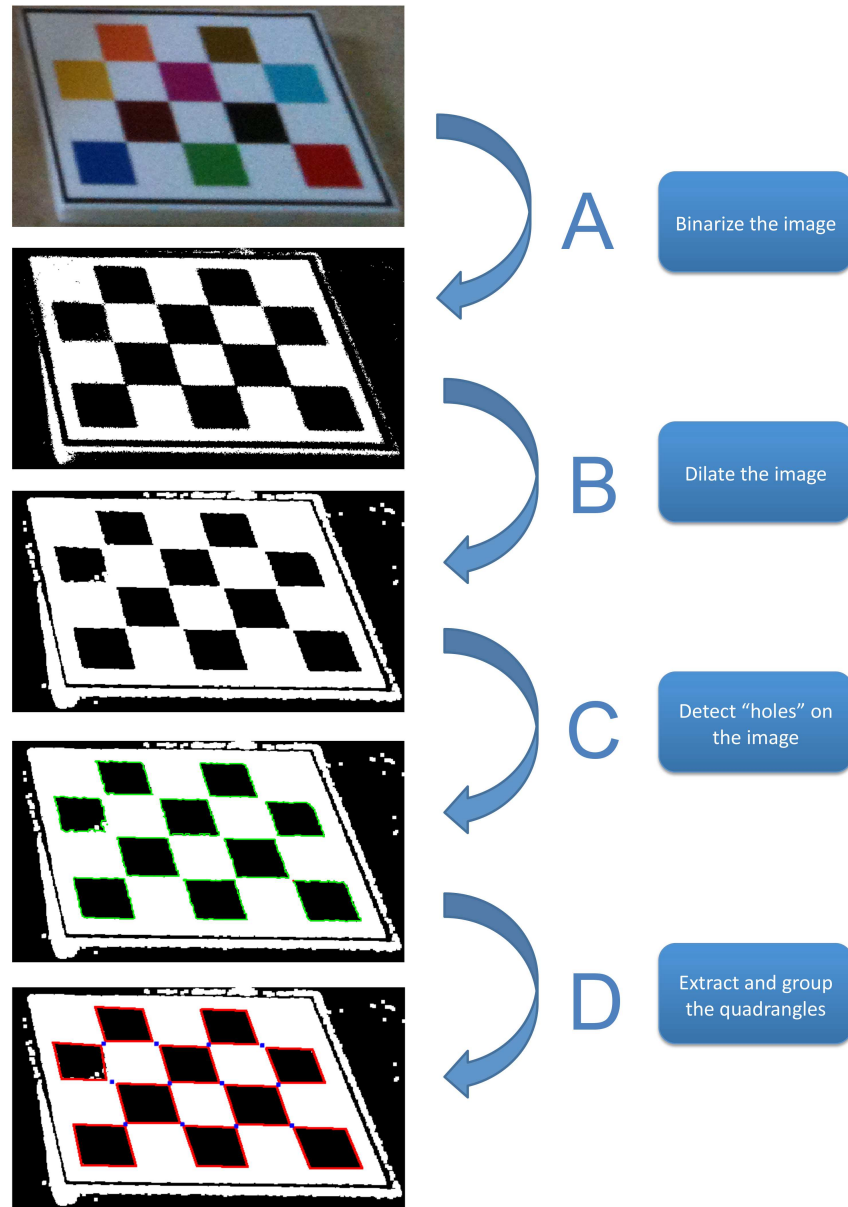


Fig. 4.8. Examples of internal corners estimation using quadrangles.

where max and min are the maximum and minimum value in the grayscale image respectively. An example of local and fixed value thresholding is shown in Figure 4.9. In the original image - Figure 4.9(a), max is 187, min is 0, and therefore, $T_l = 93.5$. And if

we use a fixed value $T = 155$ as the threshold, two top left squares of the checkerboard will be connected to the white background due to the low illumination condition as shown in Figure 4.9(c).

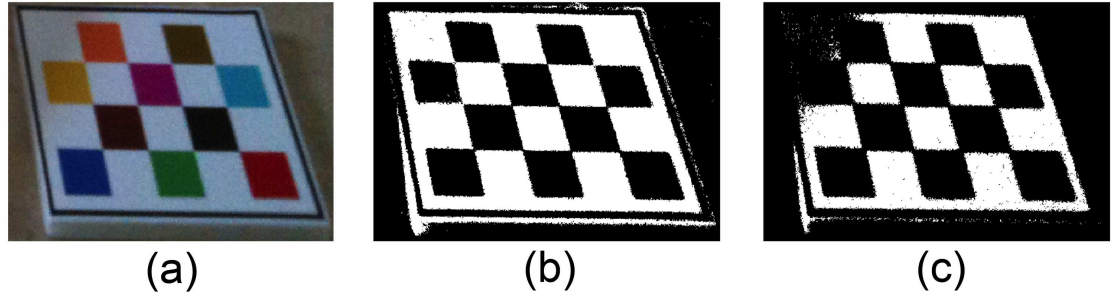


Fig. 4.9. An example of binarizing the image with a fixed threshold and a local threshold. (a) - original checkerboard image, (b) - binary image with local thresholding, and (c) - binary image with fixed value thresholding.

This local thresholding procedure will allow us to separate background pixels from the color patches even if they differ by a small amount.

Next, a morphological dilation operation [114] using a 3×3 square structuring element is done on the binarized version of the candidate region as shown in the third image of Figure 4.8. By using the dilation operation, the color squares in the checkerboard, which are typically the “holes” on the white background will be separated from each other as shown in Figure 4.8.

After dilating the image and finding the holes in the checkerboard image, we need to select the holes which can be well approximated by a quadrilateral. This step is done by using the Douglas-Peucker (DP) method [115] for estimating a contour from a sequence of points. The DP method finds the given number of dominant points (corners of a polygon) that represent a polygon approximation to the contour. Figure 4.10 demonstrates an example of extracting four dominant points from a series of points.

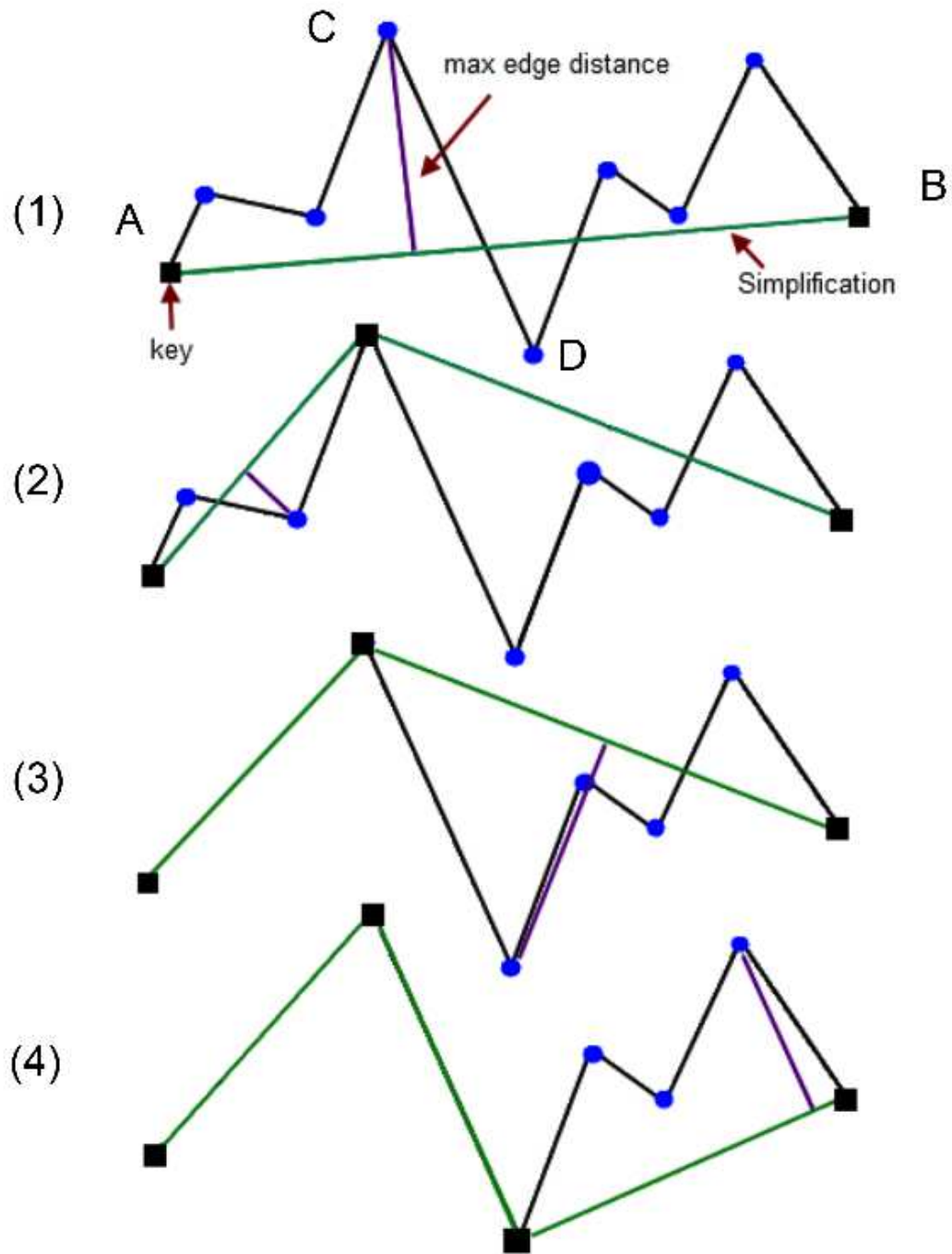


Fig. 4.10. An example of the Douglas-Peucker (DP) method.

In our implementation, we first look for two boundary points *A* and *B* on the contour with the largest mutual Euclidean distance and connect them with a line as shown in Fig-

ure 4.10(1). Then, we find another dominant point with the largest distance from this line, and if the distance is larger than all the rest of distances from other points, we add it into the list of dominant points. As shown in Figure 4.10(2), point C is the new dominant point found in the first iteration.

We then recursively find the next most distant point until we reach the total number of dominant points we need to obtain. Alternatively it can find the fifth dominant point since we are only interested in polygons with four points as candidate regions for the fiducial marker. Therefore, once we found the fourth dominant points, we will stop as shown in Figure 4.10(4).

In this way, we separate the connected area between checkerboard patches and obtain a group of quadrangles as shown in D of Figure 4.8, where all the quadrangles are marked with red lines.

Quadrangle Grouping

Groups of quadrangles are found using their position and neighborhood relation with respect to other quadrangles. Neighboring quadrangles are estimated according to the distance to every corner of every other quadrangle. The smallest of such distances are stored along with the respective corner and quadrangle's index.

Corner Estimation

A group of quadrangles should have been detected using our above quadrangle generation and grouping method. Finally, the groups of quadrangles obtained above are compared with the connected quadrangles for the desired pattern. First, the total number of quadrangles should be 10 and the size of each quadrangle should be similar. There will be 7 quadrangles with only two neighboring quadrangles and 3 quadrangles with 4 neighbors. The set of connected quadrangles, which has the same property as our checkerboard pattern, is declared as the checkerboard region.

The four dominant points of these quadrangles obtained from DP method are the external corners of the fiducial marker, while the internal corners of the fiducial marker will lie on the middle of two neighbor corners of different quadrangles as shown in the blur dots of D in Figure 4.8.

In summary, our checkerboard corner detection method starts with binarizing and dilating the image as shown in A and B in Figure 4.8. Then, we detect the color squares in the binary image and obtain the contours of all of them as shown in C in Figure 4.8. After that, we approximate all the contours into quadrangles using the Douglas-Peucker (DP) method, and the final corners are computed based on the dominant points of all the quadrangles as shown in D in Figure 4.8.

4.3 Blur Detection

Blur is one of the most frequent causes of image distortion. Image blur may be caused by a number of factors, such as out of focus blur, motion blur and loss of high frequency data during acquisition, compression or processing. As shown in Figure 4.11.a is an original food image, Figure 4.11.b is simulated motion blur, and Figure 4.11.c is simulated Gaussian blur. Out of focus blur results from incorrect focus of the entire or part of the image and is frequently modeled by convolution with a Gaussian kernel.

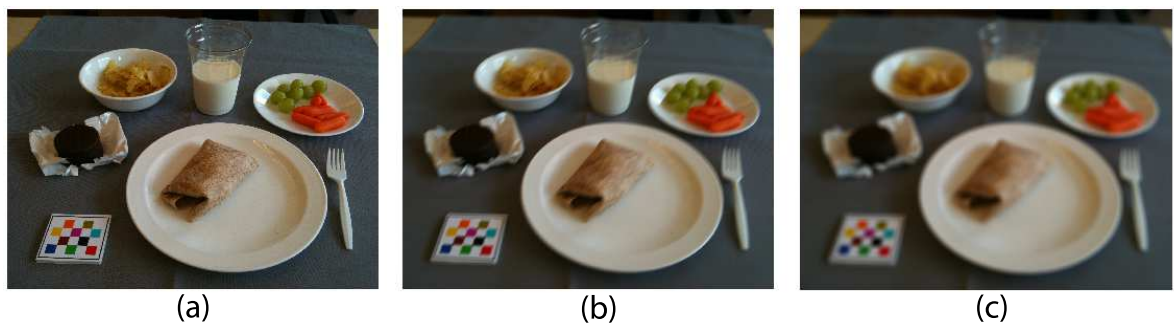


Fig. 4.11. An food image with motion and Gaussian blur.

Motion blur results from relative motion between the camera and the objects during exposure and is modeled by attenuating specific coefficients in the frequency domain as shown in Figure 4.12 [116]. As we can see from the figure, motion blur brings a huge amount of noise into the high frequency domain of the image, and as the degree of blur increases, the noise level becomes more significant.

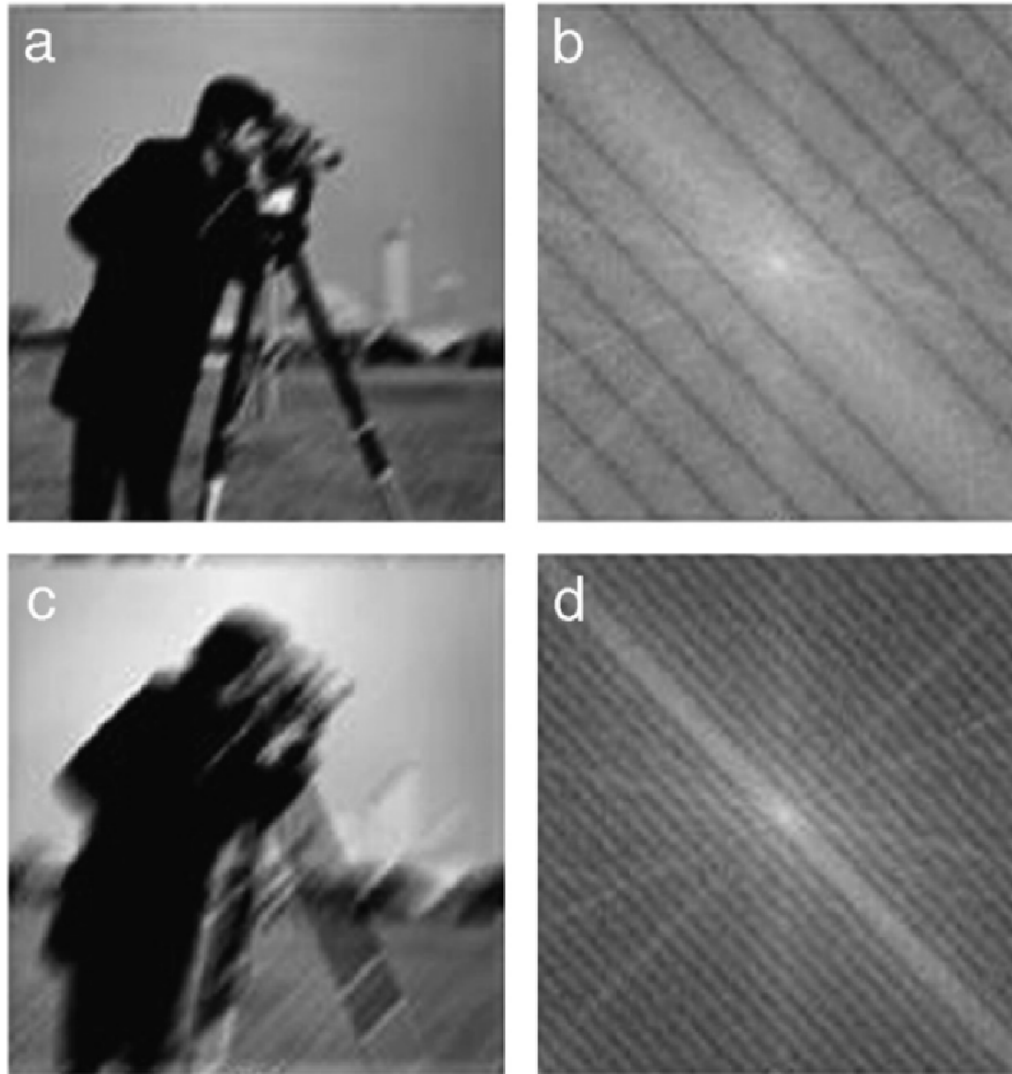


Fig. 4.12. (a) - Motion blurred image with $L = 10$, $\phi = 45^\circ$; (c) - motion blurred image with $L = 30$, $\phi = 45^\circ$; (b) and (d) - frequency response of (a) and (c) from [116].

A high degree of blur can change the edge structure of an image, resulting in incorrect segmentation due to missing boundaries and wrong classification due to changes in appearance of foods, especially the texture features. For object classification, some features, such as an RGB color histogram, are robust to a smoothing effect due to blur. However, some descriptors, mostly edge-based descriptors such as SIFT and SURF, are relatively sensitive to blur. Juan and Gwun [117] has conducted an experiment to compare the effect of blur on three local descriptors: PCA-SIFT [118], SIFT [86], and SURF [119]. They used Gaussian blur as shown in Equation 4.4:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.4)$$

with a radius from 0.5 to 9 using a set of images and the matching accuracy is computed for all the descriptors as shown in Figure 4.13. The accuracy of the matching for SIFT and SURF dropped dramatically as the degree of blur increases. Even though, PCA-SIFT seems robust to blur. However, the accuracy of PCA-SIFT is already low from the beginning.

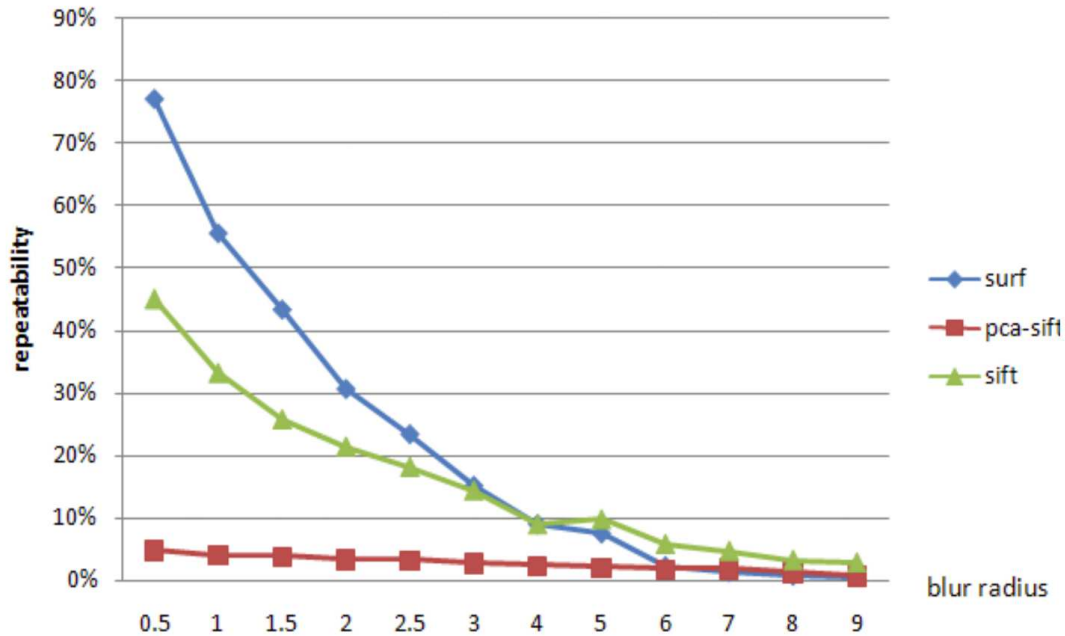


Fig. 4.13. The effect radius of the blur kernel on matching accuracy [117]

To investigate the effect of blur on meal images, we also conducted several experiments, using both real and simulated blurred images, to quantify the effect of blur on food image analysis. In the experiment, linear motion blur is applied to thirty food items. Linear motion blur is generally modeled as a convolution as shown in Equation 4.5:

$$h(x, y) = \begin{cases} \frac{1}{L} & \text{if } \sqrt{x^2 + y^2} \leq L \text{ and } \frac{y}{x} = \tan(\phi) \\ 0 & \text{otherwise} \end{cases}. \quad (4.5)$$

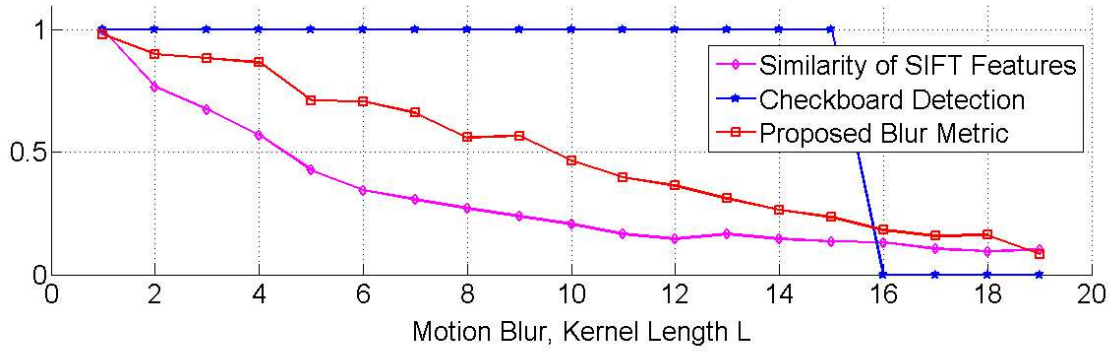
This is characterized by two parameters: L represents motion length and ϕ denotes motion direction [116].

We evaluated the effect of motion blur on the widely used SIFT features [86]. SIFT descriptors of an original food image are matched with those corresponding to different distorted images generated by linear motion blur with different motion lengths L and blur directions ϕ . SIFT accuracy is estimated by counting the percentage of SIFT points from the original image that could be matched to the SIFT points of the distorted image. Figure 4.14 shows the results of this experiment.

As shown in Figure 4.14, the purple line is the accuracy of SIFT matching, the results of our proposed sharpness metric (discussed below) are represented with the red dots, and the indicator of the success detection of the checkerboard is shown in blue dots. From the figure, we can see our proposed sharpness metric is consistent: when L increases, the sharpness metric decreases respectively. Also, when kernel length $L = 15$, the checkerboard is too blurred and we cannot detect it with our method. Further, the matching accuracy of SIFT descriptors is reduced quickly with the increase of motion blur kernel length L . Our result is consistent with the result presented by Juan and Gwun [117].

Hence, for successful food identification, it is important to eliminate the image quality degradation due to blur since blur has considerable influence on food identification and quantification.

Problems due to blur can be alleviated by either restoring the blurred image by post-processing or obtaining a blur assessment on the mobile device and if needed retaking the image. A number of methods have been proposed to restore blurred images [120–122].



Original Food Image



Motion Blur with L = 9



Motion Blur with L = 15

Fig. 4.14. Effect of image blur on SIFT features and the proposed blur metric.

However, there is no guarantee that even the best restoration method will be able to recover all the useful information from a degraded image. Furthermore, blur restoration methods are computationally expensive and may result in other distortions such as ringing.

Therefore, for our application it is more practical to do fast blur estimation on the mobile device and if needed, prompt the user to retake the image [9].

4.3.1 Related Work

Most of the existing blur assessment methods can be classified into two categories: frequency domain methods and spatial domain methods. Frequency domain methods estimate blur by utilizing the fact that blur could be caused by attenuation in the high-frequency coefficients. For example, the distribution of null DCT coefficients can be used for blur assessment [123]. Coefficients on the central diagonal of DCT matrix are used for this

purpose since they are good representatives of global blur and their histogram provides a blur metric.

Spatial domain methods focus on the “appearance” (e.g. edge and gradient) of an image in spatial domain. Mariliano et al. [124] proposed that the smoothing effect of blur on sharp edges can be used as an indicator of blur and the spread of edges can be used as a blur metric. A Sobel operator in the vertical direction is used to locate the edges in the luminance component of the image. The edge width, shown in Figure 4.15, which is defined as the distance between the start and end of local extreme is obtained. Finally, a global sharpness metric is obtained by averaging the edge widths for all the edge pixels found in the image. Generally, spatial domain methods are more efficient than the frequency domain methods as they do not require an additional transformation.

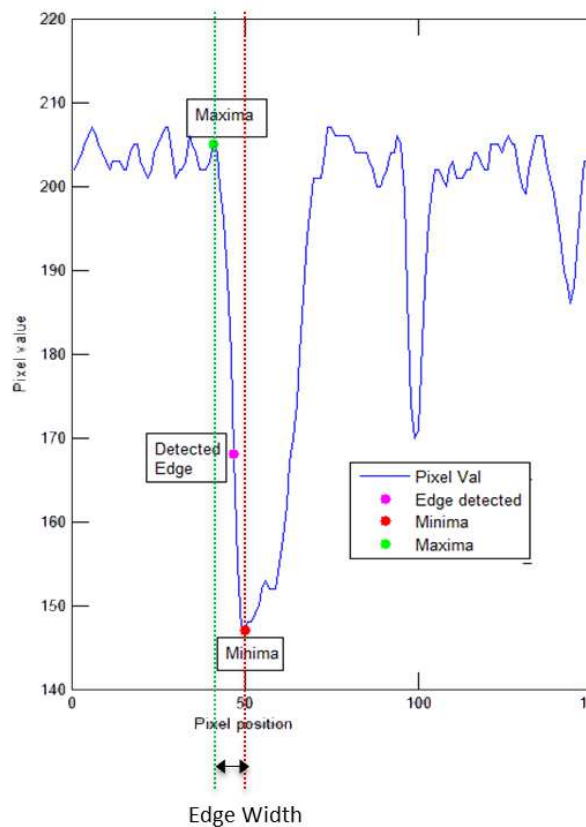


Fig. 4.15. An illustration of the measurement of the width around an edge pixel.

In this thesis we developed a low complexity blur metric by modifying a well known method, the cumulative probability of blur detection (CPBD) which utilizes the probability distribution of edge widths [125].

In [125], a no-reference blur metric is proposed that combines the edge width method [124] and the concept of “just noticeable blur”. Blur estimation begins with using an edge operator, such the Sobel operator, to estimate the vertical direction of the luminance component of the image. An edge binary map is constructed to indicate the location of points where the gradient magnitude assumes a local maximum in the vertical direction. Since the smooth areas have little influence on blur estimation, the image is divided into 64×64 blocks to determine the regions of interest. The blocks containing a number of edge points larger than a fixed threshold are considered as edge blocks. At each edge pixel within the edge blocks, the edge width is estimated as the distance between the start and end of local extrema [124]. Let e_i denote an edge pixel and $\omega(e_i)$ its corresponding edge width. Then, the probability of blur detection P_{BLUR} at each edge pixel can be expressed in the following form [125]:

$$P_{BLUR}(e_i) = 1 - \exp\left(-\left|\frac{\omega(e_i)}{\omega_{JNB}(e_i)}\right|^\beta\right), \quad (4.6)$$

where β is a value chosen between 3.4 and 3.8 with a mean value of 3.6 and JNB width, $\omega_{JNB}(e_i)$, depends on the local contrast C of the edge block corresponding to edge pixel e_i .

Finally, cumulative probability of blur detection (CPBD) is estimated as:

$$CPBD = P(P_{BLUR} \leq P_{JNB}) = \sum_0^{P_{JNB}} P(P_{BLUR}) \quad (4.7)$$

where $P(P_{BLUR})$ denotes the value of probability distribution function at a given P_{BLUR} [125].

The CPBD is the overall blur metric.

4.3.2 Our Proposed Method

For deploying blur assessment on a mobile device one needs low computation and memory requirements. Compared with the CPBD method, our contribution is modifying Equation 4.6, we removed the exponential part in the equation and replaced it with a look-up table structure. Therefore, the complexity is greatly improved and it is suitable for deployment on a mobile device. This section describes our modifications to CPBD. The blur metric CPBD (Equation 4.7) can be represented as:

$$CPBD = \frac{|S_1|}{|S_e|}, \quad (4.8)$$

where S_e denotes the set that contains all the edge pixels, and S_1 denotes the set of edge pixels with $P_{BLUR} \leq P_{JNB}$. Using Equation 4.6, this condition is same as:

$$1 - \exp\left(-\left|\frac{\omega(e_i)}{\gamma}\right|^\beta\right) \leq 0.63 \Rightarrow \omega(e_i) \leq \gamma \times [-\ln(0.37)]^{\frac{1}{\beta}}, \quad (4.9)$$

where γ donates the JNB width $\omega_{JNB}(e_i)$. Since the JNB width, γ , takes on only two values 5 and 3 when the local contrast is $C \leq 50$ and $C > 50$ respectively [125] (the local contrast C is obtained by subtracting minimum intensity from the maximum intensity within the same edge block). Therefore,

$$\gamma \times [-\ln(0.37)]^{\frac{1}{\beta}} = \begin{cases} 5 \times 0.9984, & \text{if } C \leq 50 \\ 3 \times 0.9984 & \text{if } C > 50, \end{cases} \quad (4.10)$$

$$\Rightarrow \omega(e_i) \in \begin{cases} \{2, 3, 4\} & \text{if } C \leq 50 \\ \{2\} & \text{if } C > 50, \end{cases} \quad (4.11)$$

In the last step we utilized the fact that $\omega(e_i)$ is an integer since it represents the edge width. Let $H(\gamma, k)$ denote the number of edge pixels with JNB width γ and edge width k . Then Equation 4.8 becomes:

$$CPBD = \frac{\sum_{\gamma=\{3,5\}} \sum_{k=2}^{\gamma-1} H(\gamma, k)}{|S_e|}, \quad (4.12)$$

Estimation $H(\gamma, k)$ can be done by storing it as a 5×4 matrix and following the steps in Algorithm 1.

Input: Gradient Mask and Edge Blocks

Output: $H(\gamma, k)$

Initialize each $H(\gamma, k)$ to zero.

foreach *Edge Block* W **do**

$C \leftarrow \text{Max}(I_W) - \text{Min}(I_W)$ **if** $C \leq 50$ **then**

$\gamma \leftarrow 5$

else

$\gamma \leftarrow 3$

end

foreach *Edge Pixel* $e_i \in W$ *such that* $w(e_i) < \gamma$ **do**

$H(\gamma, w(e_i)) \leftarrow H(\gamma, w(e_i)) + 1$

end

end

Algorithm 1: Estimation of H

Edge width estimation is done by utilizing the fact that we need edge widths only when they are less than γ (3 or 5). Larger edge widths need not be computed. Our proposed method of blur metric assessment saves considerable time compared to the original method since it involves only simple additions and multiplication (for gradient estimation) and none of the exponential (Equation 4.6) need to be computed.

4.4 Experimental Results

Our proposed methods for fiducial marker detection and blur estimation were deployed on an Apple iPhone and were tested under different imaging conditions including various

backgrounds and with complex illumination conditions. For fiducial marker detection the pattern matching process is done within a small part of the image. Therefore, we save time spent on exhaustively searching for potential corners, as done in traditional methods [105–107]. Experiments with the version deployed on the iPhone show that our method takes approximately 1 second to detect the fiducial marker and is 10 times faster than the widely used OpenCV implementation [112].

Our proposed blur metric was evaluated on a set of images with thirty food items with different kernel length of motion blur. Figure 4.14 shows that the sharpness metric is consistent when the kernel length of blur increases, the sharpness metric decreases correspondingly. On the TADA mobile application (mpFR), a sharpness metric threshold of $T_{CPBD} = 0.5$ is used to detect the images with too much blur. The average computational time of our method is reduced from 9 seconds to 1 second on the mobile device for 2048×1536 sized images compared with the original CPBD method.

4.5 Overview Of Color Correction

Color can serve as one of the key variables in imaging [126]. A consistent color descriptor of an object is very useful to improve the results of image analysis. The colors of objects recorded by camera depend on three factors. First, illumination conditions in the scene are unknown in most circumstances. Second, objects in the scene have different intrinsic surface properties which produces spectral reflection given the light conditions. Moreover, the capturing device has various photo metric parameters (e.g., exposure time, white balancing, gamma correction) that affect the color representation of the objects in a scene. Therefore how one can increase the robustness of the color descriptor remains one of the major challenges for current color imaging systems.

Some approaches seek to overcome these problems by estimating illumination invariant color descriptors from a database of images. These features include the RGB histogram, color moments, and C-SIFT [127, 128]. In [127], a combined set of color descriptors with

invariant properties surpass the performance of intensity based descriptors by 8% on category recognition.

Another approach to deal with this issue is to estimate the intrinsic relationship between images illuminated with reference lighting and images acquired with unknown lighting conditions. In this approach, color correction is often used for determining perceptual consistency. By using this process, the overall color characteristics of the image will be improved, and at the same time, color differences under low light conditions will be enhanced, which can also be beneficial to image segmentation.

Color correction, also known as *color balancing* is the global adjustment of color intensities in an image. The goal of such adjustment is to render neutral colors in an image correctly. Color correction changes the overall colors in an image and is often used for colors other than neutrals to appear correct or pleasing. Methods for this type of correction are generally known as gray balance, neutral balance or white balance [129]. There are many color correction methods [130–132]. In general, most of these methods contains two steps: first, estimate the illumination color temperature of the scene by using the image data and statistical information. Then, use the illumination parameters to obtain the color corrected image. However, in our application, we not only have the illumination parameters which are estimated from the white patch on the fiducial marker, we also have the other color information from all 11 color patches on the fiducial marker. Our goal is to improve the overall color correction accuracy using all the information from the fiducial marker.

In an earlier approach we used the color correction method described in [133]. This approach represented illumination color features using the Macbeth color board with 24 colors. Conversion vectors were defined from a source illumination to a target illumination. Assume that illumination A is the target, a conversion vector from illumination B to illumination A can be defined for each color patch on the Macbeth color board. For each color path i , the conversion vector C_i is defined as the difference between RGB color of each path in illuminations A and B . Thus, the conversion vector from illumination B

to illumination A , C_{A-B} , is an average vector of conversion vectors of all color patches, equivalently:

$$C_{A-B} = \frac{1}{24} \sum_{i=1}^{24} C_i = (C_r, C_g, C_b) \quad (4.13)$$

The illumination conversion vector is then used according to adjust (or correct) the pixel color in images of illumination B . Let the RGB value of a pixel at (x, y) in the image of illumination B be (R_{xy}, G_{xy}, B_{xy}) . This color value (R_{xy}, G_{xy}, B_{xy}) is corrected by adding the illumination conversion vector C_{A-B} above as follows

$$(R'_{xy}, G'_{xy}, B'_{xy}) = (R_{xy}, G_{xy}, B_{xy}) + (C_r, C_g, C_b) \quad (4.14)$$

We also adopted a simplified version of the approach proposed by Srivasrava, et al [134] to address the problem of visually matching two known display devices in color management systems. The method is implemented through the use of 3D look-up tables (LUT). In our case, we consider a uniformly sampled LUT of size $3 \times 3 \times 3$. Interpolation methods have been proposed in one and more dimensional spaces and on regular or irregular shaped data grids. In general cases, an input point $[x, y, z]_c$, whose output needs to be predicted, can have k neighbors $[x, y, z]_i$ for $i = 0, 1, \dots, k - 1$. Let $d(c, i)$ be some metric of distance between the points c and its neighbors i . Note that each neighbor is an entry in the table and hence their outputs $f([x, y, z])_i$ are known. Then using interpolation

$$f([x, y, z])_c = \psi(f_i, d_i) \quad \text{for } i = 0, 1, \dots, k - 1 \quad (4.15)$$

where ψ is determined by the chosen method. For example, a simple 1D linear interpolation is

$$f_c = (f_0 \cdot d(c, 1) + f_1 \cdot d(c, 0)) / (d(c, 0) + d(c, 1)) \quad (4.16)$$

However, illumination change from the reference image to the target image is often not uniform in the RGB color space and the color appearance of the scene is a complex process affected several things including camera behavior, the illumination condition and surface reflection. Therefore a single conversion vector or 3D interpolation may not be sufficient to accurately correct colors in different regions in the image.

To correct colors in an image containing the color fiducial maker from unknown sources or illuminations, we propose below three chromatic adaptation models that use more perceptually uniform color space models in the following Section 4.6. Compared with the white balance method introduced in Chapter 5, this method does not require any user input. The color information in the checkerboard in the scene is utilized.

4.6 Scene Illumination Detection Using Color Mapping

Our methods begin with capturing an image of the color fiducial marker using a digital still camera or a mobile device camera under an unknown illumination and the use of the known color patches in the fiducial marker as the reference colors. As shown in Figure 4.1, the next step is fiducial marker detection which is been discussed in Section 4.2.

After estimating the 11 colors from the fiducial marker, we then evaluate the illumination condition by determining if the colors on the fiducial marker fall within a prescribed range of RGB values. The reason for the illumination check is to avoid bad illumination conditions. Specifically conditions that are too dark, too bright or have incorrect color temperature. We can also use this to assist the user in capturing a better image by providing feedback to them [9]. Since the mobile device has limited computational power and memory resources, the color correction step is done on the server (see Figure 1.1).

The RGB components of the color patches on the fiducial marker from the acquired image are extracted and a mapping between the reference fiducial marker colors and the colors from the fiducial marker from the acquired image is established [135].

Finally, this mapping is used to correct the colors of the acquired image to match the reference colors. We investigated three different color correction methods: a linear RGB to RGB transform, a nonlinear RGB to RGB transform, and a linear model in CIELAB color space.

4.6.1 Color Space Models

First, a linear RGB mapping color correction method based on the von-Kries model [136] is introduced. A chromatic adaptation transformation (CAT) mapping XYZ_1 in viewing condition 1 to XYZ_2 in viewing condition 2 to achieve a perceptual match can be formulated as follows:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{H}^{-1} \begin{bmatrix} \frac{L_{w2}}{L_{w1}} & 0 & 0 \\ 0 & \frac{M_{w2}}{M_{w1}} & 0 \\ 0 & 0 & \frac{S_{w2}}{S_{w1}} \end{bmatrix} \mathbf{H} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}, \quad (4.17)$$

where LMS_{wi} , ($i = 1, 2$), is the LMS cone responses to reference white w_i . \mathbf{H} is a 3×3 non-singular matrix which represents a transformation from XYZ to LMS. Based on von-Kries model [136], the color transformation due to the change of scene illumination (near daylight) can be modeled as a linear transformation in LMS color space [137], a diagonal transformation matrix in LMS color space need to be computed to transform the color of the image into corrected color.

The transformation between RGB and CIEXYZ can be represented by a Bradford transform [138]:

$$\begin{bmatrix} R & G & B \end{bmatrix}^T = T_{3 \times 3} \begin{bmatrix} X & Y & Z \end{bmatrix}^T, \quad (4.18)$$

where $T_{3 \times 3}$ is a forward transformation which is determined by the capturing device and illumination conditions.

We substituted Equation 4.17 into Equation 4.18. Then, we obtained a linear model which contains conversion matrix D in Figure 4.16 as follows:

$$\begin{bmatrix} R_{cor} \\ G_{cor} \\ B_{cor} \end{bmatrix} = \underbrace{\mathbf{T}_2 \mathbf{H}^{-1} \begin{bmatrix} \frac{L_{w2}}{L_{w1}} & 0 & 0 \\ 0 & \frac{M_{w2}}{M_{w1}} & 0 \\ 0 & 0 & \frac{S_{w2}}{S_{w1}} \end{bmatrix} \mathbf{H} \mathbf{T}_1^{-1}}_{\mathbf{D}_{3 \times 3}} \begin{bmatrix} R_{ori} \\ G_{ori} \\ B_{ori} \end{bmatrix}, \quad (4.19)$$

where R_{ori} , G_{ori} and B_{ori} is the RGB values of the original image and R_{cor} , G_{cor} and B_{cor} is the RGB values of the corrected image, respectively. The calibrated RGB value is simplified to the multiplication of the original RGB value with a 3×3 matrix D.

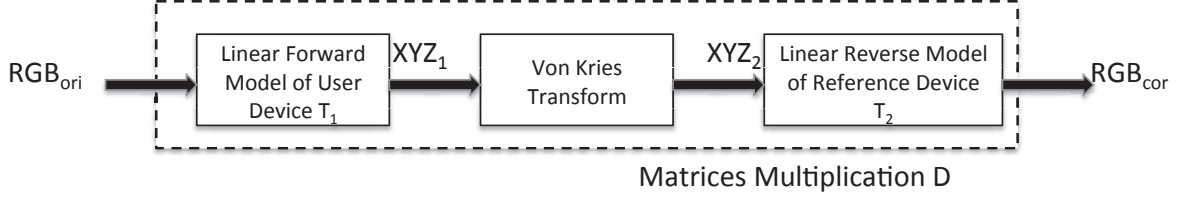


Fig. 4.16. The linear model from RGB to XYZ.

To determine the conversion matrix D , we formulated a least square regression problem that best transforms the 11 colors in the fiducial marker into corresponding reference colors:

$$D_{3 \times 3} = \arg \min_{D_{3 \times 3}} \sum_{i=1}^{11} \left\| (RGB_i)_{ref}^t - D_{3 \times 3} (RGB_i)_{ori}^t \right\|^2, \quad (4.20)$$

where “ ref ” is the sRGB values of the 11 colors on the fiducial marker under the D65 illumination [135].

As described above, this method suggests a linear transformation between two RGB space images. However, this could lead to a over simplified system with low accuracy. We propose next a non-linear model with more color fidelity, where the color variation introduced by illumination conditions and camera models are considered.

In the previous model, we proposed a linear system based on the assumption that there is no interaction between each RGB channel. This assumption is not satisfied in many cases due to the factors including gamma correction, non-uniform illumination and image processing process integrated inside the camera. To account for the non-linearity, we describe an approach for color correction using a non-linear 10×3 mapping between source image and target image. Let the vector $P_{10 \times 1}$ contain the cross product and second order terms of “ R_{ref} ”, “ G_{ref} ” and “ B_{ref} ” described above:

$$P_{10 \times 1} = \left[R_{ori} \quad G_{ori} \quad B_{ori} \quad R_{ori}^2 \quad G_{ori}^2 \quad B_{ori}^2 \quad R_{ori}B_{ori} \quad R_{ori}G_{ori} \quad G_{ori}B_{ori} \quad 1 \right]^T \quad (4.21)$$

$$\begin{bmatrix} R_{cor} \\ G_{cor} \\ B_{cor} \end{bmatrix} = \mathbf{D}_{3 \times 10} \mathbf{P}_{10 \times 1} \quad (4.22)$$

By using the 11 RGB values of the colors from our fiducial marker on the reference image and target image, the matrix \mathbf{D} is formulated as an optimization problem.

$$\mathbf{D}_{3 \times 10} = \arg \min_{\mathbf{D}_{3 \times 10}} \sum_{i=1}^{11} \left\| (\mathbf{RGB}_i)_{ref}^t - \mathbf{D}_{3 \times 10} \mathbf{P}_{10 \times 11} \right\|^2 \quad (4.23)$$

Similarly to the last step of the linear model, the color correction image is obtained by multiplying the RGB value of the target image with 10×3 matrix \mathbf{D} .

The last chromatic adaptation model we implemented is shown in Figure 4.17 and is based on the uniform CIELAB color space. Though both CIEXYZ and CIELAB are considered device-independent color spaces, the CIELAB color space includes all perceivable colors and is considered more perceptually uniform than CIEXYZ [139, 140]. Therefore, a linear mapping deployed in CIELAB color space might be more reasonable. As shown

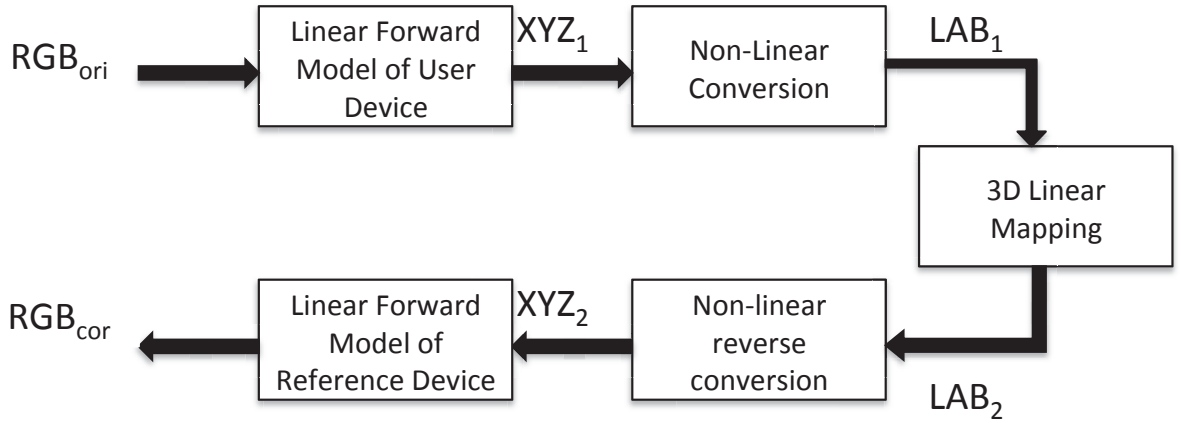


Fig. 4.17. The color model from RGB to LAB.

in Figure 4.17, a gamma correction is first used to obtain the linear RGB values. Since we are considering the general case where the parameters of imaging devices are unknown, we set the value of gamma to 1.1. After this, we map the original image from RGB to XYZ

space with the Bradford Transform. The forward transformation matrix we are using is the sRGB standard with D65 illumination and can be found in [141]. The following step is to normalize XYZ for the D65 white point. Finally, a non-linear forward transform between the normalized XYZ to LAB is obtained as follows [142]:

$$L = 116 * f(Y/Y_n) - 16 \quad (4.24)$$

$$a = 500 * [f(X/X_n) - f(Y/Y_n)] \quad (4.25)$$

$$b = 200 * [f(Y/Y_n) - f(Z/Z_n)] \quad (4.26)$$

$$f(x) = \begin{cases} (x)^{1/3} & \text{if } x > 0.008856 \\ 7.787(x) + 16/116 & \text{otherwise} \end{cases} \quad (4.27)$$

X_n , Y_n , and Z_n are the tristimulus values of the white point. The L coordinate in CIELAB is correlated to perceived lightness. The a and b coordinate are the red-green and yellow-blue of the color-opponent respectively.

The transform in CIELAB space is perceptually uniform. We use a 3-dimensional linear transformation that converts the 11 LAB_1 values of the color patches from the fiducial marker in viewing condition 1 to the 11 LAB_2 values of the color patches in viewing condition 2. The 3×3 mapping matrix D is obtained as follow:

$$D_{3 \times 3} = \arg \min_{D_{3 \times 3}} \sum_{i=1}^{11} \left\| (LAB_i)_{ref}^t - D_{3 \times 3} (LAB_i)_1^t \right\|^2. \quad (4.28)$$

Finally, a reverse transform is done to convert the image from CIELAB to RGB.

Sample color correction results using these methods, as well as comparison to the methods described by Choi [133] and Srivastava [134] are described next in Section 4.7.

4.7 Experimental Results

To evaluate the performance of our proposed color correction methods the following experiment is performed. Two test targets are used in the experiment, specifically a color fiducial marker as shown in Figure 4.2 and a GretagMacbeth Colorchecker [143] which is

a calibrated color reference chart. Both targets are placed inside a "SpectraLight II" illumination booth which provides four uniform calibrated light sources: simulated daylight (CIE D65, 6500 K), horizon daylight (simulated early morning sunrise or afternoon sunset, 2300 K), CIE A (incandescent home lighting, 2856 K), and commercial fluorescent (cool white, 4000 K). Under each illumination an image of the targets was acquired using the camera of an iPhone 3GS.

The captured images under non-D65 illumination were then color corrected using the three methods described in the previous section, as well as two other methods described by Choi [133] and Srivastava [134]. To evaluate the accuracy of each method, the euclidean distance between the average color of each color patch on the GretagMacbeth Colorchecker $(\tilde{R}_i^t, \tilde{G}_i^t, \tilde{B}_i^t)$ and the known sRGB values of each patch under D65 illumination (R_i^r, G_i^r, B_i^r) is obtained using Equation 4.29 and is shown in Table 4.1. The "Before" column represent the absolute color different between the colors on GretagMacbeth checkerboard under the testing illumination and reference illumination. The euclidean distance is defined as:

$$\Delta = \frac{1}{24} \sum_{i=1}^{24} \left\| \left(\tilde{R}_i^t, \tilde{G}_i^t, \tilde{B}_i^t \right)^t - \left(R_i^r, G_i^r, B_i^r \right)^t \right\| \quad (4.29)$$

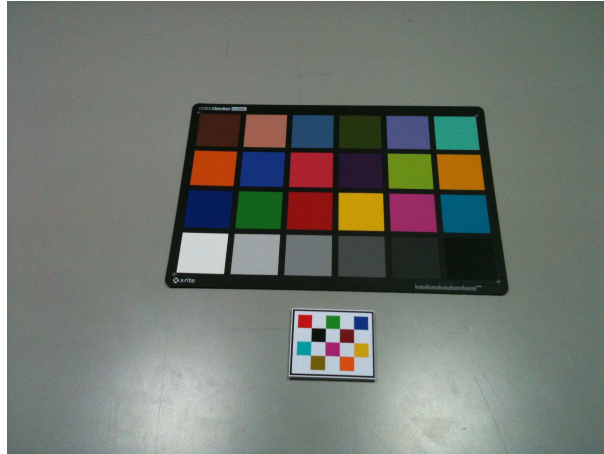


Fig. 4.18. The image with reference light condition (D65).

Table 4.1
Mean RGB channels errors (Δ) between the reference image and transformed images

Lighting	Error	Before	Linear-LAB	Linear-RGB	Polynomial	Srivastava	Choi
Horizon Lig	RGB	37.20	16.13	25.47	26.26	33.91	34.04
	Red	14.87	7.18	9.36	8.30	13.20	9.81
	Green	7.57	3.88	5.16	5.22	6.54	12.93
	Blue	31.21	11.67	21.03	22.34	26.90	26.49
Incandescen	RGB	22.11	18.89	22.06	20.23	19.46	21.64
	Red	10.56	8.75	12.57	9.64	8.40	9.84
	Green	8.14	9.65	8.82	8.73	8.07	8.78
	Blue	14.84	10.34	13.39	13.59	13.14	13.64
Coolwhite	RGB	17.23	7.92	12.34	12.23	15.13	16.29
	Red	12.04	3.52	9.60	9.58	11.10	11.22
	Green	6.53	4.41	5.03	4.91	6.28	6.14
	Blue	6.74	3.97	3.69	3.61	4.47	5.72

Figure 4.19 and 4.20 show the images after color correction. From the results summarized in Table 4.1 and Figure 4.19 and 4.20, we can show that our proposed methods are offering better color consistency in various illumination tests than the methods derived by Choi [133] and Srivastava [134]. Table 4.1 lists the mean RGB channels errors (Δ) between the reference image and transformed images for each method under three different illumination, where each row lists the errors of all five methods under one lighting source, and each column lists the errors of one method under all three lighting sources. For instance, the overall error of each method (Original, Linear-LAB, Linear-RGB, Polynomial, Srivastava and Choi) under the horizon lighting is 16.13, 25.47, 26.26, 33.91, 34.04 respectively.

The linear model in CIELAB color space has the best performance based on overall RGB error (see Table 4.1). When the white point is very different from D65 (i.e. Horizon light), it performs better than any other method. However, if we look at the individual color channels separately, it does not consistently have better correction. For example, the errors

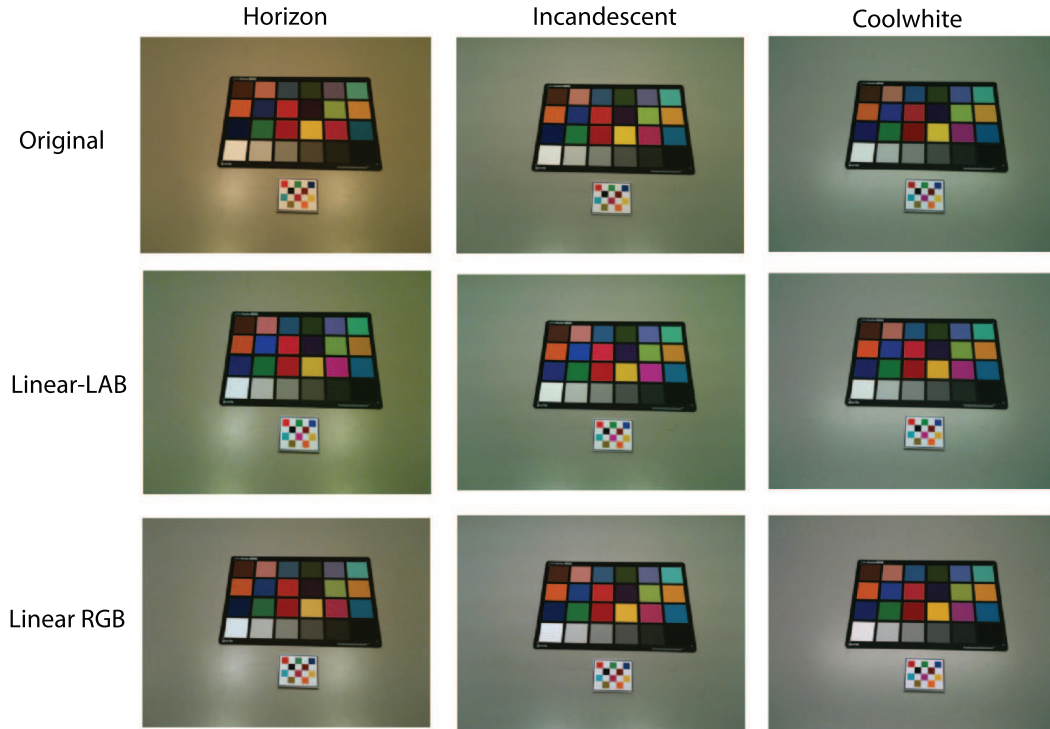


Fig. 4.19. Comparison of the five color correction methods - part1

of the red and green channel under the incandescent light and the blue channel under the cool white light are 8.75, 9.65 and 3.97, which performs worse than some methods. This error could be introduced by the gamma correction step. Since we assume that we do not have any knowledge about the capturing device, the gamma value is fixed. The gamma value should be independent in each color channel and each device.

The linear model and the nonlinear model in RGB color space are better than the methods described by Choi [133] and Srivastava [134]. They both perform similarly and improve the overall color consistency. Nevertheless, we observed that these two methods introduce more error in the red and green channels under incandescent light. The reason for this error could be the simplified nature of our color appearance model.

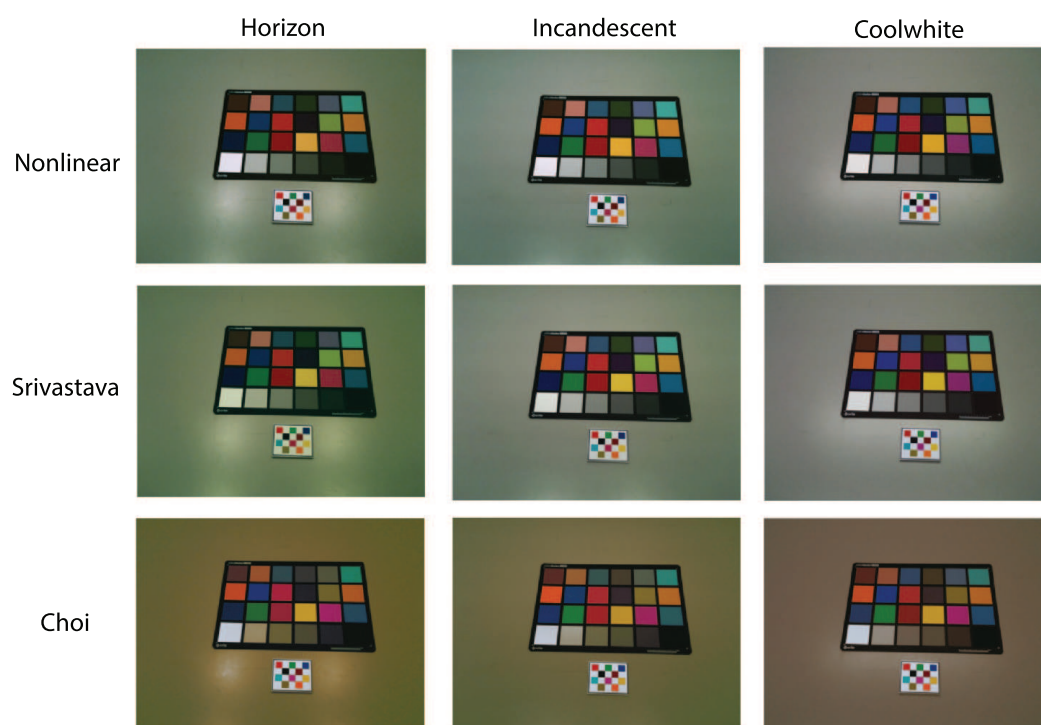


Fig. 4.20. Comparison of the five color correction methods - part2

5. WHITE BALANCE ON MOBILE DEVICES

5.1 Introduction

In Chapter 4, we have described a color correction method which utilizes a known object - the color checkerboard in the image as a color reference. Based on 11 colors extracted from the checkerboard in the scene, a linear or non-linear color transform is computed and used to correct the image. Compared with traditional methods [144–146] no white point or color temperature needs to be estimated.

In image processing, color balance or color correction is the adjustment of the intensities of the colors. An important goal of this adjustment is to render specific colors, particularly neutral colors, correctly. The general method is sometimes known as gray balance, neutral balance, or white balance.

In this chapter, we will introduce a white balance method utilizing user inputs. All the user inputs will be weighted and a white point will be computed based on them. We will also study the impact of illumination on imaging while the other variables remain unaltered [147].¹

Digital image processing is the key component in a digital still color camera (DSC). An overview of the image processing pipeline is presented in [148]. As we can see in Figure 5.1, the light rays from the scene field of view pass through the sensor, aperture, and lens. The focus and exposure are adjusted to capture the scene. Pre-processing is deployed after the sensor block in the camera in order to remove noise and other artifacts. Then white balancing is done to retain the color consistency. The demosaicing step increases the correction using neighboring pixels. In this chapter, we will mainly focus on improving the white balance by utilizing the user input.

¹The work in this chapter was jointly done with my former Purdue colleague Dr. Satyam Srivastava [147].

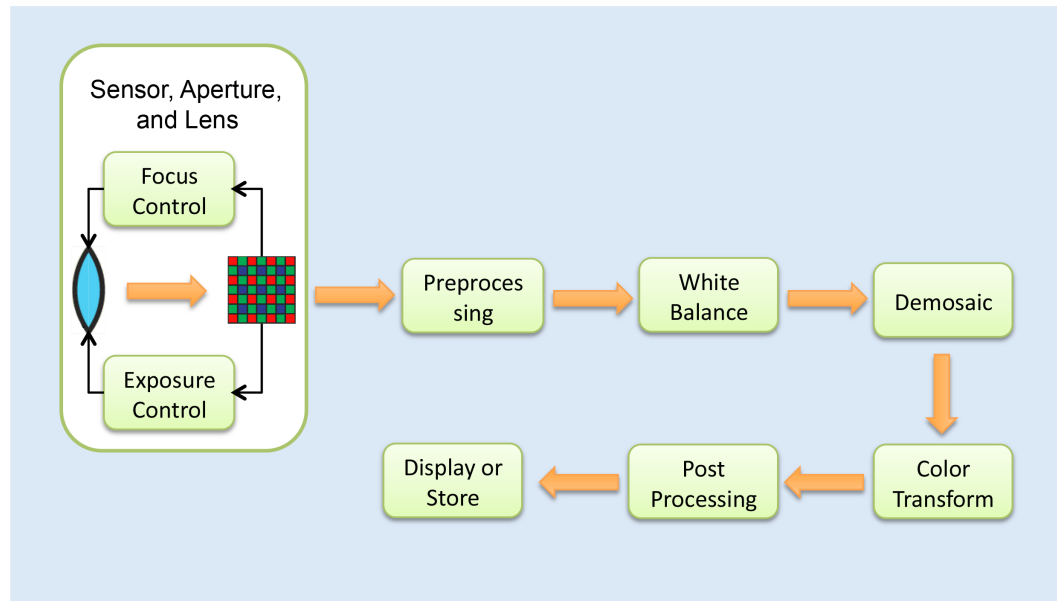


Fig. 5.1. The image processing inside a digital camera.

The problem of color balance or white balance has been widely investigated and several solutions have been developed. We review some of the techniques in Section 5.2. Many of these techniques are effective in common usage scenarios and are deployed in commercial imaging devices. While most techniques are based on autonomous methods, others require the user to either manually specify the illumination white point or allow the device to capture a white target. We extend the method of taking user input to arbitrary colors and estimating the white point from these. The captured image is then color corrected using this estimated white point.

There are two main contributions in this chapter. First, we note that human vision is remarkably resilient to viewing conditions when generating a color descriptor for an object. Even under extreme conditions when the visual system fails, a user can still generate a veridical color descriptor based on prior knowledge about objects in the scene. We devise an intuitive method for a user to specify veridical color descriptors for color patches in the captured image. This is facilitated by the availability of graphical interfaces on many mobile imaging devices such as smart phones. Second, we describe a scheme to synthesize the white point descriptor using a weighted combination of the colors provided by the

user. These methods are described in Section 5.3 and the testing results are provided in Section 5.4.

5.2 Review Of White Balancing Techniques

Color is an important attribute in digital imaging for photography and image analysis. It is a complex phenomenon (even after discounting the human perceptual system) that is sensitive to an object surface characteristics, illumination, ambient conditions, and the electro-optical properties of the capture and display device. In many imaging applications the goal of color balancing is to obtain a color descriptor that is consistent with the “veridical” color descriptor of the object being captured. A veridical descriptor is the color that a “typical human observer” would associate with the object under typical viewing conditions [149].

The Human Visual System (HSV) is able to determine what is white in the scene under various lighting conditions. In the contrast, some digital color cameras are lack of such ability, which is also called white balancing. White balancing is a technique to recover the illumination independent representation of the digital images. As shown in Figure 5.2, the left image shows a photo directly came from the digital camera. The right image shows the white balanced image using the Gray World method [145]. Note that the color temperature of a light source is the temperature of an ideal black body radiator that radiates light of comparable hue to that of the light source.

It is known that an object may have different color appearance when the incident illumination is changed [149]. The light reflected from an object is mainly determined by two factors: First, it depends on the color temperature of the incident light source in the scene. The color temperature is the surface temperature of a spectrum of light radiated from an ideal black body. In practice, color temperature is used to describe a characteristic of visible light, whether it appears more yellow (“warm”) or more blue (“cool”), in degrees of Kelvin.



Fig. 5.2. A same scene with different color temperature.

Second, it depends on the surface reflectance of the objects scene, which are characterized by their reflectance coefficients [149]. The human vision system is very good at perceiving stable colors across changes in illumination. For example, when a cup of milk is illuminated under a light with low color temperature, the reflection recorded by the imaging device will appear reddish [149]. However, the human vision system is capable of maintaining the constancy of the color under different lighting conditions and still recognize the color of milk as white. Nevertheless, when a digital device captures an image of a scene, the camera sensor has a fixed response for colors under different illumination conditions. Therefore, a process to compensate for this characteristic needs to be found to adjust the color delivered by camera.

As indicated above, to remove “unrealistic” color caused by different illumination conditions and account for the changes in human visual sensitivity under various illumination sources, a white balance operation is implemented in most digital cameras.

This process is usually done by using automatic white balance (AWB), custom white balance, or manual illumination selection. We proposed a method that is categorized as a manual illumination selection.

White balance is the global adjustment of the intensities of the colors in order to render a better or more pleasure color casting for the digital images. Automatic white balance (AWB), a traditional operation in most digital cameras, consists of two parts: First, the illumination color temperature of the scene is estimated using the image and statistical information. Then, based on the estimated illumination, a color adjustment is done to preserve the gray tones. Some popular techniques include the gray world method [145], the perfect reflector [146], the use of fuzzy rules [150], the Chikanas Method [144] and the use of dynamic thresholds [151]. We will introduce some of these methods in detail below.

1. The gray world method (GWM) [145] is based on the assumption that “for a given image with sufficient amount of color variation, the average value of the red, green, and blue components of the image should average out to a common gray value” [145]. Therefore, the method attempts to maintain this assumption by scaling the red, green, and blue color components with different scale factors α, β, γ . The gray world method provides a constancy solution independent of the illuminant color by its average value as shown in Equation 5.1:

$$(\alpha R, \beta G, \gamma B) \mapsto \left(\frac{\alpha R}{\frac{\alpha}{n} \sum_i R}, \frac{\beta G}{\frac{\beta}{n} \sum_i G}, \frac{\gamma B}{\frac{\gamma}{n} \sum_i B} \right), \quad (5.1)$$

where α, β, γ are the scale factors for the red, green, and blue color channels, n is the total number of pixels, and R, G, B are three values for each pixel in RGB color space.

2. The perfect reflector [146] is another widely used approach for color balancing. The assumption of the perfect reflector model is “an illuminated glossy surface reflects a highlight known as Perfect Reflector or Specular Reflection,” which means the brightest pixel in the image is often considered as a specular surface. Therefore, the illuminant of the scene can be approximated as the color of this specular point.

Then, the largest value for R, G, and B components in the image is used to scale the RGBs of the entire image in order to perform white balance adjustment as shown in Equation 5.2.

$$(R, G, B) \mapsto \left(\frac{R \cdot G_{max}}{R_{max}}, G, \frac{B \cdot G_{max}}{B_{max}} \right), \quad (5.2)$$

where R_{max} , G_{max} and B_{max} are the maximum values for RGB channel respectively.

3. In the fuzzy rules model [150] the image is converted from the RGB color space to the YCrCb color space. And then the color characteristics in the YCrCb color space are used for white balance. Finally, the color corrected YCrCb image is converted back to RGB color space.

Figure 5.3 shows the GretagMacbeth ColorChecker [143] illuminated with daylight and the results obtained by the three methods described above. They also visually improve the color consistency for the image and can be used in a digital camera for white balancing.

These methods often fail when the image contains large objects or has a background with a uniform color or the scene contains few relative white points.

The “white point” is the chromaticity of a white object under the illuminant in image, which is often represented as a set of tristimulus values [139]. Most existing white balance methods are based on the computation of the illumination condition, also known as the white point. Therefore, these methods are highly dependent on whether the reference white is correctly estimated. We propose a white balance method to address this issue.

Instead of estimating the white point, a custom white point attempts to use the known neutral reference in the scene. The color of a white or a neutral gray object in the image is used as the reference temperature of the light source to establish the white balance. Having the user select a white point, methods such as Von Kries chromatic adaptation [152] will obtain reasonable color compensation as shown in Equation 5.3.

A chromatic adaptation transformation (CAT) mapping XYZ_1 in viewing condition 1 to XYZ_2 in viewing condition 2 to achieve a perceptual match can be formulated in

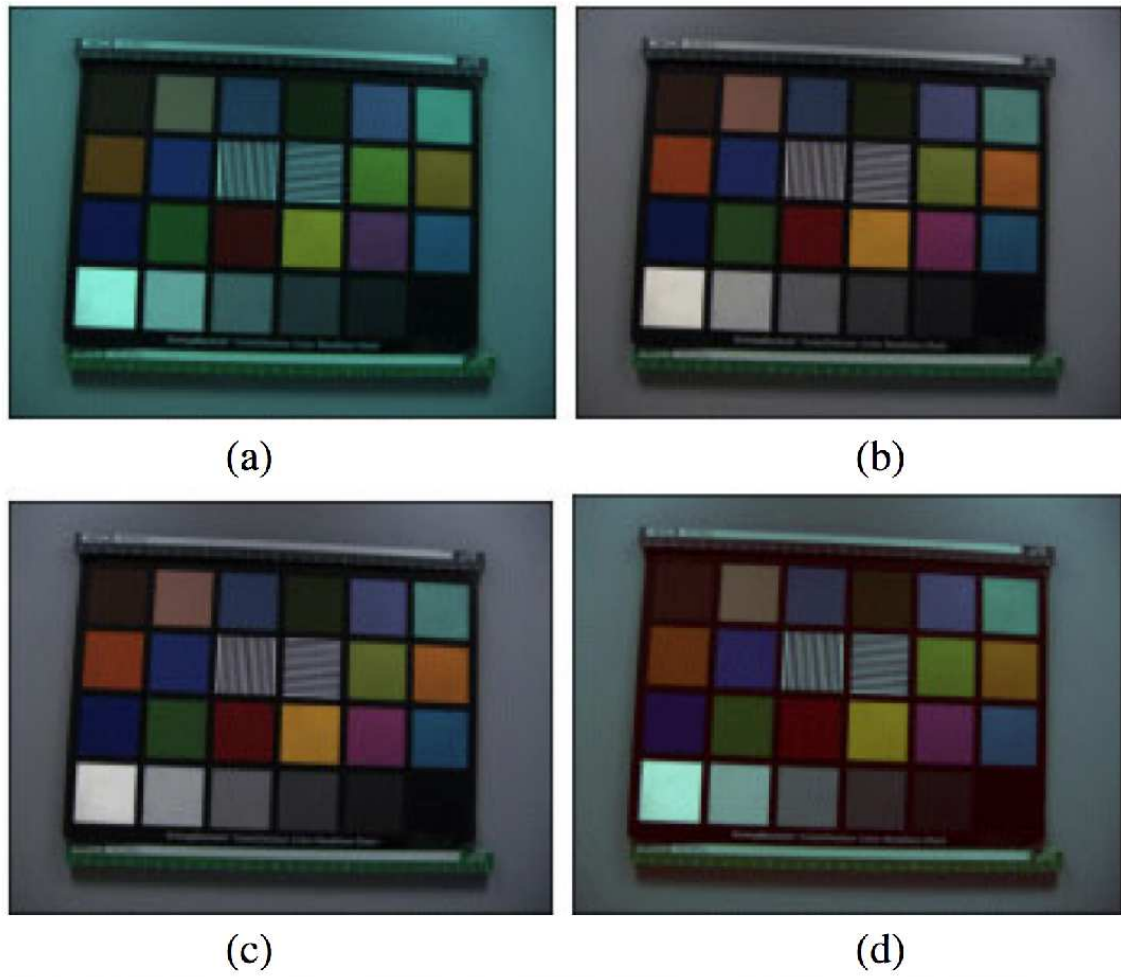


Fig. 5.3. (a) Original image and results obtained by applying (b) GWM, (c) perfect reflector, (d) fuzzy rules [144]

Equation 5.3. The white point in LMS color space is often used as $w1$ to correct the color casting of the image.

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \mathbf{H}^{-1} \begin{bmatrix} \frac{L_{w2}}{L_{w1}} & 0 & 0 \\ 0 & \frac{M_{w2}}{M_{w1}} & 0 \\ 0 & 0 & \frac{S_{w2}}{S_{w1}} \end{bmatrix} \mathbf{H} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}; \quad (5.3)$$

where LMS_{wi} , ($i = 1, 2$), is the LMS cone responses to reference white w_i . \mathbf{H} is a 3×3 non-singular matrix which represents a transformation from XYZ to LMS.

This approach typically results in improved color balance when compared with AWB, but it requires a white object in the scene which is not always the case.

Our method will address this shortcoming by pairing any arbitrary colors in the image with the corrected colors.

5.3 White Balancing With User Input

We first consider a general color correction problem where the goal is to match the output of different imaging devices to a reference device. We describe a model based method to determine a color transformation assuming that both the reference white and the given device's white points are known. Next, we describe our method for synthesizing a given device's white point with weighted combination of user specified colors.

5.3.1 Model Based Color Correction

Let an object be imaged with k cameras, possibly under k different illumination conditions. We assume that each illumination remains fairly constant over time. Thus, we can combine a camera-illumination pair into a single entity which we denote as an *imaging unit* (IU). Let the output of the k IUs at any time t be represented by $I_0(t), I_1(t), \dots, I_{k-1}(t)$. Note that each $I_r(t)$ consists of pixel-wise color coordinates for the scene. For all practical purposes, we can safely assume that the color at a pixel is specified as its RGB components in the camera output. Further, we note that it is unrealistic to assume that an object would be imaged at the same time by all the IUs. Therefore, we drop the time index in subsequent discussions.

It is natural to expect significant variation in the images (I_r) of an object generated by the different imaging units. If the object is first imaged with IU a and then by IU b , one would like to transform I_b such that the colors resemble those in I_a . Thus, we would need an RGB-RGB transform for every pair of imaging units in the system. Let \mathfrak{S} represent the

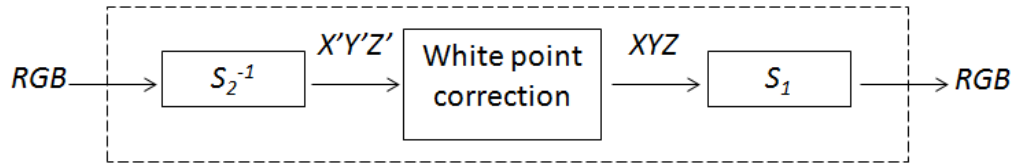


Fig. 5.4. A block diagram of the device to device transformation CCMX (Φ).

commonly used RGB color space. We require a function $\Phi : \mathfrak{S} \rightarrow \mathfrak{S}$ such that $\Phi(I_b) = I_a$, for the same shade. If the camera model is represented by S , then Figure 5.4 illustrates the construction of Φ . This method is denoted by CCMX (Camera to Camera Model-Based Transformation). The white correction step consists of a technique to account for the different illumination conditions.

We first reduce the problem by assuming a reference IU exists which represents a typical capture device under typical illumination conditions. Therefore, our goal is to match the output of every IU with that of the reference IU. The problem now scales linearly with the number of IUs. Without loss of generality we designate I_0 as the output from the reference IU.

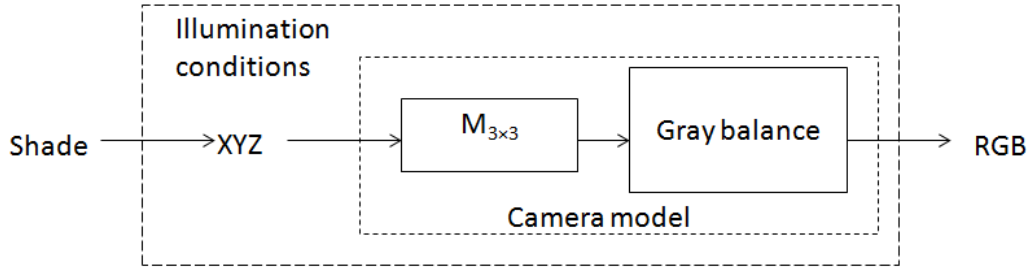


Fig. 5.5. A schematic diagram of an imaging unit.

Since the effects of illumination are abstracted into the IUs, the only invariant property of a subject would be its surface reflectance. However, to make the discussion more intuitive, we refer to this as “shade.” Shade is the identifier that a human would associate with an object with such surface reflectance and viewed under typical illumination. Thus,

we require that subjects with the same shade produce similar RGB components in all the IUs. The actual relation between the shade and the observed tri-stimuli (under given illumination) is difficult to establish and requires extensive radiometric measurements. Instead, we make this key assumption that the two quantities are related through the illumination white point. The resultant tri-stimuli in CIEXYZ would be the input to the camera which itself can be modeled with gray balancing and a linear transformation [153]. The complete model of an IU is illustrated in Figure 5.5.

An advantage of selecting a reference is that the discussion and modeling can be build for two camera systems ($k = 2$) and extended to multi-camera systems without any changes. In this section, we describe a model-based method to match the output of a test imaging unit (I_1) with that of a reference IU (I_0).

A typical imaging device takes a visual tristimulus $(X, Y, Z)^T$ as input and generates a numerical triplet (R, G, B) as output. The output follows some commonly accepted standard such as *sRGB* [149] and the internal transformations are hidden from the user (except in cameras that support RAW output). Modeling allows us to tap into any stage of the imaging pipeline in order to obtain the most fitting data format. We choose a camera model which consists of gray balancing and a linear transform. Thus, a camera model S transforms a stimulus $(X, Y, Z)^T$ to $(R, G, B)^T$ as follows:

$$\begin{bmatrix} R_n^{lin} \\ G_n^{lin} \\ B_n^{lin} \end{bmatrix} = \begin{bmatrix} & & \\ & M_{3 \times 3} & \\ & & \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (5.4)$$

Here $(R, G, B)_n^{lin}$ signify the normalized linear values and M is a linear transform. Absolute linear values can be obtained by scaling with the absolute linear values of the device white. This can also be represented using the inverse but more intuitive normalization operation,

$$R_n^{lin} = \frac{R_{abs}^{lin}}{R_{abs}^{lin w}}; G_n^{lin} = \frac{G_{abs}^{lin}}{G_{abs}^{lin w}}; B_n^{lin} = \frac{B_{abs}^{lin}}{B_{abs}^{lin w}}, \quad (5.5)$$

where $(R, G, B)_{abs}^{lin}$ represent the absolute linear values which are proportional to the photon count at the camera sensor. Finally, the device RGB output is obtained by gamma correction or gray balancing [154]. This can be represented by a 3D function

$$f : \mathbb{R}^3 \rightarrow \{0, 1, \dots, 255\}^3. \quad (5.6)$$

In our modeling, we use a gain-gamma-offset model [155] for the function f , and obtain the transformation matrix M by linear regression. The parameters of the models are computed by measuring the RGB and CIEXYZ values for a small number of printed patches. We used only 24 colored patches (shown in Figure 5.6) and measured the XYZ values with a spectroradiometer PR-705.



Fig. 5.6. A printed sheet of colored patches used to construct the camera models.

While many white point compensation techniques have been proposed in the literature [149], we use a simple rescaling based approach because it is easy to use in real situations with very little information about the illumination available. It also comprises operations similar to the von Kries chromatic adaptation theory [152] although we do not use the LMS cone space for the diagonal transforms. Let the reference illumination white point be

$(X, Y, Z)_w^0$ and the test white point be $(X, Y, Z)_w^1$. Then to transform an object's tristimulus (X', Y', Z') to the reference conditions, we obtain

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{X_w^0}{X_w^1} & 0 & 0 \\ 0 & \frac{Y_w^0}{Y_w^1} & 0 \\ 0 & 0 & \frac{Z_w^0}{Z_w^1} \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}. \quad (5.7)$$

5.3.2 White Synthesis From Arbitrary Colors

As discussed in Section 5.3 estimation of the scene's white point is the critical step in using a Von Kries-like color correction technique. We propose obtaining an estimate of the scene white by combining arbitrary colors provided by a human user as input (in the form of RGB pairs). In the following, we first describe a method to obtain user specification of colors in the scene and then present the reasoning for one combination technique to derive a corresponding weighting formula.

Consider the image pair shown in Figure 5.7. The illustration assumes that the left half of the figure is a captured image (or a preview on the imaging device) and the right half shows a collection of commonly occurring colors. If a user is presented with such an interface during image acquisition (or preview or even post- processing) and is able to select areas on either half by touch/pointing devices, some veridical color descriptors can be obtained.

Let the user input be represented by a set $\mathfrak{S} = \{I_1, I_2, \dots, I_n\}$ where n is the number of input points. Each input point consists of two color triplets:

$$I_1 = \{(r_1, g_1, b_1), (R_1, G_1, B_1)\}. \quad (5.8)$$

For this discussion we assume that a color triplet is the RGB components. Note that we will not constrain the RGB components to be real numbers or 8-bit integers, linear or non-linear (gamma corrected).

With these conventions an input point (as in Equation 5.8) consists of the RGB components of a pixel in the uncorrected image (r_1, g_1, b_1) and the corresponding veridical color on the color grid according to the user (R_1, G_1, B_1) . The veridical descriptor for

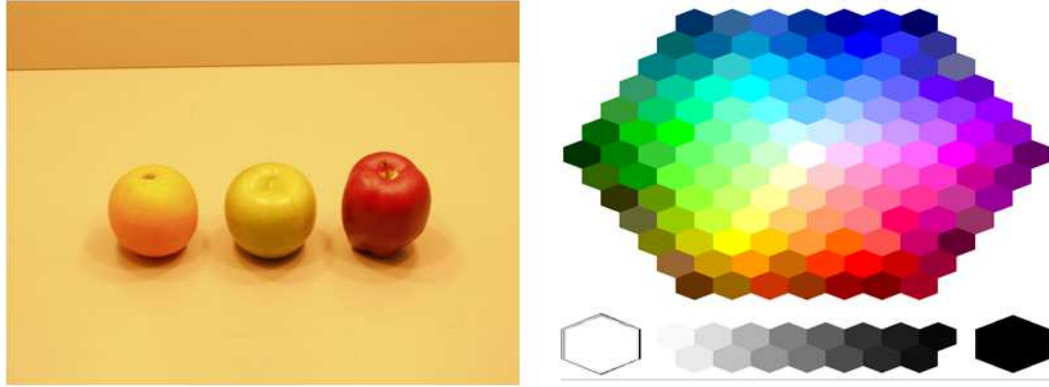


Fig. 5.7. An example of the interface for user input. Image on the left is captured and displayed alongside a grid of common colors (right) for the user to match.

a white object (R_w, G_w, B_w) should have maximum value for each channel, for example $(255, 255, 255)$ for 8-bit per channel integer RGB. However, the corresponding captured descriptor (r_w, g_w, b_w) is unknown. If a truly white target was available to be imaged (as is the case in custom white balancing) this value would be known. Our goal is to estimate (r_w, g_w, b_w) using the values in \mathfrak{S} . This estimation could then be used in the model based color correction method described earlier.

Our weighting method has the following salient features:

- The color channels are processed independent of each other.
- Within a channel, the captured values are weighted relative to each other based on the user input for that channel.
- The weighted value is extrapolated toward a maximum.

Due to independent processing of the channels, we discuss the weighting formula for any one channel. Let \mathfrak{R} be a subset of the user input \mathfrak{S} that consists of only the red channel. Thus,

$$\mathfrak{R} = \{\{r_1, R_1\}, \{r_2, R_2\}, \dots, \{r_n, R_n\}\}. \quad (5.9)$$

Each captured value (r_i) is weighted by the user input (R_i). Therefore, after relative weighting we obtain

$$\tilde{r} = \frac{r_1 R_1 + r_2 R_2 + \dots + r_n R_n}{R_1 + R_2 + \dots + R_n}. \quad (5.10)$$

This weighted quantity is scaled linearly towards the desired maximum. In simple terms if \tilde{r} corresponds to a user specified value of 128, then to obtain a user specified value of 255 we would need to double \tilde{r} . Thus,

$$\hat{r}_w = \tilde{r} \frac{R_{max}}{\frac{1}{n}(R_1 + R_2 + \dots + R_n)}, \quad (5.11)$$

where R_{max} would be 255 for 8-bit image format. Upon simplification the final formula is:

$$\hat{r}_w = \frac{255n(r_1 R_1 + r_2 R_2 + \dots + r_n R_n)}{(R_1 + R_2 + \dots + R_n)^2}. \quad (5.12)$$

The green and blue components of the estimated white can be similarly determined.

5.3.3 Deployment on Mobile Imaging Systems

Mobile telephones that have advanced computing capability are widely available. These handsets, sometimes known as smartphones, have almost the same amount of processing power as a personal computer of a few years ago. The high-performance processors, high-resolution cameras and sensitive touch screens make smartphones ideal for computational processing related to imaging.

We have exploited the unique capabilities of a mobile telephone camera by developing a tool to help users adjust the color compensation of their device. As shown in Figure 5.8, after a user starts the application, the image capture stage will be performed. After the user takes the image with the high resolution camera on the smartphone, the user can select the choice of reference colors. The user can select from hundreds of reference colors. Also, we designed some of the color selection according to the most common colors contained in camera images, such as light skin, blue sky, and foliage. Then, the user is asked to pair the colors in the image with the reference color at the lower part of the screen. A pin will display in the image with the name of the reference color once the user indicates

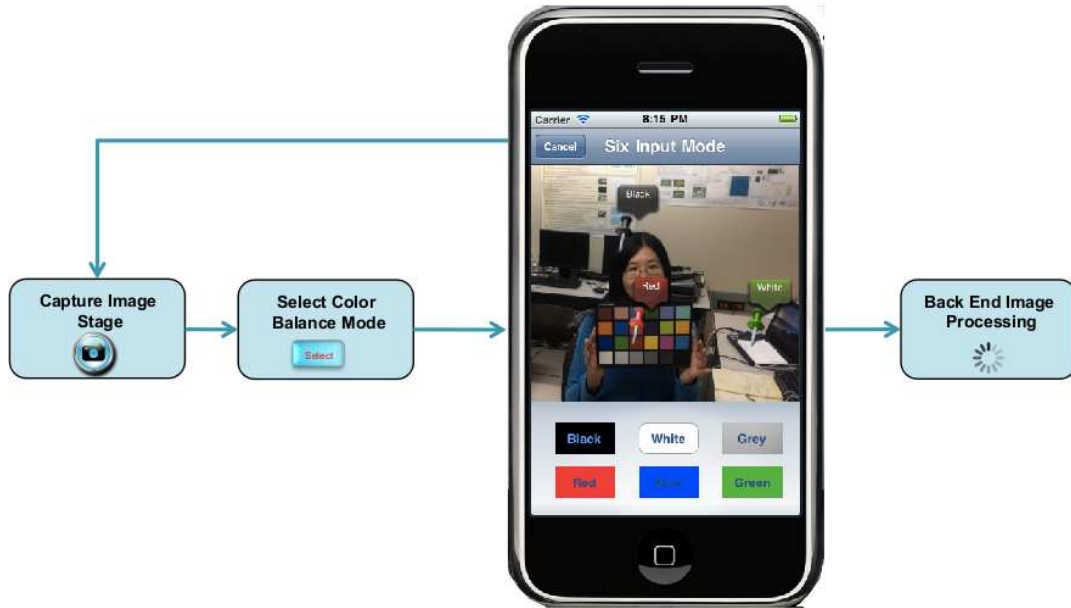


Fig. 5.8. The flow chat of the user interface.

the correspondence between them. Then, our color balancing method described above will convert the image to a more natural look. Our method is currently implemented on an iPhone running iOS 5. Our method should be easily ported to any mobile device with a digital camera and a touch screen.

5.4 Experimental Results

We tested the effectiveness of our method by color correcting images taken under different illumination conditions with multiple devices (digital still cameras, still capture from digital video cameras, and mobile telephone cameras). When taking the reference image (under D65 illumination [149]) and the uncorrected input images, the camera settings were not changed. This was needed to ensure our camera models were accurate for color space conversions. Our method is compared with images acquired with auto white balancing enabled. For these images all the camera settings were allowed to auto-adjust.

First we present the result of color correcting an image taken under two different illumination conditions by using one of them as the reference illuminant. This is illustrated in Figure 5.9. For this part of the experiment we assumed that both illuminants are known (or can be estimated with a white target). These results verify the diagonal transformation technique described in Section 5.3.1. Note that this is a subset of the graphical results presented in [156].



Fig. 5.9. Result of correcting an image with our method assuming known illumination. The columns represent the input image (left), the white corrected image (center), and the image taken under reference conditions (right).

In order to test our complete color correction method, we assume that only the reference white is known. Figure 5.10 shows the interface of our system when the user is allowed to annotate known colors in the acquired image (left column) with elements on a grid of commonly occurring colors (center). The reference image (right column) is provided for illustration only and may not be available to the user when identifying colors. The arrows are overlaid to show the actual user inputs that were used in the experiment.

Using the user inputs and the methods described in Section 5.3 the input image was processed with the goal of being visually similar to the reference image (acquired under D65 illuminant). The result is presented in Figure 5.11. It can be seen that our method results in significant similarity to the reference image even with few arbitrary user inputs. In contrast the image acquired with automatic white balance enabled does not compensate for the illuminant completely or over-corrects some channels. While the failure of auto

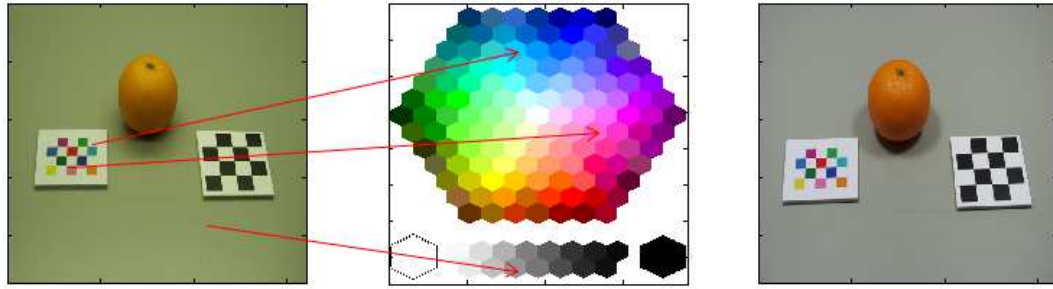


Fig. 5.10. An illustration of user-specified color matching for white synthesis. The reference image on the right is only provided for comparison.

white balance only occurs in corner cases, our method is very effective in such extreme conditions.

The experimental results show that our method achieved the best performance with varying object and varying light conditions. The draw back of AWB is that it only use the some statistics characteristic of the color distribution on the image, and sometimes it may under-correct or over-correct the color cast on the image.

The unique aspect of our method is utilizing the user input and it also opens new ways for white balancing or color correction. Our method works under all possible conditions because the user specifies color pairs - patches in the scene and veridical colors on the grid, used to compute the white point under a color cast. In this way, our method uses an interpolation technique to assign weights to the arbitrary colors which are then used to estimate the RGB components corresponding to a white target. The complexity of our proposed method is also acceptable and it is suitable to be implemented on mobile devices.

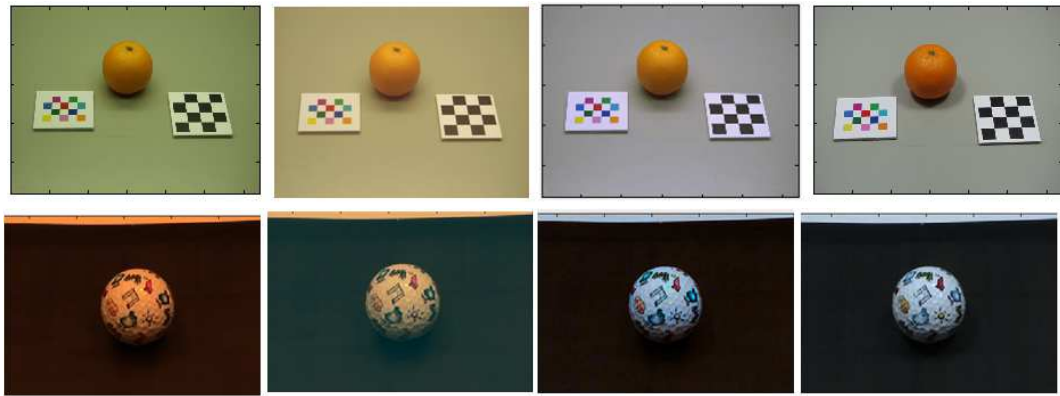


Fig. 5.11. Two examples of images color corrected with our method and user input. The columns represent (i) input image, (ii) image acquired with automatic settings, (iii) input image corrected with our method, and (iv) image under reference illumination.

6. SUMMARY AND FUTURE WORK

6.1 Conclusions

This thesis presented single-view and multi-view volume estimation methods in order to accurately estimate the nutrition intake from a meal image. We developed a single-view volume estimation method utilizing the shape dictionary and we proposed a multi-view volume estimation method using “Shape from Silhouettes.” Low complexity image quality assessment methods for fiducial marker detection, blur assessment on mobile devices and three color correction models are also developed in this thesis.

We also extended the custom white balancing technique available in many imaging devices by allowing a user to specify arbitrary colors in the scene. We derived an interpolation technique to assign weights to the arbitrary colors which are then used to estimate the RGB components corresponding to a white target. We obtained the user input by displaying a captured image alongside a color grid of commonly occurring colors. We obtained encouraging results from testing our method on images acquired under different illumination conditions. Furthermore, due to low computational requirements, our approach is very suitable for mobile devices.

The main contributions in this thesis are listed as follows:

- We proposed a novel food portion size estimation method using a single image. The single view volume estimate is implemented using a model-driven system for creating 3D reconstructions of specific food items. A pre-built or defined 3D model of a food item is projected back to the image plane. Subsequently, the portion size and degrees-of-freedom (DOFs) for the final pose is estimated by an image similarity measure.

- We described a multi-view volume estimation method to automatically estimate the food portion size. We demonstrate the use of stereo vision in order to improve the accuracy of food segmentation and volume estimation. Also, the multi-view shape recovery method is implemented using a combination of shape from silhouettes and shape from correspondence techniques.
- We proposed an approach to automatically detect the presence of the fiducial marker. The method is based on region search, which is less sensitive to illumination changes and noise than the corner or line based methods. We compare this method with these traditional methods. The proposed approach reduces the computational time for locating the corners, speeds up the image pre-processing, and adapts to automatic image quality assessment on mobile devices.
- We developed a low complexity blur metric by suitably modifying a well known method known as cumulative probability of blur detection (CPBD) which utilizes probability distribution (CPBD) of edge widths. The average computational runtime of the original CPBD method is reduced from 9 seconds to 1 second on a mobile device for 2048×1536 sized images.
- We propose three chromatic adaptation models that use more perceptually uniform color space models: a linear RGB to RGB transform, a nonlinear RGB to RGB transform, and a linear model in CIELAB color space. From experimental results, our proposed methods offer better color consistency in various illumination tests than the other well known techniques.
- We devised an intuitive method for a user to specify veridical color descriptors for color patches in a scene. This is facilitated by the availability of graphical interfaces on many mobile devices such as smartphones. A scheme to synthesize the white point descriptor using a weighted combination of the colors provided by the user is also introduced.

The methods described in this thesis have been integrated in the TADA system to complement and enhance the existing food image analysis methods.

6.2 Future Work

In the TADA mobile phone food record (mpFR) food portion estimation is crucial to automatically identifying and quantifying foods and beverages consumed based on analyzing meal images captured with a mobile device. One focus of the future work will be to improve the estimation the food portion size by exploring the use of the single view and multi-view 3D reconstruction techniques. Also, in order to automatically identify and estimate portions of foods in an image of a meal, it is crucial to obtain high quality images, which aids image analysis. Illumination detection on a mobile device and several color correction methods are also presented in this work.

Potential topics for future work include:

- Currently, the shape templates of our single view volume estimation framework only include five geometrical shapes. More shape templates need to be implemented such as “half sphere” and “half cone.” In our approach we only use minimal geometric information, typically just the shape of the object. An alternative method to reconstruct the 3D rigid object by inferring the camera translation and rotation matrix, the segmentation mask, the food label, and use this information with a hierarchical Bayesian model. Furthermore, one could investigate how unsupervised machine learning methods can be used to automatically choose or generate the volume shape model for a food item which is not in our database.
- For multi-view geometric reconstruction, one could explore the use of other 3D reconstruction techniques (e.g. shape from correspondence). For the “shape from silhouette” method to work several constraints must be satisfied: the image sequence must be taken from different viewing angles (at least 6 views) such as a turn-table. All of the intrinsic and extrinsic camera parameters need to be determined for each image. The segmentation mask of the food items for each image must be extracted

and the segmentation error has to be small and there must be no hole or self occlusion for the reconstructed objects. Our current multi-view 3D reconstruction techniques have only been tested in a controlled laboratory environment. Next steps may include making the method more robust to the segmentation errors/noise. This can be achieved by using different segmentation methods (e.g. active contours) to refine the segmentation mask or assign a validation score to each voxel. Second, find an alternative method to estimate the camera motion without the use of the checkerboard. The checkerboard is sometimes not present in the image scene as the user may forget to place the checkerboard, or the checkerboard is outside the field of view. We should still be able to calculate the intrinsic camera matrix using point correspondence and the “RANdom SAmple Consensus” (RANSAC) method [84]. Without the checkerboard, there will also be a scale ambiguity with the motion estimation and 3D reconstruction. This global scale ambiguity can be addressed by computing the motion of a third camera given the reconstructed structure from frames one and two. This technique, commonly referred as the P_nP problem [157], where n is the number of images obtained. Finally, with some prior information, one could handle missing information, as it often occurs with the images obtained by our acquisition systems.

- Stereo view and three view geometric reconstruction should be further studied. Our problem is described as follows: Given two, three, or many images of a scene, taken from different point of views, reconstruct the 3D structure of the scene along with the intrinsic camera matrix, translation and rotation vectors. This problem is also referred to as “shape from motion.” When only given two or three views of the scene, the “shape from silhouette” method we used to automatically recover the 3D shape of the objects will produce many errors. Stereo matching methods, which use the segments [158], local and affine invariant regions [159], and feature points [160] for image reconstruction will be a very attractive topic to investigate. As shown in Figure 6.1, to recover the 3D shape of the food items, we need to use all the

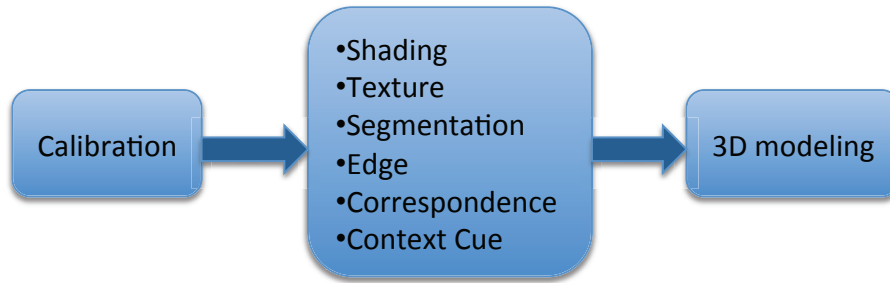


Fig. 6.1. A future 3D reconstruction system.

available cues: material shading, texture gradients, contour of the object, edges, point correspondence, and context cue.

- In our TADA image quality assessment system, the image quality metrics are computed on the mobile phone. However, the user has the option to save and send the images with poor quality. One should investigate methods to recover the image quality from artifacts (e.g. blur, noise, and insufficient lighting) in the poor quality images.
- The proposed methods for color correction are deployed in the TADA system and are being tested under various lighting conditions. In the future, one might want to study the use of the gamma for each specific color channel and specific device and by exploring other factors (e.g. the reflection properties of the object, or the spectral sampling properties of the capturing device) that affect the color appearance model.
- Our proposed white balancing technique is very suitable for mobile devices because most such devices are equipped with only moderately sophisticated imaging systems and our method allows better color capture with minimal common-sense user input. One could develop a new chromatic adaptation model incorporating multiple input from users. One can implement our method on these devices since many such devices have built-in tools for graphical user input. As a result, our method can be useful in photography and image analysis applications.

6.3 Publications Resulting From This Work

Journal Papers

1. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Volume Estimation with Application in Dietary Assessment," *IEEE Transactions on Information Technology in Biomedicine*, in preparation.
2. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Image Classification with Contextual Dietary Information," *IEEE Transactions on Multimedia*, in preparation.
3. **Chang Xu**, Satyam Srivastava, and Edward J. Delp, "User Assisted White Synthesis on Mobile Device," *Journal of Imaging Science and Technology*, in preparation.

Conference Papers

1. **Chang Xu**, Ye He, Albert Parra Pozo, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Image-Based Food Volume Estimation," *Proceedings of ACM International Conference on Multimedia*, Barcelona, Spain, October 2013, pp.75-80.
2. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Model-based food volume estimation using 3D pose," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2534-2538.
3. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Context based food image analysis," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2748-2752.
4. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of the IEEE International Conference on Multimedia and Expo*, San Jose, CA, July 2013, pp.1-6.

5. **Chang Xu**, Fengqing Zhu, Nitin Khanna, Carol J. Boushey, Edward 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. 82960Q110, San Francisco Airport, California, January, 2012.
6. **Chang Xu**, Nitin Khanna, Carol J. Boushey, Edward J. Delp, “Low Complexity Image Quality Measures for Dietary Assessment Using Mobile Devices,” *Proceedings of the IEEE International Symposium on Multimedia*, Dana Point, California, December, 2012, pp. 351–356.
7. Satyam Srivastava, **Chang Xu**, Edward J. Delp, “White synthesis with user input for color balancing on mobile camera system,” *Proceedings of the IS&T/SPIE Conference on Multimedia on Mobile Devices 2012*, Vol. 8304, pp. 83 040F110, San Francisco Airport, California, January, 2012.

LIST OF REFERENCES

LIST OF REFERENCES

- [1] M. B. E. Livingstone, P. J. Robson, and J. M. W. Wallace, "Issues in dietary intake assessment of children and adolescents," *British Journal of Nutrition*, vol. 92, pp. S213–S222, October 2004.
- [2] M. U. Waling and C. L. Larsson, "Energy intake of Swedish overweight and obese children is underestimated using a diet history interview," *Journal of Nutrition*, vol. 139, no. 3, pp. 522–527, March 2009.
- [3] C. J. Boushey, D. A. Kerr, J. Wright, K. D. Lutes, D. S. Ebert, and E. J. Delp, "Use of technology in children's dietary assessment," *European Journal of Clinical Nutrition*, vol. 63, pp. S50–S57, February 2009.
- [4] F. Zhu, "Multilevel image segmentation with application in dietary assessment and evaluation," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, December 2011.
- [5] M. Bosch, "Visual feature modeling and refinement with application in dietary assessment," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, May 2012.
- [6] F. Zhu, A. Mariappan, D. Kerr, C. Boushey, K. Lutes, D. Ebert, and E. J. Delp, "Technology-assisted dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VI*, vol. 6814, San Jose, CA, January 2008.
- [7] A. Mariappan, M. Bosch, F. Zhu, C. J. Boushey, D. A. Kerr, D. S. Ebert, and E. J. Delp, "Personal dietary assessment using mobile devices," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VII*, vol. 7246, San Jose, CA, January 2009.
- [8] F. Zhu, M. Bosch, I. Woo, S. Kim, C. J. Boushey, D. S. Ebert, and E. J. Delp, "The use of mobile devices in aiding dietary assessment and evaluation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 756–766, August 2010.
- [9] Z. Ahmad, N. Khanna, C. Boushey, and E. Delp, "A mobile phone user interface for image-based dietary assessment," *Proceedings of the IS&T/SPIE Conference on Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications*, vol. 9030, San Francisco, CA, February 2014, pp. 903 007–1–9.
- [10] C. Xu, Y. He, A. Parra, N. Khanna, C. Boushey, and E. Delp, "Image-based food volume estimation," *Proceedings of ACM International Conference on Multimedia*, Barcelona, Spain, October 2013, pp. 75–80.
- [11] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Context based food image analysis," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp. 2748 – 2752.

- [12] Y. He, "Context based image analysis with application in dietary assessment and evaluation," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, May 2014.
- [13] F. E. Thompson, A. F. Subar, C. M. Loria, J. L. Reedy, and T. Baranowski, "Need for technological innovation in dietary assessment," *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 48–51, February 2010.
- [14] F. Thompson and A. Subar, *Nutrition in the Prevention and Treatment of Disease*, 2nd ed. Burlington, MA, USA: Elsevier Academic Press, 2008, ch. Dietary assessment methodology, pp. 3–39.
- [15] (2010) Weight Loss Report at Myfooddiary.com. MyFoodDiary.com. [Online]. Available: http://www.myfooddiary.com/weight_loss_reports.asp
- [16] (2009, Sep) Inside the Pyramid - What are discretionary calories? United States Department of Agriculture. [Online]. Available: http://www.mypyramid.gov/pyramid/discretionary_calories.html
- [17] (2010) Dietorganizer PC. MulberrySoft. [Online]. Available: <http://www.dietorganizer.com/>
- [18] K. Kitamura, T. Yamasaki, and K. Aizawa, "Food log by analyzing food images," *Proceedings of the ACM International Conference on Multimedia*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 999–1000.
- [19] —, "Foodlog: capture, analysis and retrieval of personal food images via web," *Proceedings of the ACM Workshop on Multimedia for Cooking and Eating Activities*, New York, USA, November 2009, pp. 23–30.
- [20] E. Årsand, J. T. Tufano, J. D. Ralston, and P. Hjortdahl, "Designing mobile dietary management support technologies for people with diabetes," *Journal of telemedicine and telecare*, vol. 14, no. 7, pp. 329–332, June 2008.
- [21] B. L. Six, T. E. Schap, F. Zhu, A. Mariappan, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, "Evidence-based development of a mobile telephone food record," *Journal of American Dietetic Association*, vol. 110, pp. 74–79, January 2010.
- [22] T. R. E. Schap, B. L. Six, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, "Adolescents in the united states can identify familiar foods at the time of consumption and when prompted with an image 14 h postprandial, but poorly estimate portions," *Public Health Nutrition*, vol. 1, no. 1, pp. 1–8, July 2011.
- [23] B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. Zhu, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, "Novel technologies for assessing dietary intake: Evaluating the usability of a mobile telephone food record among adults and adolescents," *Journal of Medical Internet Research*, vol. 14, no. 2, pp. 156–167, April 2012.
- [24] D. A. Kerr, C. Pollard, P. A. Howat, E. J. Delp, M. Pickering, K. R. Kerr, S. S. Dhaliwal, I. S. Pratt, J. Wright, and C. J. Boushey, "Connecting health and technology (chat): Protocol of a randomized controlled trial to improve nutrition behaviours using mobile devices and tailored text messaging in young adults," *BMC public health*, vol. 12, no. 447, pp. 1–10, June 2012.

- [25] F. Zhu, M. Bosch, N. Khanna, C. J. Boushey, and E. J. Delp, "Multiple hypotheses image segmentation and classification with application to dietary assessment," *IEEE Journal of Biomedical and Health Informatics*, vol. 99, pp. 1–15, February 2014.
- [26] I. Woo, K. Otsmo, S. Kim, D. S. Ebert, E. J. Delp, and C. J. Boushey, "Automatic portion estimation and visual refinement in mobile dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VIII*, vol. 7533, San Jose, CA, January 2010, pp. 75 330O1–10.
- [27] J. Chae, I. Woo, S. Kim, R. Maciejewski, F. Zhu, E. J. Delp, C. J. Boushey, and D. S. 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, San Francisco, California, USA, January 2011, pp. 78 730K1–8.
- [28] "USDA food and nutrient database for dietary studies, 1.0." Beltsville, MD: Agricultural Research Service, Food Surveys Research Group, 2004.
- [29] M. Bosch, T. Schap, F. Zhu, N. Khanna, C. Boushey, and E. Delp, "Integrated database system for mobile dietary assessment and analysis," *Proceedings of the 1st IEEE International Conference Workshop on Multimedia Services and Technologies for E-health in conjunction with the International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011, pp. 1 – 6.
- [30] M. R. Okos and C. Boushey, "Density standards meeting organized by purdue university: A report," *International Journal of Food Properties*, vol. 15, no. 2, pp. 467–470, February 2012.
- [31] S. Kelkar, S. Stella, C. Boushey, and M. Okos, "Developing novel 3D measurement techniques and prediction method for food density determination," *Procedia Food Science*, vol. 1, pp. 483–491, May 2011.
- [32] 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, San Francisco, USA, 2012, pp. 82 960Q1–10.
- [33] W. Jia, Y. Yue, J. Fernstrom, Z. Zhang, Y. Yang, M. Sun, *et al.*, "3D localization of circular feature in 2d image and application to food volume estimation," *Proceeding of 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, San Diego, CA, USA, March 2012, pp. 4545–4548.
- [34] B. L. Six, T. E. Schap, D. A. Kerr, and C. J. Boushey, "Evaluation of the food and nutrient database for dietary studies for use with a mobile telephone food record," *Journal of Food Composition and Analysis*, vol. 24, pp. 1160–1167, 2011.
- [35] M. Bosch, F. Zhu, N. Khanna, C. J. Boushey, and E. J. Delp, "Combining global and local features for food identification and dietary assessment," *Proceedings of the International Conference on Image Processing*, Brussels, Belgium, December 2011, pp. 1789–1792.
- [36] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, "Recognition and volume estimation of food intake using a mobile device," *Proceedings of the Workshop on Applications of Computer Vision*, Snowbird, UT, USA, December 2009, pp. 1 –8.

- [37] N. Khanna, C. J. Boushey, D. Kerr, M. Okos, D. S. Ebert, and E. J. Delp, "An overview of the technology assisted dietary assessment project at purdue university," *Proceedings of the IEEE International Symposium on Multimedia*, Taichung, Taiwan, December 2010, pp. 290–295.
- [38] F. Zhu, M. Bosch, and E. J. Delp, "An image analysis system for dietary assessment and evaluation," *Proceedings of the IEEE International Conference on Image Processing*, Hong Kong, China, September 2010, pp. 1853–1856.
- [39] Y. He, N. Khanna, C. Boushey, and E. Delp, "Image segmentation for image-based dietary assessment: A comparative study," *Proceedings of the IEEE International Symposium on Signals, Circuits & Systems*, Iasi, Romania, July 2013, pp. 1–4.
- [40] Y. He, C. Xu, N. Khanna, C. Boushey, and E. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of IEEE International Conference on Multimedia and Expo*, San Jose, CA, July 2013, pp. 1–10.
- [41] Y. He, N. Khanna, C. Boushey, and E. Delp, "Snakes assisted food image segmentation," *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, Banff, Canada, September 2012, pp. 181–185.
- [42] M. Bosch, F. Zhu, N. Khanna, C. Boushey, and E. Delp, "Combining global and local features for food identification and dietary assessment," *Proceedings of the International Conference on Image Processing*, Brussels, Belgium, September 2011, pp. 1789 – 1792.
- [43] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Providence, RI, USA, August 2011, pp. 1297 – 1304.
- [44] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "Structure from motion without correspondence," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Hilton Head Island, SC, USA, June 2000, pp. 557–564.
- [45] S. Liu, K. Kang, J. Tarel, and D. Cooper, "Free-form object reconstruction from silhouettes, occluding edges and texture edges: A unified and robust operator based on duality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 131–146, January 2008.
- [46] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, July 2000.
- [47] E. Prados and O. Faugeras, "Shape from shading," *Handbook of mathematical models in computer vision*. Springer, 2006, pp. 375–388.
- [48] M. Pollefeys, D. Nistér, J. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, *et al.*, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2, pp. 143–167, July 2008.
- [49] B. L. Daugherty, T. E. Schap, R. Ettienne-Gittens, F. Zhu, M. Bosch, E. J. Delp, D. S. Ebert, D. A. Kerr, and C. J. Boushey, "Novel technologies for assessing dietary intake: Evaluating the usability of a mobile telephone food record among adults and adolescents," *Journal of Medical Internet Research*, vol. 14, no. 2, p. e58, April 2012.

- [50] W. Jia, Y. Yue, J. Fernstrom, N. Yao, R. Scabassi, M. Fernstrom, and M. Sun, "Corrigendum to imaged based estimation of food volume using circular referents in dietary assessment," *Journal of food engineering*, vol. 110, no. 3, pp. 76–86, March 2012.
- [51] F. Kong and J. Tan, "Dietcam: Automatic dietary assessment with mobile camera phones," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 147–163, February 2012.
- [52] C. D. Lee, J. Chae, T. E. Schap, D. A. Kerr, E. J. Delp, D. S. Ebert, and C. J. Boushey, "Comparison of known food weights with image-based portion-size automated estimation and adolescents' self-reported portion size," *Journal of diabetes science and technology*, vol. 6, no. 2, pp. 428–434, March 2012.
- [53] H. Chen, W. Jia, Z. Li, Y. Sun, and M. Sun, "3D/2D model-to-image registration for quantitative dietary assessment," *Proceedings of 38th Annual Northeast Bioengineering Conference (NEBEC)*, Philadelphia, PA, USA, March 2012, pp. 95–96.
- [54] F. Kong and J. Tan, "Dietcam: Regular shape food recognition with a camera phone," *Proceedings of the International Conference on Body Sensor Networks (BSN)*, Dallas, TX, USA, May 2011, pp. 127–132.
- [55] M. Sun, J. D. Fernstrom, W. Jia, S. A. Hackworth, N. Yao, Y. Li, C. Li, M. H. Fernstrom, and R. J. Scabassi, "A wearable electronic system for objective dietary assessment," *Journal American Dietetic Association*, vol. 110, pp. 45–47, January 2010.
- [56] J. Shang, M. Duong, E. Pepin, X. Zhang, K. Sandara-Rajan, A. Mamishev, and A. Kristal, "A mobile structured light system for food volume estimation," *Proceeding of IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain, November 2011, pp. 100–101.
- [57] F. Zhu, M. Bosch, N. Khanna, C. Boushey, and E. Delp, "Multilevel segmentation for food classification in dietary assessment," *Proceedings of the 7th International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, September 2011, pp. 337–342.
- [58] C. Xu, F. Zhu, N. Khanna, C. Boushey, and E. 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, San Francisco, USA, February 2012, pp. 82 960Q1–10.
- [59] C. Xu, Y. He, N. Khanna, C. Boushey, and E. Delp, "Model-based food volume estimation using 3D pose," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp. 2534 – 2538.
- [60] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, August 2000.
- [61] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [62] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, January 2006.

- [63] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, June 1997, pp. 1106–1112.
- [64] "Jeita cp-3451 exchangeable image file format for digital still cameras: Exif version 2.2," Japan Electronics and Information Technology Industries Association, April 2002.
- [65] M. Prasad and A. Fitzgibbon, "Single view reconstruction of curved surfaces," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, New York, NY, USA, June 2006, pp. 1345–1354.
- [66] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, Zurich, Switzerland, August 1999, pp. 434–441.
- [67] A. Blake and A. Zisserman, *Visual reconstruction*. MIT press Cambridge, 1987, vol. 2.
- [68] D. Terzopoulos, "The computation of visible-surface representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 417–438, July 1988.
- [69] A. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Math. Dokl.*, vol. 5, 1963, pp. 1035–1038.
- [70] R. L. Stevenson, "Invariant reconstruction of curves and surfaces with discontinuities with applications in computer vision," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, August 1990.
- [71] R. L. Stevenson and E. J. Delp, "Viewpoint invariant recovery of visual surfaces from sparse data," *Proceedings of International Conference on Computer Vision*, Osaka, Japan, December 1990, pp. 309–312.
- [72] R. L. Stevenson, B. E. Schmitz, and E. J. Delp, "Discontinuity preserving regularization of inverse visual problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 455–469, March 1994.
- [73] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 17–45, August 1981.
- [74] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 53–69, January 2008.
- [75] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, April 2002.
- [76] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, June 2007, pp. 1–8.

- [77] D. Min, J. Lu, and M. Do, "Joint histogram based cost aggregation for stereo matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2539–2545, January 2013.
- [78] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, June 2011, pp. 3017–3024.
- [79] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, April 1997.
- [80] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, November 1992.
- [81] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, December 2008.
- [82] K. Ostmo, "Automatic portion estimation and visual refinement in mobile dietary assessment," Master's thesis, Purdue University, May 2009.
- [83] L. Guibas and J. Stolfi, "Primitives for the manipulation of general subdivisions and the computation of voronoi," *ACM Transactions on Graphics (TOG)*, vol. 4, no. 2, pp. 74–123, April 1985.
- [84] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
- [85] C. Foshée, "Goal-driven three-dimensional object inspection from limited view backprojection reconstruction," Ph.D. dissertation, Purdue University, West Lafayette, IN, USA, December 1991.
- [86] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, November 2004.
- [87] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New York, NY, August 1996, pp. 303–312.
- [88] F. S. Hill Jr and S. M. Kelley, *Computer Graphics Using OpenGL*, 3rd ed. Pearson Press, 2006.
- [89] C. J. Boushey, D. A. Kerr, J. Wright, K. D. Lutes, D. S. Ebert, and E. J. Delp, "Use of technology in children's dietary assessment," *European Journal of Clinical Nutrition*, vol. 63 Suppl 1, pp. S50–57, February 2009.
- [90] C. Boushey, D. Kerr, T. Schap, and B. Daugherty, "Importance of user interaction with automated dietary assessment methods," *European Journal of Clinical Nutrition*, vol. 66, no. 5, p. 648, May 2012.

- [91] J. Fan, D. K. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1454–1466, August 2001.
- [92] R. Pohle and K. D. Toennies, "Segmentation of medical images using adaptive region growing," *Proceedings of Medical Imaging 2001*, vol. 4322, San Diego, CA, USA, July 2001, pp. 1337–1346.
- [93] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, January 1988.
- [94] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," *Proceedings of Eighth IEEE International Conference on Computer Vision*, vol. 1, Vancouver, BC, July 2001, pp. 105–112.
- [95] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE transactions on Image processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [96] M. Rahman, *Food properties handbook*, 2nd ed. Boca Raton, FL, USA: CRC Press, Taylor & Francis Group, 2009.
- [97] C. Li, J. Fernstrom, R. Scabassi, M. Fernstrom, W. Jia, and M. Sun, "Food density estimation using fuzzy logic inference," *Proceedings of the IEEE 36th Annual Northeast Bioengineering Conference*, New York, NY, March 2010, pp. 1 – 2.
- [98] J. Rodríguez-Ramírez, L. Méndez-Lagunas, A. López-Ortiz, and S. S. Torres, "True density and apparent density during the drying process for vegetables and fruits: A review," *Journal of food science*, vol. 77, no. 12, pp. R146–R154, November 2012.
- [99] M. Bosch, F. Zhu, T. Schap, C. J. Boushey, D. Kerr, N. Khanna, and E. J. Delp, "An integrated image-based food database system with application in dietary assessment," *Presentation at the 2010 mHealth Summit*, Washington, DC, November 2010.
- [100] (2010, November) United States Department of Agriculture, Report of the Dietary Guidelines Advisory Committee on the Dietary Guidelines for Americans. [Online]. Available: <http://www.cnpp.usda.gov/DGAs2010-DGACReport.htm>
- [101] B. W. Keelan and H. Urabe, "ISO 20462: a psychophysical image quality measurement standard," *Proceedings of the IS&T/SPIE Conference on Image Quality and System Performance*, vol. 5294, San Jose, CA, December 2003, pp. 181–189.
- [102] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.
- [103] Z. Wang and E. P. Simoncelli, "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model," *Proceedings of the IS&T/SPIE Conference on Human Vision and Electronic Imaging*, vol. 5666, San Jose, CA, 2005, pp. 149–159.
- [104] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, October 2006.

- [105] J. Bouguet, "Camera calibration toolbox for matlab," 2004.
- [106] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, 1st ed. O'Reilly Media, 2008.
- [107] M. Ruffi, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, Nice, France, September 2008, pp. 3121–3126.
- [108] Z. Wang, Z. Wang, and Y. Wu, "Recognition of corners of planar pattern image," *Proceedings of the World Congress on Intelligent Control and Automation*, Jinan, China, July 2010, pp. 6342–6346.
- [109] A. de la Escalera and J. M. Armingol, "Automatic Chessboard Detection for Intrinsic and Extrinsic Camera Parameter Calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, March 2010.
- [110] V. N. Dao and M. Sugimoto, "A Robust Recognition Technique for Dense Checkerboard Patterns," *Proceedings of the International Conference on Pattern Recognition*, Istanbul, Turkey, August 2010, pp. 3081–3084.
- [111] W. Sun, X. Yang, S. Xiao, and W. Hu, "Robust checkerboard recognition for efficient nonplanar geometry registration in projector-camera systems," *Proceedings of the ACM/IEEE International Workshop on Projector camera systems*, vol. 1, no. 212, Marina del Rey, CA, 2008, pp. 2.1–2.7.
- [112] V. Vezhnevets, "OpenCV calibration object detection," October 2005. [Online]. Available: <http://graphicon.ru/oldgr/en/research/calibration/opencv.html>
- [113] R. M. Haralock and L. G. Shapiro, *Computer and robot vision*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., 1992, vol. 1.
- [114] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, 1993.
- [115] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, December 1973.
- [116] M. Ebrahimi Moghaddam and M. Jamzad, "Motion blur identification in noisy images using mathematical models and statistical measures," *Pattern recognition*, vol. 40, no. 7, pp. 1946–1957, 2007.
- [117] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing*, vol. 3, no. 4, pp. 143–152, 2009.
- [118] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, June 2004, pp. 506–513.
- [119] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

- [120] N. Joshi, S. Kang, C. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 1–9, July 2010.
- [121] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1–9, July 2007.
- [122] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transaction on Graphics*, vol. 27, no. 3, pp. 73–83, August 2008.
- [123] 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*, vol. 2, Kobe, Japan, October 1999, pp. 386–390.
- [124] 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, February 2004.
- [125] 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, September 2011.
- [126] G. Sharma, *Digital Color Imaging Handbook*. Boca Raton, Florida: CRC Press, 2002.
- [127] 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.
- [128] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons, "Moment invariants for recognition under changing viewpoint and illumination," *Computer Vision and Image Understanding*, vol. 94, no. 1-3, pp. 3–27, April/May/June 2004.
- [129] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Transactions on Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, September/October 2001.
- [130] G. D. Finlayson, S. D. Hordley, and P. M. 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.
- [131] H. Siddiqui and C. Bouman, "Hierarchical color correction for camera cell phone images," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2138–2155, November 2008.
- [132] K. Barnard, V. Cardei, and B. Funt, "A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 972–984, September 2002.
- [133] Y. J. Choi, Y. B. Lee, and W. D. Cho, "Color correction for object identification from images with different color illumination," *Proceedings of the International Joint Conference on INC, IMS and IDC*, Seoul, Korea, August 2009, pp. 1598–1603.
- [134] S. Srivastava, E. J. Delp, T. H. Ha, and J. P. Allebach, "Color management using optimal three-dimensional look-up tables," *Journal of Imaging Science and Technology*, vol. 54, no. 3, pp. 30 402–1–30 402–14, May-June 2010.

- [135] J. J. McCann, "Color spaces for color-gamut mapping," *Journal of Electronic Imaging*, vol. 8, no. 4, pp. 354–364, October 1999.
- [136] J. von Kries, "Chromatic adaptation," *Festschrift der Albrecht-Ludwigs-Universit*, pp. 145–158, 1902.
- [137] B. A. Wandell, *Foundations of vision*. Sunderland, MA, US: Sinauer Associates, 1995.
- [138] G. D. Finlayson and S. Susstrunk, "Performance of a chromatic adaptation transform based on spectral sharpening," *Proceedings of the IS and T/SID Color Imaging Conference*, Scottsdale, AZ, November 2000, pp. 49–55.
- [139] R. W. G. Hunt, *The Reproduction of Color*. Hoboken, New Jersey: Wiley, 2004.
- [140] M. D. Fairchild, *Color Appearance Models*. Chichester, UK: Wiley, 2005.
- [141] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet-SRGB," *Microsoft and Hewlett-Packard Joint Report*, 1996.
- [142] G. M. Johnson and M. D. Fairchild, "Visual psychophysics and color appearance," *Digital color imaging handbook*, CRC Press, pp. 115–172, 2003.
- [143] D. Pascale, "RGB coordinates of the macbeth color checker," The BabelColor Company, 2006. [Online]. Available: http://www.babelcolor.com/main_level/ColorChecker.htm
- [144] V. Chikane and C. Fuh, "Automatic white balance for digital still cameras," *Journal of Information Science and Engineering*, vol. 22, no. 3, pp. 497–509, May 2006.
- [145] G. Buchsbaum, "A spatial processor model for object colour perception," *Journal of the Franklin Institute*, vol. 310, no. 1, pp. 1–26, July 1980.
- [146] J. Chiang and F. Al-Turkait, "Color balancing experiments with the HP-photo smart-C30 digital camera," *PSYCH221/EE362 course project, Department of Psychology, Stanford University, USA*, 1999.
- [147] 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 2012*, vol. 8304, San Francisco, CA, 2012, pp. 83 040F1–10.
- [148] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, January 2005.
- [149] G. Sharma, Ed., *Digital Color Imaging Handbook*. Boca Raton, Florida: CRC Press, 2002.
- [150] Y. Liu, W. Chan, and Y. Chen, "Automatic white balance for digital still camera," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 460–466, August 1995.
- [151] C. Weng, H. Chen, and C. Fuh, "A novel automatic white balance method for digital still cameras," *Proceedings of the IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 3801–3804.

- [152] S. Lee Guth, "Model for color vision and light adaptation," *Journal of the Optical Society of America A*, vol. 8, no. 6, pp. 976–993, June 1991.
- [153] F. M. Verdu, J. Pujol, and P. Capilla, "Characterization of a digital camera as an absolute tristimulus colorimeter," *Journal of Imaging Science and Technology*, vol. 47, no. 4, pp. 279–295, July-August 2003.
- [154] C. Poynton, *Digital Video and HDTV: Algorithms and Interfaces*. San Fransisco, California: Morgan Kaufmann, 2003.
- [155] C. Yang-Ho, I. M. Hye-Bong, and H. A. Yeong-Ho, "Inverse characterization method of alternate gain-offset-gamma model for accurate color reproduction in display devices," *Journal of Imaging Science and Technology*, vol. 50, no. 2, pp. 139–148, March-April 2006.
- [156] S. Srivastava, K. K. Ng, and E. J. Delp, "Color correction for object tracking across multiple cameras," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 1821–1824.
- [157] D. DeMenthon and L. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100–1105, August 1992.
- [158] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," *Proceedings of the International Conference on Pattern Recognition*, vol. 3, Hong Kong, China, September 2006, pp. 15–18.
- [159] T. Tuytelaars and V. G. L., "Wide baseline stereo matching based on local, affine invariant regions," *Proceedings of the British Machine Vision Conference*, Bristol, UK, September 2000, pp. 412 – 422.
- [160] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, USA, July 2005, pp. 1075–1082.

VITA

VITA

Chang Xu was born in Baishan, Jilin Province, China. She obtained her Bachelor of Science degree from Beijing Institute of Technology (BIT) in 2007. Chang obtained her M.S. in Engineering from Purdue University Calumet in 2009. She joined the Ph.D. program at Purdue University in 2009.

Since 2009, she has served as a Research Assistant in the Video and Image Processing Laboratory (VIPER). Her thesis advisor, Professor Edward J. Delp, is the Charles William Harrison Distinguished Professor of Electrical and Computer Engineering. While in the graduate program, she worked on projects sponsored by the US National Institutes of Health (NIH).

During the summer of 2012, she was a Research Intern working on Augmented Reality at Qualcomm Research, San Diego, California.

Her current research interests include computer vision, image and video processing, image analysis, machine learning, color science and medical imaging.

She is a student member of the IEEE and the IEEE Signal Processing Society.

Chang Xu's publications from this research work include:

Journal Papers

1. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Volume Estimation with Application in Dietary Assessment," *IEEE Transactions on Information Technology in Biomedicine*, in preparation.
2. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food Image Classification with Contextual Dietary Information," *IEEE Transactions on Multimedia*, in preparation.
3. **Chang Xu**, Satyam Srivastava, and Edward J. Delp, "User Assisted White Synthesis on Mobile Device," *Journal of Imaging Science and Technology*, in preparation.

Conference Papers

1. **Chang Xu**, Ye He, Albert Parra Pozo, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Image-Based Food Volume Estimation," *Proceedings of ACM International Conference on Multimedia*, Barcelona, Spain, October 2013, pp.75-80.
2. **Chang Xu**, Ye He, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Model-based food volume estimation using 3D pose," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2534-2538.
3. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Context based food image analysis," *Proceedings of IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013, pp.2748-2752.
4. Ye He, **Chang Xu**, Nitin Khanna, Carol J. Boushey, and Edward J. Delp, "Food image analysis: Segmentation, identification and weight estimation," *Proceedings of the IEEE International Conference on Multimedia and Expo*, San Jose, CA, July 2013, pp.1-6.
5. **Chang Xu**, Fengqing Zhu, Nitin Khanna, Carol J. Boushey, Edward 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. 82960Q110, San Francisco Airport, California, January, 2012.

6. **Chang Xu**, Nitin Khanna, Carol J. Boushey, Edward J. Delp, “Low Complexity Image Quality Measures for Dietary Assessment Using Mobile Devices,” *Proceedings of the IEEE International Symposium on Multimedia*, Dana Point, California, December, 2012, pp. 351–356.
7. Satyam Srivastava, **Chang Xu**, Edward J. Delp, “White synthesis with user input for color balancing on mobile camera system,” *Proceedings of the IS&T/SPIE Conference on Multimedia on Mobile Devices 2012*, Vol. 8304, pp. 83 040F110, San Francisco Airport, California, January, 2012.