Graduate School Form 9
(Revised 3/11)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Fengqing Zhu

Entitled

Multilevel Image Segmentation with Application in Dietary Assessment and Evaluation

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

1. ______________________ 5. ______________________

Chair

2. David S Ebert 6. ______________________

3. ______________________ 7. ______________________

4. Carol Boushey/ 8. ______________________

Format Approved by:

______________________ or ______________________

Chair, Final Examining Committee **Department Thesis Format Advisor**

This thesis ☐ **is** ☒ **is not to be regarded as confidential.**

____
Major Professor

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

____
Major Professor

Approved by: Michael R Melloch 12/6/11

Head of the Graduate Program **Date**

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:
Multilevel Image Segmentation with Application in Dietary Assessment and Evaluation

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Fengqing Zhu
Printed Name and Signature of Candidate

11/18/2011
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

MULTILEVEL IMAGE SEGMENTATION WITH APPLICATION IN DIETARY ASSESSMENT AND EVALUATION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Fengqing Zhu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2011

Purdue University

West Lafayette, Indiana

To my Savior, through whom I am renewed.

To my families, whose love continues to nourish me.

ACKNOWLEDGMENTS

Upon the completion of this dissertation, I realized that there will never be the perfect words to express my gratitude towards those who have inspired, encouraged and motivated me through my years at Purdue. Yet, these names and faces will always remain deep in my heart.

First of all, I would like to thank my major advisor, Professor Edward J. Delp. I thank him for taking me under his guidance. He has provided me with the opportunity to expand my learning horizon by becoming a member of the VIPER lab. I am grateful for his constant encouragement of independent thinking and learning through struggle. Moreover, I have enjoyed many non-academic related conversations with him, from which I have grew further respect for his broad and diverse knowledge of various topics.

I would like to thank my other committee members, Professor Jan. P. Allebach, Professor Carol J. Boushey, and Professor David S. Ebert for their advice, guidance and criticism. I have enjoyed my learning experience with them both inside and outside of classroom discussions on various topics. I am grateful for their patience and insightful suggestions. I especially like to thank Professor Boushey for her advice on nutrition related aspects of the TADA project.

I would like to thank the National Institutes of Health for their sponsorship of the research in this dissertation. Their generous support has made this dissertation possible.

I would like to especially thank Dr. Martin Okos, Dr. Heather Eicher-Miller, Dr. Nitin Khanna, Ziad Ahmad, Marc Bosch, Junghoon Chae, Shivangi Kelkar, SungYe Kim, Anand Mariappan, Karl Otsmo, TusaRebecca Schap, Bethany Six, Scott Stella, Insoo Woo, and Chang (Joy) Xu for their collaboration on the TADA project at Purdue University. I have also had the pleasure to visit and work with our project partner at the Curtin University in Australia, in particular, Dr. Deborah Kerr and Katherine Kerr.

During my time at the VIPER lab, I have been lucky to be surrounded by my colleagues who are bright individuals capable of critical thinking. I appreciate the support and friendship of my fellow colleagues in the VIPER lab: Dr. Golnaz Abdollahian, Ziad Ahmad, Marc Bosch, Andrew W. Haddad, Ye He, Dr. Nitin Khanna, Dr. Hyung Cook Kim, Dean King-Smith, Dr. Zhen Li, Dr. Liang Liang, Dr. Limin Liu, Kevin Lorenz, Dr. Ying Chen Lou, Anand Mariappan, Ashok Mariappan, Dr. Anthony Martone, Aravind Mikkilineni, Ka Ki Ng, Albert Parra, Dr. Satyam Srivastava, Dr. Hwayoung Um, Carlos Wang, Chang (Joy) Xu, Meilin Yang, and Bin Zhao.

I would like to thank my parents for their enduring support and encouragement. They have made many sacrifices throughout the years I have been away from home so that I can pursue my academic career. I thank them for giving me life and the opportunity to view the world with different perspectives. Finally, I would like to express my sincere appreciation to my husband, Minghao Qi, for his love, patience, understanding and encouragement.

This work was sponsored by grants from the National Institutes of Health under grants NIDDK 1R01DK073711-01A1 and NCI 1U01CA130784-01.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

DLW	Doubly Labeled Water
DSC	Digital Still Camera
E-TADA	Experiment TADA
FFQ	Food Frequency Questionnaire
FNDDS	Food and Nutrient Database for Dietary Studies
FR	Food Record
GTK	GIMP Toolkit
GUI	Graphical User Interface
HMM	Hidden Markov Models
IDW	Inverse distance weighting
I-TADA	Image TADA
LUT	Look-up Tables
mpFR	mobile telephone food record
SDK	Software Development Kit
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
SDK	Software Development Kit
T-FNDDS	TADA FNDDS
TADA	Technology Assisted Dietary Assessment
VCM	Visual Characterization Matrix

ABSTRACT

Zhu, Fengqing Ph.D., Purdue University, December 2011. Multilevel Image Segmentation with Application in Dietary Assessment and Evaluation. Major Professor: Edward J. Delp.

This thesis describes methods for image analysis, including image calibration, image segmentation, features extraction and classification with emphasis on the segmentation of non-rigid objects. We developed a color fiducial marker to correct colors of unknown image illumination that appear in the scene so that this information can be used by image analysis tasks in our dietary assessment system. We proposed and implemented a multiple hypothesis segmentation technique to select optimal segmentations based on confidence scores assigned to each segment and showed improvements in both segmentation and classification accuracy. We demonstrated the use of active contour models to refine image segmentation. We examined both quantitative performance and classification based evaluation to validate proposed segmentation methods.

The proposed image analysis methods were developed for a dietary assessment application. There is a growing concern with respect to chronic diseases and other health problems related to diet including obesity and cancer. The need to accurately measure diet (what foods a person consumes) becomes imperative. Dietary intake provides valuable insights for mounting intervention programs for prevention of chronic diseases. Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields. We describe a novel mobile telephone food record that can provide an accurate account of daily food and nutrient intake by analyzing images of the food eaten by a user. Our approach includes the use of image analysis tools for identification and quantification of food that is consumed at a meal.

1. INTRODUCTION

Nutritional epidemiology is concerned with quantifying dietary exposures and the association of these exposures with risks for disease. Diet represents one of the most universal biological exposures; however accurate assessment of food and beverage intake is problematic. This research focuses on developing part of a food record method using a mobile device that will provide an accurate account of daily food and nutrient intake. The method employs the use of image analysis tools for identification and quantification of food consumption. This work is sponsored by grants from the National Institutes of Health as part of the Genes, Environment and Health Initiative. The availability of "smart" mobile telephones with higher resolution imaging capability, improved memory capacity, network connectivity, and faster processors allow these devices to be used in health care applications. A dietary assessment application for a mobile telephone provides a unique mechanism for collecting dietary information that reduces burden on record keepers and will be of value to practicing dietitians and researchers. Along with my colleagues, we have developed a mobile telephone food record which uses a mobile device with a built-in camera, integrated image analysis and visualization tools with a nutrient database to allow a user to discretely record foods eaten. This project is the result of a collaboration between various departments at Purdue University, the University of Hawaii, and the Curtin University of Technology in Australia.

There is a growing concern with respect to chronic diseases and other health problems related to diet including obesity and cancer. Dietary intake, the process of determining what someone eats during the course of a day, provides valuable insights for mounting intervention programs for prevention of many chronic diseases. Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields. Our research addresses the challenges facing self-reporting methods that are prone to measurement error and other biases. These problems are further compounded by high respondent burden and high researcher burden. A food record system using a mobile device, a backend server, and database has been developed. The approach includes the use of image analysis for identification and quantification of food consumption based on images taken of the food items. Data from images obtained before and after food is consumed can be used to link the identified foods and amounts eaten with a food composition database. To aid with interaction design, the application has been tested by adolescents and adults from various age groups in both controlled meal sessions as well as free-living scenarios.

Particular interest of this thesis lies in the development of methods to automatically estimate the food consumed at a meal from images acquired using mobile device. Each food item is segmented, identified, and its volume is estimated. “Before” meal and “after” meal images can be used to estimate the food intake. From this information, the energy and nutrients consumed can be determined. A prototype system has been deployed on the Apple iPhone and its functionality has been verified with various combinations of foods [1].

The main thrust of this thesis is to develop methods for segmenting the food items from image acquired by the mobile device.

1.1 Technology Assisted Dietary Assessment

The increasing prevalence of obesity among the youth is of great concern [2] and has been linked to an increase in type 2 diabetes mellitus [3]. Accurate methods and tools to assess food and nutrient intake are essential in monitoring nutritional status. The collection of food intake and dietary information provides some of the most valuable insights into the occurrence of disease and subsequent approaches for mounting intervention programs for prevention. The assessment of food intake has been evaluated in the past by a food record (FR), the 24-hour dietary recall (24HR), and a food frequency questionnaire (FFQ) with external validation by doubly-labeled water (DLW) and urinary nitrogen [4–8]. Currently, there are few validation studies to justify one particular method over another for any given study design.

1.1.1 Review of Current Dietary Assessment Methods

A review of some of the most popular dietary assessment methods is provided in this section. The objective here is to describe the advantages and major drawbacks of these methods. This will demonstrate the significance of our mobile telephone food record which can be used for population and clinical based studies to improve the understanding of diet.

The 24-hour dietary recall (24HR) consists of a listing of foods and beverages consumed the previous day or the 24 hours prior to the recall interview. Foods and amounts are recalled from memory with the aid of an interviewer who has been trained in methods for soliciting dietary information. A brief activity history may be incorporated into the interview to facilitate probing (i.e. asking questions) for foods and beverages consumed. The Food Surveys Research Group (FSRG) of the United States Department of Agriculture (USDA) has devoted considerable effort to improving the accuracy of this method. The multiple-pass method provides a structured interview format with specific probes.

The major drawback of the 24HR is the issue of underreporting of the food consumed [9]. Factors such as obesity, gender, social desirability, restrained eating and hunger, education, literacy, perceived health status, age, and race/ethnicity have been shown to be related to underreporting [10–13]. Harnack, et al. [14] found significant underreporting of large food portions when food models showing recommended serving sizes were used as visual aids for respondents. Given that larger food portions have been observed as occurring over the past 20 to 30 years [15, 16], this may be a contributor to underreporting and methods to capture accurate portion sizes are needed. Youth, in particular, are limited in their abilities to estimate portion sizes accurately [4]. The most common method of evaluating the accuracy of the 24HR with children is through observation of school lunch and/or school breakfast [17] and comparing foods recalled with foods either observed as eaten or foods actually weighed. These recalls have demonstrated both under-reporting and over-reporting, and incorrect identification of foods.

The food record is especially vulnerable to underreporting due to the complexity of recording food [18, 19]. As adolescents snack frequently, have unstructured eating patterns, and consume greater amounts of food away from the home, their burden of recording is much greater compared to adults. It has been suggested that these factors, along with a combination of forgetfulness, irritation, and boredom caused by having to record intake frequently may be contributing to the underreporting in this age group [20]. Dietary assessment methods perceived as less burdensome and time-consuming may improve compliance [20].

Portion size estimation may be one contributor to underreporting. In [21] it was found that 45 minutes of training in portion-size estimation among 9-10 year olds significantly improved estimates for solid foods which were measured by dimensions or cups, and liquids estimated by cups. Amorphous foods were estimated least accurately even after training and some foods still exhibited an error rate of over 100%. Thus, training can improve portion size estimation, however, more than one session may be needed and accuracy may be unattainable.

There is a tremendous need for new methods for collecting dietary information.

1.1.2 The Use of Mobile Devices

The accurate assessment of diet is problematic, especially in adolescents [4, 22]. Mobile telephones can provide a unique mechanism for collecting dietary information that reduces burden on record keepers. A dietary assessment application that usses a mobile telephone for collecting information would be of value to practicing dietitians and researchers [23]. Previous results among adolescents showed that dietary assessment methods using a technology-based approach, e.g., a personal digital assistant with or without a camera or a disposable camera, were preferred over the traditional paper food record [24]. This suggests that for adolescents, dietary methods that incorporate new mobile technology may improve cooperation and accuracy.

We describe in [25, 26] a mobile telephone food record (mpFR) that the team at Purdue University developed using a mobile device (e.g. a mobile telephone or PDA-like device) to provide an accurate account of daily food and nutrient intake. Our goal is to use the mobile device with a built-in camera to allow a user to discretely record foods eaten. Each food item in the image is segmented, identified, and its volume is estimated [1, 27, 28]. Images acquired before and after foods are eaten can be used to estimate the food intake.

This system is known as the Technology Assisted Dietary Assessment System or the TADA System. The TADA system consists of two main parts: a mobile application we refer to as the Mobile Phone Food Record (mpFR) and the “backend” system consisting of the compute server and database system. Figure 1.1 shows the overall architecture of our proposed system. The first step is to send the image acquired with the mobile telephone and metadata to the server for automatic analysis, including image segmentation, food identification and volume estimation (steps 2 and 3). These results are sent back to the user where the user confirms and/or adjusts this information (step 4). In step 5, the server receives the confirmed information from the user. Based on the user feedback, refinements are applied to image segmentation and food labeling. Nutrient information is extracted using the USDA Food and Nutrient Database for Dietary Studies (FNDDS) database [29]. FNDDS is a database containing foods eaten in the U.S., their nutrient values, and weights for different standardized food portions (step 6). Finally these results can be sent to the research community for further analysis (step 7). We have deployed this system on an Apple iPhone and it is currently being used by dietitians and nutritionists in the Department of Foods and Nutrition at Purdue University for various adolescent and adult nutritional studies.

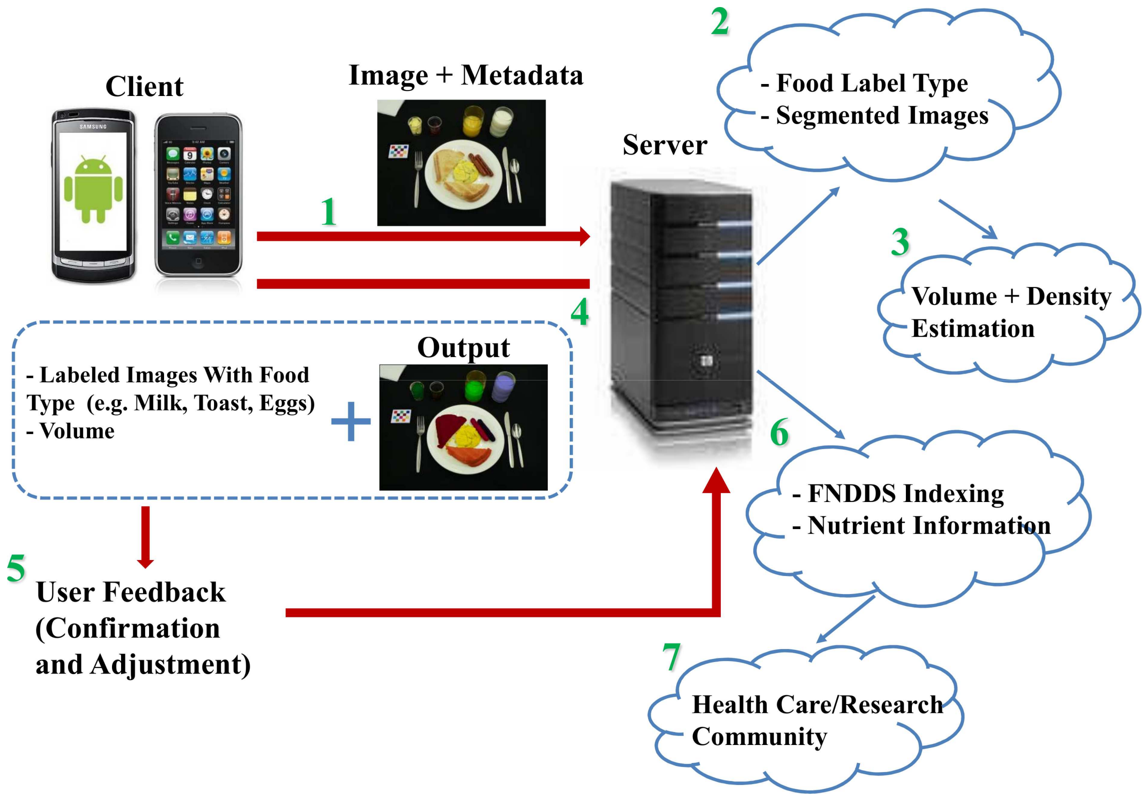

Client
Image + Metadata
1
Server
2
- Food Label Type
- Segmented Images
3
Volume + Density Estimation
4
Output
- Labeled Images With Food Type (e.g. Milk, Toast, Eggs)
- Volume
5
User Feedback (Confirmation and Adjustment)
6
- FNDDS Indexing
- Nutrient Information
7
Health Care/Research Community

Fig. 1.1. The Architecture of the TADA System.

1.1.3 Segmentation

An important aspect of our system includes locating and identifying food objects from meal images. This is the segmentation problem which is the main contribution of this thesis.

There has been previous segmentation techniques reported for food items and products. However, food image segmentation is still an unsolved problem because of its complex and under constrained attributes. Jimenez et.al [30] described an automatic fruit recognition system, which recognized spherical fruit in various situations such as shadows, bright areas, occlusions and overlapping fruit. A three-dimensional scanner was used to scan the scene and generate five images to represent the azimuth and elevation angles, range, attenuation and reflectance. The position of the fruit obtained by thresholding and clustering with the Circular Hough Transform was used to identify the center and radius of the fruits. A robust method to segment the food items from the background of color images was proposed in [31]. A color image was converted to a high contrast grayscale image from an optimal linear combination of the RGB color components. The image is then segmented using a global threshold estimated by a statistical approach to minimize the intraclass variance. The segmented regions were subjected to a morphological process to remove small objects, to close the binary image by dilation followed by erosion and to fill the holes in the segmented regions. The work presented in [32] used a stick growing and merging method to segment complex food images. The image was first pre-processed by an edge-preserving smoothing technique and a large number of horizontal lines ("sticks") were built containing homogeneous pixels with the sticks ends corresponding to edge points. Once the sticks were built, adjacent sticks were merged to form a sub region based on a stick-stick homogeneity criterion. The sub regions are then merged if it satisfies a minimal sub region merging criterion. The regions with non-stick areas appear as noises or edges which cause less smooth boundary. Boundary modification step is used to reduce the degree of boundary roughness. This method is also successful in segmenting many complex food images including pizza, apple, pork and potato.

Automatic identification of foods from an image is an example of object classification. It is a difficult problem since foods can dramatically vary in appearance. Such variations may arise not only from changes in illumination and viewpoint but also from non-rigid deformations, and intraclass variability in shape, texture, color and other visual properties. There has been recent efforts to address challenges in food identification. In [33] a method for food identification was described by exploiting the spatial relationship among different ingredients (such as meat and bread in a sandwich). The food items were represented by pairwise statistics between local features of the different ingredients of the food items. In [34], a multiple kernel learning method was described to integrate three sets of features namely color, texture, and SIFT descriptors. All three features were fused together forming one single feature vector by assigning different weights to combine them. In [35], an efficient fusion of color and texture features for fruit recognition was proposed. The recognition was done by minimum distance classifier based upon the statistical and co-occurrence features derived from the Wavelet transformed sub-bands. Finally in [36], an online food-logging system was presented, which distinguished food images from other images, analyzed the food balance, and visualized the log. Global and local features were used to describe food items and classify them using a Support Vector Machine (SVM).

In our image analysis system, once a food image is acquired, we need to locate the object boundaries for the food items within the image. This is accomplished by image segmentation. The ideal segmentation is to group pixels in the image that share certain visual characteristics perceptually meaningful to human observers. Although segmentation is a difficult task, it is very important because good segmentation can help with recognition, registration, and image database retrieval. In our system, the results of the segmentation are used for food labeling and automatic portion estimation. Thus, the accuracy of segmentation plays a crucial role in the overall performance of our system.

In this thesis we have investigated various approaches to segment food items in an image such as connected component labeling, active contours, normalized cuts and semi-automatic methods [1,25,37,38]. We proposed multiple hypothesis segmentation to select optimal segmentations based on confidence scores assigned to each segment [39]. This approach combined two ideas: a set of segmented objects could be partitioned into perceptually similar object classes based on global and local features; and perceptually similar object classes could be used to assess the accuracy of image segmentation. Both segmentation and classification accuracy were improved by generating multiple segmentations of each image using multiscale graph decomposition. In this approach, we first detected regions where potential food items were located instead of segmenting the entire image. Once these regions of interest were detected, suitable segmentation techniques could be used for each of these regions to find the precise boundaries of the food items. Each of these segments were classified into a particular food label using the features extracted from that segment.

In the TADA system, segmentation is followed by feature extraction and classification. Earlier approaches in the TADA system included the extraction of global color and texture features for each segment, followed by classification of the segment using support vector machines [1, 25, 26, 37, 38]. This work was done by me and is included in this thesis. Since then, we have investigated more features to efficiently characterize food items visually, including texture and local descriptors [40]. This is mainly the work by my colleague Marc Bosch [41]. We proposed a food classification framework [42] where multiple features spaces were independently classified and fused according to a set of rules to achieve a final labeling decision. Potential misclassifications were corrected by using different sources of contextual information, namely food/object combination likelihood, and information from the confusion matrix on the validation dataset.

1.2 Contributions of This Thesis

The research in this thesis focuses on developing methods for image segmentation and particularly for the segmentation of food images. My main contributions are as follows:

- Developed a multiple hypothesis segmentation system to select optimal segmentations based on confidence scores assigned to each segment. This approach combined two ideas: a set of segmented objects could be partitioned into perceptually similar object classes based on global and local features; and perceptually similar object classes could be used to assess the accuracy of image segmentation. Both segmentation and classification accuracy were improved by generating multiple segmentations of each image using multiscale graph decomposition.

- Evaluated the quantitative performance of our multilevel segmentation approach based on comparing region boundaries with ground-truth data for consistency. A separate evaluation was performed in the context of object classification to assess pixel level accuracy for identifying each food class.

- Demonstrated the use of region-based active contour models to refine the image segmentation based on user feedback from the TADA mpFR. With such feedback, an initial curve, was deformed to the boundary of the object under constraints from the image.

- Developed a color fiducial marker used as the reference object in images captured under various illumination conditions. Constructed a transformation based on the reference illumination to correct the colors of the unknown image. The realization of the method was achieved through the use of a 3D LUT.

- Examined color and texture features for each segmented food region. The was part of the initial classification methods used in the TADA system. These features were classified using SVM.

1.3 Publications Resulting from This Work

Journal Papers:

1. **Fengqing Zhu**, Marc Bosch, Insoo Woo, SungYe Kim, Carol J. Boushey, David S. Ebert, and Edward 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, August 2010, pp. 756-766.

2. Bethany L. Six, TusaRebecca E. Schap, **Fengqing Zhu**, Anand Mariappan, Marc Bosch, Edward J. Delp, David S. Ebert, Deborah A. Kerr, and Carol J. Boushey, "Evidence-Based Development of a Mobile Telephone Food Record," *Journal of American Dietetic Association*, January 2010, pp. 74-79.

3. TusaRebecca E Schap, **Fengqing Zhu**, Edward J. Delp, and Carol J. Boushey, "Merging Dietary Assessment with the Adolescent Lifestyle," *Journal of Human Nutrition and Dietetics*, submitted.

Conference Papers:

1. **Fengqing Zhu**, Marc Bosch, Ziad Ahmad, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Challenges in Using a Mobile Device Food Record Among Adults in Free-living Situations," *mHealth Summit*, Washington DC, December 2011.

2. **Fengqing Zhu**, Marc Bosch, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Multilevel Segmentation for Food Classification in Dietary Assessment," *Proceedings of* 7th *International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, September 4-6, 2011, pp. 337-342.

3. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Combining Global and Local Features for Food Identification and Dietary Assessment," *Proceedings of the International Conference on Image Processing*, Brussels, Belgium, September 2011.

4. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Food Texture Descriptors Based on Fractal and Local Gradient Information," *Proceedings of the* 19th *European Signal Processing Conference*, Barcelona, Spain, September 2011.

5. Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, **Fengqing Zhu**, Carol J. Boushey and Edward J. Delp, "Integrated Databases System for Mobile Dietary Assessment and Analysis," *Proceedings of the* 1st *IEEE International Workshop on Multimedia Services and Technologies for E-health in conjunction with the International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011.

6. **Fengqing Zhu**, Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, David S. Ebert, Carol J. Boushey and Edward J. Delp, "Segmentation Assisted Food

Classification for Dietary Assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, Vol. 7873, Burlingame, California, January 2011.

7. JungHoon Chae, Insoo Woo, SungYe Kim, Ross Maciejewski, **Fengqing Zhu**, Edward J. Delp, Carol J. Boushey, and David 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, Burlingame, California, January 2011.

8. **Fengqing Zhu**, Marc Bosch, Carol J. Boushey and Edward J. Delp, "An Image Analysis System for Dietary Assessment and Evaluation," *Proceedings of the International Conference on Image Processing*, Hong Kong, China, September, 2010.

9. Anand Mariappan, Marc Bosch, **Fengqing Zhu**, Carol J. Boushey, David S. Ebert, Deborah A. Kerr, and Edward J. Delp, "Personal Dietary Assessment Using Mobile Devices," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VII*, Vol. 7246, San Jose, California, January 2009.

10. **Fengqing Zhu**, Anand Mariappan, Carol J. Boushey, Deborah A. Kerr, Kyle Lutes, David S. Ebert, and Edward J. Delp, "Technology-Assisted Dietary Assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VI*, Vol. 6814, San Jose, California, January 2008.

2. IMAGE ANALYSIS

Our goal is to identify food items in a scene. The ideal image analysis system is shown in Figure 2.1, where each food item is segmented and identified. The emphasis of this thesis will be on addressing the problem of image segmentation.

A block diagram of our proposed image analysis system is shown in Figure 2.2. Our overall goal is to automatically determine the regions in an image where a particular food is located (segmentation) and correctly identify the food type based on its features (classification or food labeling). Since we are interested in measuring the amount of food in the image, we have developed a very simple protocol for users of our system [23, 24]. This protocol involves the use of a calibrated fiducial marker consisting of a checkerboard (color checkerboard) that is placed in the field of view of the camera. This allows both geometric and color correction of the images so that the amount of food present can be estimated.

Automatic identification of food items in an image is not an easy problem. We fully understand that we will not be able to recognize every food. Some food items look very similar, e.g. margarine and butter. In other cases, the packaging or the way the food is served will present problems for automatic recognition. For example, if the food is in an opaque container then we will not be able to identify it. In some cases, if a food is not correctly identified, it may not make much difference with respect to the energy or nutrients consumed. An example of this is if our system identifies a “brownie” as “chocolate cake”, there is not a significant difference in the energy or nutrient content. Similarly, if we incorrectly estimate the amount of lettuce consumed, this will also have little impact on the estimate of the energy or nutrients consumed in the meal due to the low energy content of lettuce [23, 24]. Again, we emphasize that our goal is to provide a tool for better assessment of dietary intake to

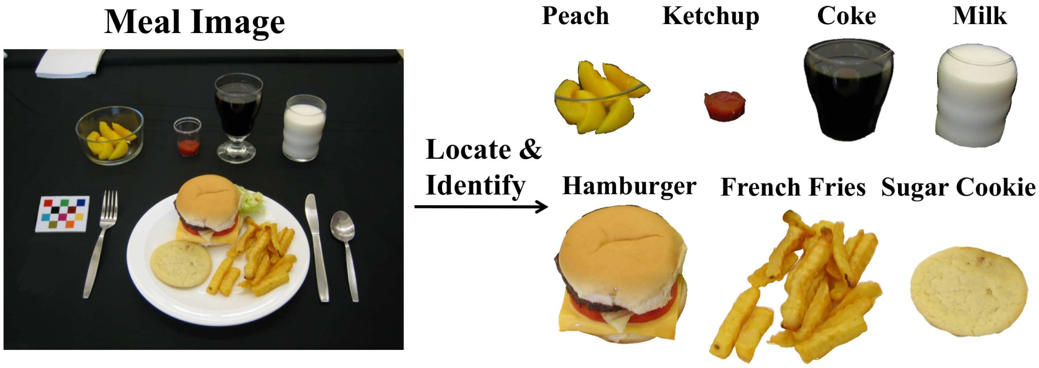


Fig. 2.1. An Ideal Food Image Analysis System.

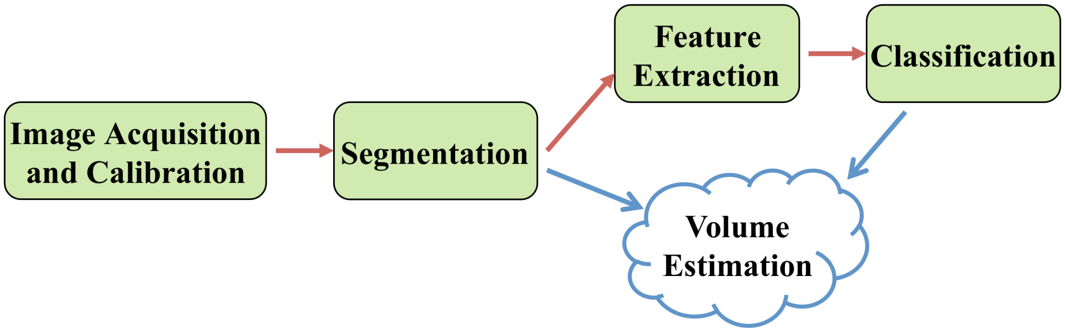


Fig. 2.2. Proposed Approach for Image Analysis System.

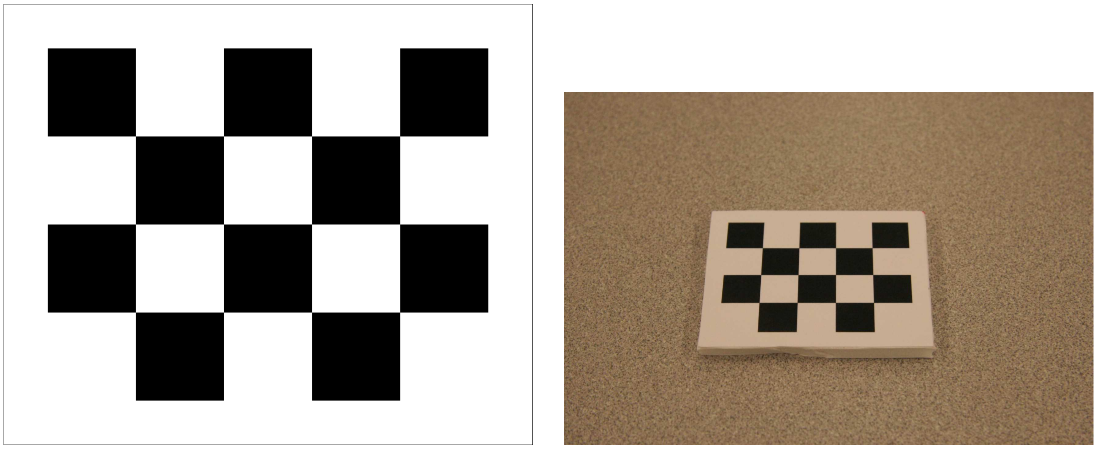
professional dietitians and researchers than that is currently available using existing methods.

2.1 Image Acquisition and Calibration

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 in November 2011 a user takes only one image of the food and 3D models are used to construct the 3D object. Other approaches include acquire multiple images of the food scene from different “views” of the food. The methods used for constructing the 3D information is beyond the scope of this thesis. Whatever approach is used, we still a calibrated imaging system that is calibrated booth spatially and with respect to the color represented in the scene. This could be accomplished by having the user acquire the image with a known object, e.g., a pen or PDA stylus, placed next to the food so one could use this to “calibrate sizes in the image. We might also use the known dimensions of a plate or cup in a scene. Other *a priori* information in the scene such as the the pattern on the tablecloth could also be used. We have chosen to use 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. Several controlled feeding studies were conducted by the Department of Foods and Nutrition at Purdue University whereby participants were asked to take images of their food before and after meals [23]. Responses from large proportion of the participants in these studies indicated that it would be easy to use a credit card-sized fiducial marker due to the convenient incorporation into their current lifestyles.

2.1.1 The Fiducial Marker

The initial development of the fiducial marker was done in collaboration with Karl Ostmo as part of his Master thesis [43]. The OpenCV library [44] was used to detect the presence of fiducial marker in the image since it contains built-in support for the detection of a checkerboard (chessboard) calibration pattern. The OpenCV library uses the Hough transform to extract feature points from the checkerboard. The Hough transform locates lines, then derives corner points by thresholding intersections of the resulting lines. Through experimentation, the optimal properties of the checkerboard have been determined as follows: the pattern is an asymmetric checkerboard with [odd] × [even] dimensions, minimum 4 tiles per side, high contrast, matte finish, and rigid mounting. Lighting conditions and camera quality can affect the appearance of color in the acquired images. Since the recognition of food is based on color, it is



(a) (b)

Fig. 2.3. Examples of a Black And White Fiducial Marker, (a) shows the source used to create the marker, (b) shows an image of the fiducial marker.

important to incorporate color information into the design. We need to detect and located the fiducial marker in the scene so that we can use the calibration information. This is done using only the gray scale version of the image acquired from the camera. Since we use the color fiducial marker to also calibrate the color information in the scene, the gray scale image of the color marker has somewhat less contrast than a true black and white marker and can affect the ability of OpenCV to recognize the corners. Therefore we must take this into account when designing the color marker. Examples of the black and white fiducial marker and a version of the color fiducial marker are shown in Figure 2.3 and Figure 2.4.

2.1.2 Color Correction

Color correction, also referred to as *color balance* is the global adjustment of the color intensities in an image when used with still images. The goal of such adjustment is to render neutral colors in an image correctly. Methods for such correction are generally known as gray balance, neutral balance or white balance [45]. Color correction

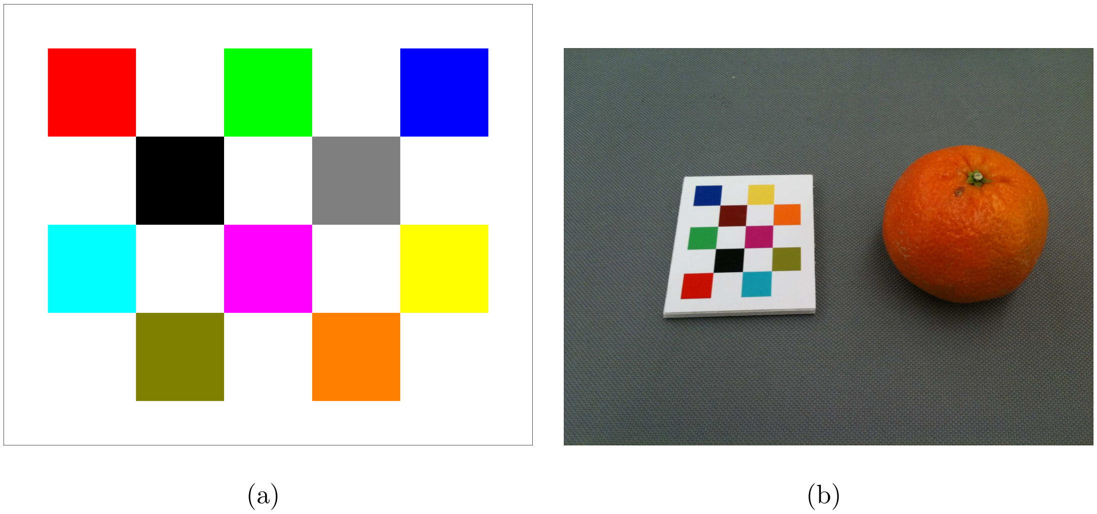


Fig. 2.4. Examples of the Color Fiducial Marker, (a) shows the source used to create the marker, (b) shows an image of the fiducial marker.

changes the overall colors in an image and is often used for colors other than neutrals to appear correct or pleasing. Image data acquired by sensors such as a CCD sensor must be transformed from the acquired image values to values that are appropriate for color display and reproduction. Such color correction is essential for several aspects of the acquisition and display process. These include the fact that acquisition sensors do not match the sensors in the human vision system, that the properties of the display medium must be considered, and that the ambient viewing conditions of the acquisition may have large variations. In addition, images captured by mobile telephone cameras have lower quality than most digital still cameras (DSCs). The observed colors in a mobile phone camera image are often “substantially incorrect.” This poses a challenge for achieving satisfying results from the image analysis.

A color calibration method for correcting the variation in RGB color components caused by the vision system was described in [46]. The calibration scheme concentrated on comprehensively estimating and removing the RGB errors under both uniform and nonuniform illuminations. Siddiqui, et al [47] proposed a hierarchical color correction method for enhancing the color of digital images captured by mobile telephone cameras obtained from low-quality images. The method is based on a multilayer hierarchical stochastic framework in which parameters are learned in an offline training process using the expectation maximization (EM) method. A color correction system using a color compensation chart was proposed by Lee, et al [48] where the system introduced the color compensation chart to estimate the transformation between the colors in the image and the reference colors. The concept of color management system (CMS) and the profile connection space (PCS) were adopted to realize the proposed system.

In [49], a color correction method is presented for automatic detection of identical objects from different images. The 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,

$$C_{A-B} = \frac{1}{24}\sum_{i=1}^{24} C_i = (C_r, C_g, C_b) \tag{2.1}$$

The illumination conversion vector is then used according to 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}) in the image of illumination B is corrected by adding the illumination conversion vector C_{A-B} above as follows

$$\left(R^{'}_{xy}, G^{'}_{xy}, B^{'}_{xy}\right) = (R_{xy}, G_{xy}, B_{xy}) + (C_r, C_g, C_b) \tag{2.2}$$

Similar to the approach described in [49], the color fiducial marker is used as the identical objects in images captured under various conditions, in particular, different illuminations. We adopted a simplified version of the approach proposed by Srivasrava, et al [50] to address the problem of visually matching two known display devices in color management systems. Our method starts with capturing an image of the color fiducial marker using a digital still camera or a mobile device camera under a known illumination and use color patches in the captured fiducial marker image as the reference marker colors. To correct the colors in an image containing the color fiducial maker from unknown sources or illuminations, RGB components of the color patches in the unknown image are extracted and a mapping between the reference marker colors and the newly obtained marker colors are established. Finally, this mapping is used to correct the colors of the unknown image to match the reference image.

The method is implemented through the use of 3D look-up tables (LUT). Look-up tables (LUT) are used in image processing to transform input data into a desired output format. Consider a function $y = f(x)$ which maps an input $x \in D$ to an output $y \in R$ where D and R are the domain and the range of f, respectively. The look-up table allows computation of discrete values of f for a set of discrete inputs $S_D \in D$ and stores them as a set $S_R \in R$. In order to evaluate this function at a given input x, we located points in S_D which surround x. In other words, elements of S_D forms a box such that x is an internal point of the box. These elements are known as the neighbors of x. Finally, $f(x)$ is obtained by interpolating between the known values of the function at its neighbors. To achieve the matching between an image from unknown sources or illuminations to the reference image, we need a function that would take an RGB input and produce an RGB output. Let $f(\cdot)$ represent the mapping between the marker colors in the unknown image and the reference maker colors. The transformation f requires a 3D LUT with a vector $[r, g, b]$ as input. Each entry in the LUT is also a vector $[r, g, b]$ corresponding to the color corrected image. Although the input space is of size $256 \times 256 \times 256$, practical LUTs are much smaller. In our case, we consider a uniformly sampled LUT of size $3 \times 3 \times 3$ and select sample points according to the set

$$\Omega = \{r_0, r_1, r_2\} \times \{g_0, g_1, g_2\} \times \{b_0, b_1, b_2\} \tag{2.3}$$

where × denotes a Cartesian product. Then, the LUT can be described by a function $\Phi(\cdot)$ such that

$$\Phi\left([r,g,b]\right) = \begin{cases} f\left([r,g,b]\right), & \forall\,[r,g,b] \in \Omega \\ \psi\left([r,g,b]\right), & \text{otherwise} \end{cases} \tag{2.4}$$

where ψ represents an interpolation operation to transform input values not aligned with any of the table entries.

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 $i = 0,1,...,k-1$. 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 the interpolation method

$$f\left([x,y,z]\right)_c = \psi\left(f_i, d_i\right) \quad \text{for } i = 0,1,...,k-1 \tag{2.5}$$

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

$$f_c = \left(f_0 \cdot d\left(c,1\right) + f_1 \cdot d\left(c,0\right)\right) / \left(d\left(c,0\right) + d\left(c,1\right)\right) \tag{2.6}$$

where $d(c,i)$ is the Euclidean distance between c and its ith neighbor. Extending to 3D, f_c is evaluated as a weighted sum of f_i for $i = 0,1,...,7$, with each f_i weighted by the Euclidean distance of the other neighbor from the input point c as shown in Figure 2.5. The euclidean distance is normalized by the total distance between all neighbors. A number of schemes exist for interpolation using different neighbor sets and distance metrics. The *Shepard interpolation* [51] is a general form of finding an interpolation value on an irregularly spread data space. It belongs to the method of *inverse distance weighting (IDW)*. Suppose we want to find an interpolated value f_c at a given point c based on samples f_i for $i = 0,1,...,N$, where N specifies the total number of neighbors, using IDW as an interpolating function

$$f_c = \begin{cases} f_i, & \text{if } w_i = 0 \text{ for any } i \in \{0,1,...,N-1\} \\ \sum\limits_{i=0}^{N-1} \frac{w_i}{\sum_{i=0}^{N-1} w_i} f_i, & \text{otherwise} \end{cases} \tag{2.7}$$

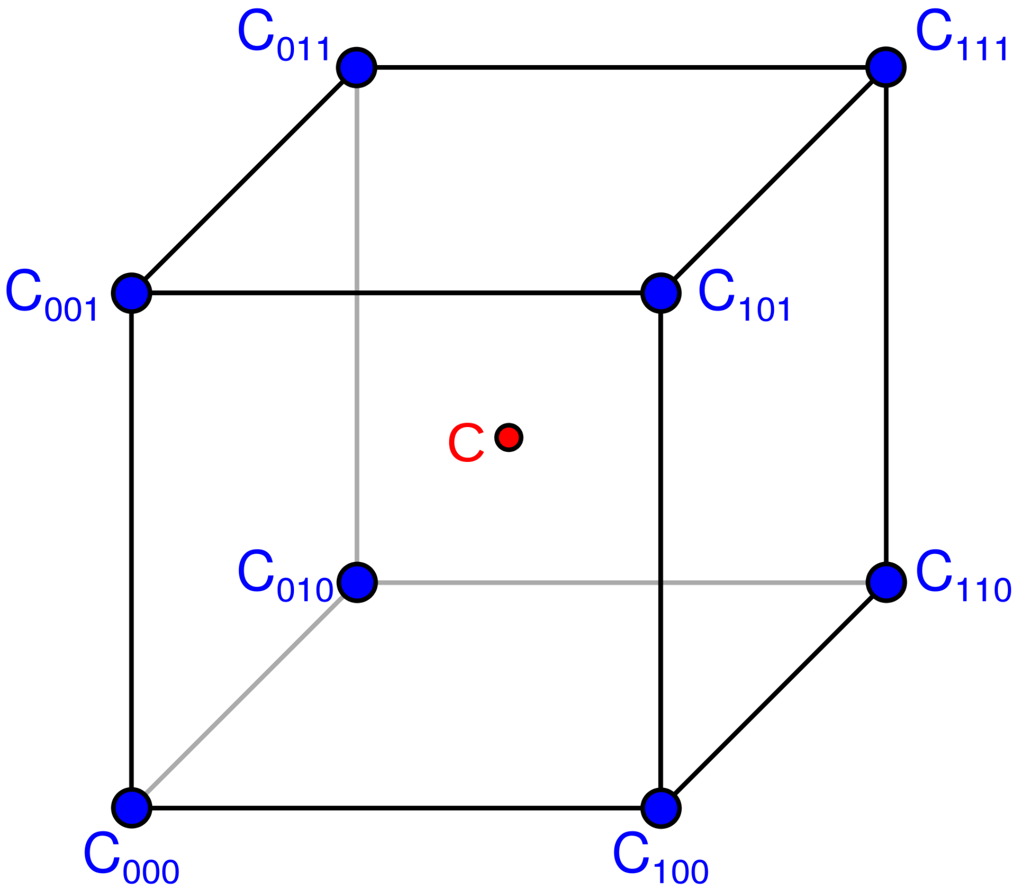


Fig. 2.5. Graphic illustration of the 3D interpolation.

where

$$w_i = \frac{1}{d\,(c, i)^p} \tag{2.8}$$

is a simple IDW weighting function defined by Shepard. Larger value of p assigns greater influence to values closest to the interpolated point. When selecting $p = 2$, the result is satisfactory for surface mapping and computationally inexpensive. Therefore, we use quadratic weighing in the 3D space, which results in the triquadratic method described above. We selected the triquadratic interpolation based on its accuracy and simplicity. However, other interpolation scheme such as splines or Gaussian processes may be selected depending on the requirements and optimization can be perform by modifying the cost function to reflect the change.

Sample color correction results using 3D LUT methods, as well as comparison to Choi's method [49] are illustrated in Section 4.1.1.

2.2 Image Segmentation

In our image analysis system, once a food image is acquired and color corrected, we need to locate the object boundaries of the food items within the image. Image segmentation is used to accomplish such goal. The ideal output of this operation is to group pixels in the image that share certain visual characteristics that are perceptually meaningful to human observers. This is a difficult problem as humans use all sorts of tricks to perform this task. However, this task is very important as good segmentation can help with recognition, tracking, image database retrieval and image compression. In our system, the output of the segmentation step is used for the food labeling and automatic portion estimation step. Thus, the accuracy of this step plays a crucial role in the overall performance of our system. We have investigated various approaches to segment food items in an image such as connected component labeling, active contours, normalized cuts, semi-automatic methods and multilevel approach [1, 25, 37–39]. Since we are interested in food objects in an image, it is more efficient to first detect regions where potential food items are located instead of segmenting the entire image. Once these regions of interest are detected, suitable segmentation techniques can be used for each of these regions to find the precise boundaries of the food items. Each of these segments are classified into a particular food label using features extracted from that segment. The TADA system also includes a review step where a user can provide feedback information such as confirmation and adjustment for the location and identification of foods in the meal image. The refinement step can then be used to generate the final segmentation based on the user feedback information. Our proposed segmentation approach is illustrated in Figure 2.6. A complete description of the approach and various segmentation methods examined, as well as evaluation of proposed techniques can be found in Chapter 3.

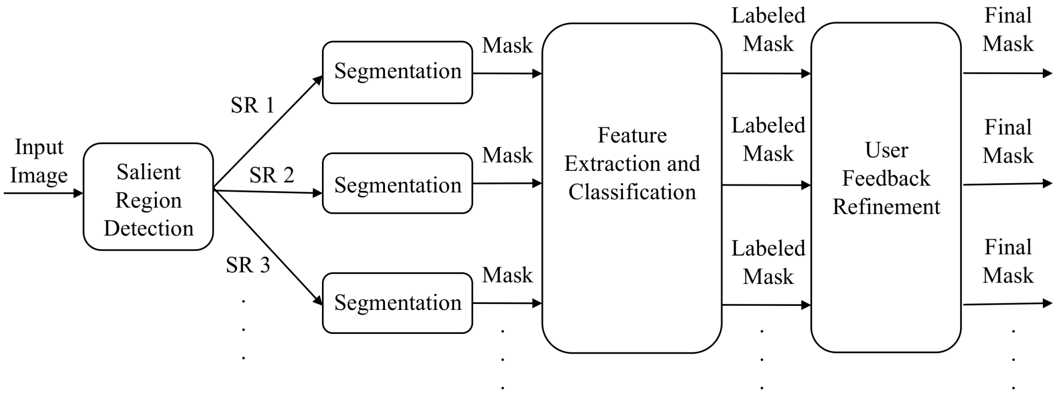


Fig. 2.6. Proposed Segmentation Approach.

2.3 Feature Extraction and Classification

An essential step in solving any object categorization/classification task is to adequately represent the visual information of the object. This is commonly known as feature extraction. Features need to be robust to image perturbations such as viewpoint and illumination changes. A solution for object classification problems is to take a part-based approach, *i.e.*, given any image segment containing, partially, the object, the pixel value information is projected into a transformed space known as the feature space. In the feature space a statistical description is estimated forming a feature vector. The features are used by a classifier, which maps observed features into a label or category, representing a food type from a set of potential foods.

The feature selection methods discussed in this section were used in our earlier deployment of the TADA system. More recent contributions in this area have been made by Marc Bosch [40–42].

2.3.1 Feature Extraction

Given the labeled pixels of an image after segmentation, we need to extract visual characteristics of each region (food item) to help with training the classifier to identify each food item. Therefore, proper selection of features is the key for correct classification. Certain criteria is necessary when selecting the features: the features should contain enough information to uniquely describe each region (food item) yet are not domain-specific; they should be easy to compute and relate well with the human visual system. Based on these criteria, two types of features are extracted/measured for each segmented food region, color and texture features. As noted above, as part of the protocol for obtaining food images the subjects are asked to take images with a calibrated fiducial marker consisting of a color checkerboard that is placed in the field of view of the camera. This allows us to correct for color imbalance in the mobile device's camera.

For color features, the average value of the pixel intensity (i.e. the gray scale) along with two color components are used. The color components are obtained by first converting the image to the CIELAB color space. The L^* component is known as the luminance and the a^* and b^* are the two chrominance components.

For texture features, we use Gabor filters to measure local texture properties in the frequency domain. Several Gabor techniques for texture-segmentation applications can be found in [52–55]. Gabor filters describe properties related to the local power spectrum of a signal and have been used for texture analysis [54]. A Gabor impulse response in the spatial domain consists of a sinusoidal plane wave of some orientation and frequency, modulated by a two-dimensional Gaussian envelope and is given by:

$$h(x,y) = exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right]\cos\left(2\pi Ux+\varphi\right) \tag{2.9}$$

In our work, we use the Gabor filter-bank proposed in [52]. It is suitable for our use where the texture features are obtained by subjecting each image (or in our case each block) to a Gabor filtering operation in a window around each pixel and then estimating the mean and the standard deviation of the energy of the filtered image. A Gabor filter-bank consists of Gabor filters with Gaussians of several sizes modulated by sinusoidal plane waves of different orientations from the same Gabor-root filter as defined in Equation (2.9), it can be represented as:

$$g_{m,n}(x,y) = a^{-m}h(\tilde{x},\tilde{y}), \quad a>1 \tag{2.10}$$

where $\tilde{x} = a^{-m}(x\cos\theta + y\sin\theta)$, $\tilde{y} = a^{-m}(-x\sin\theta + y\cos\theta)$, $\theta = n\pi/K$ (K = total orientation, $n = 0, 1, ..., K-1$, and $m = 0, 1, ..., S-1$), and $h(\cdot,\cdot)$ is defined in Equation (2.9). Given an image $I_E(r,c)$ of size $H \times W$, the discrete Gabor filtered output is given by a 2D convolution:

$$I_{g_{m,n}}(r,c) = \sum_{s,t} I_E(r-s, c-t) {g_{m,n}}^{*}(s,t), \tag{2.11}$$

As a result of this convolution, the energy of the filtered image is obtained and then the mean and standard deviation are estimated and used as features. In our implementation, we divide each segmented food item into $N \times N$ non-overlapped blocks and use Gabor filters on each block. We use the following Gabor parameters: 4 scales (S=4), and 6 orientations (K=6).

2.3.2 Classification

Once the food items are segmented and their features are extracted, the next step is to identify the food items using statistical pattern recognition techniques [56, 57]. Classifiers are used in a variety of pattern recognition and machine learning applications ranging from automatic speech recognition to the characterization of printers. We consider supervised learning, in which observed or measured data are labeled with pre-defined classes. The process often consists of two steps. In the training phase, a model is learned using training data. The model is tested using unseen data to assess the model accuracy in the testing phase. By training, the classifier "learns" how to map the features into the category or label. The training is not perfect and the classifier will make mistakes in actual operation by assigning the wrong label to an observed feature vector. Training the classifier has two goals. One goal is to determine how it will assign labels to observed feature vectors and the other is to estimate its error performance or its classification accuracy. This is accomplished by using a set of feature vectors where the label or category is known *a priori*, often referred to as "ground-truth" information. This ground-truth information is divided into two sets, a training set and a testing set. The training set or data is used to train the classifier and the testing set or data is used to test the error performance of the classifier. The feature vectors used for our system contain 51 values, 48 texture features and 3 color features. The feature vectors for the training images (which contain only one food item in the image) are extracted and a training model is generated.

For classification of food items, we use a support vector machine (SVM) [58–60]. A support vector machine (SVM) generates a hyperplane or separation boundary that separates the training data into two regions in the feature space $\Re^D$. For example, assume the training data X_i is linearly separable and are positioned in a feature space of size $\Re^2$, we can draw a line on a graph to separate training vectors with different labels into two regions R_1 and R_2 as illustrated in Figure 2.7. R_1 contains training vectors with the label $y_i = -1$ and the positions of these training vectors are denoted using $\circ$'s. R_2 contains training vectors with the label $y_i = +1$ and the positions of these training vectors are denoted using $\bullet$'s.

For theoretical illustration of the SVM, $D = 2$ classes are considered here. For training vectors with two class labels $y_i \in -1, 1$, the linear SVM defines a hyperplane as

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.12}$$

where $\mathbf{w}$ is normal to the hyperplane and $b/\|\mathbf{w}\|$ is the perpendicular distance from the hyperplane to the origin. The training vectors are assumed to be linearly separable meaning that the training vectors satisfy the boundary conditions of the following

$$\begin{aligned} \mathbf{x_i} \cdot \mathbf{w} + b \geq +1 \quad \text{for}\, y_i = +1 \\ \mathbf{x_i} \cdot \mathbf{w} + b \leq -1 \quad \text{for}\, y_i = -1 \end{aligned} \tag{2.13}$$

Support Vectors are training vectors lie closest to the separating hyperplane and the goal of SVM is to orientate this hyperplane in such a way that is as far as possible from the closest members of both classes. The training vectors on the boundaries, specified by the two planes H_1 and H_2 can be described as

$$\begin{aligned} \mathbf{x_i} \cdot \mathbf{w} + b = +1 \quad \text{for}\, H_1 \\ \mathbf{x_i} \cdot \mathbf{w} + b = -1 \quad \text{for}\, H_2 \end{aligned} \tag{2.14}$$

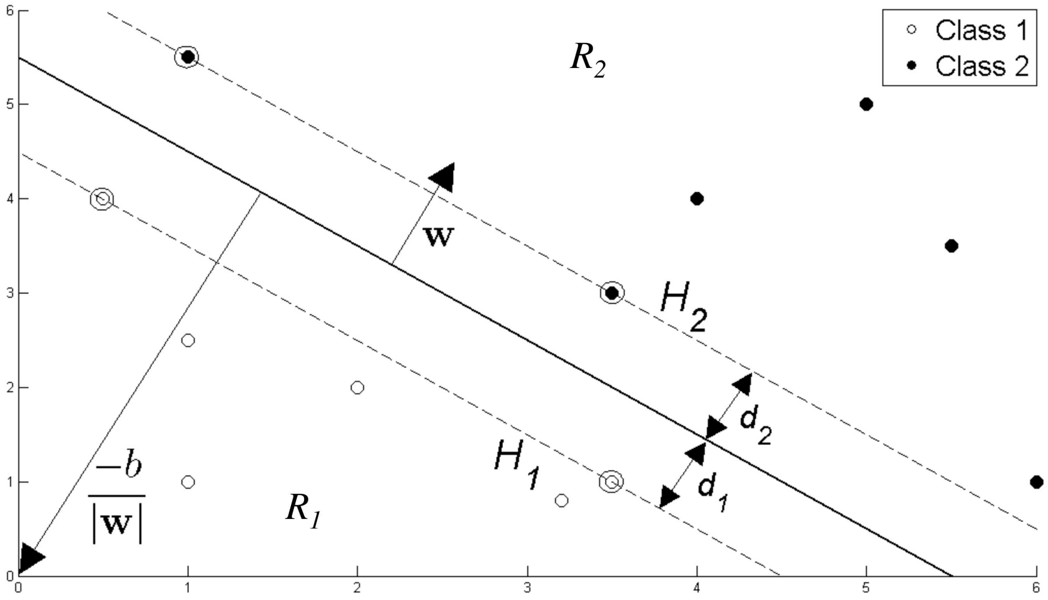


Fig. 2.7. Example of Feature Space $\Re^2$ Used by SVM Showing Hyperplane Through Two Linearly Separable Classes.

Distances from H_1 and H_2 to the hyperplane is defined as d_1 and d_2, respectively in Figure 2.7. The hyperplane's equidistance from H_1 to H_2 is the SVM's *margin*, a quantity to be maximized. A unique solution exists for maximizing this quantity using Lagrange's theorem [61]:

$$L_D = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.15}$$

under these constraints

$$\begin{aligned} &\alpha_i \geq 0 \, \forall_i \\ &\sum_{i=1}^{L} \alpha_i y_i = 0 \end{aligned} \tag{2.16}$$

where $\alpha_1, \cdots, \alpha_L$ are the Lagrange multipliers.

A nonlinear SVM is used in place of the linear SVM of the training vectors are not linearly separable. The training vectors are separated by a nonlinear boundary. The nonlinear decision boundaries are determined by mapping the training vectors to some other Euclidian space H, where the mapping function is denoted as $\Phi : \Re^D \Rightarrow H$. A hyperplane is then used in H to separate the mapped training vectors. To avoid increase complexity due to mapping the training vectors to a high dimension, the mapping of $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ is replaced by a *Kernel Function* $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. The Lagrangian of Equation 2.15 is then redefined for the nonlinear case by substituting $k(\mathbf{x}_i, \mathbf{x}_j)$ in the place of $(\mathbf{x}_i, \mathbf{x}_j)$. This produces the modified maximization problem formulated as

$$L_D = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i \cdot \mathbf{x}_j) \tag{2.17}$$

The SVM classifier generates its training model using the LIBSVM [62], a library for support vector machines that supports multi-class classification. In particular, the kernel function used in our implementation is the radial basis function (RBF)

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(\frac{-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right) \tag{2.18}$$

The labeled food type along with the segmented image are sent to the automatic portion estimation module where camera parameter estimation and model reconstruction are utilized to determine the volume of extracted foods.

The above color and texture features as well as the SVM classifier were tested on a set of food images collected from one of the diet studies conducted by the Department of Foods and Nutrition at Purdue University. Detailed description of the experiments and results are presented in Section 4.1.2.

3. SEGMENTATION OF NON-RIGID OBJECTS

Segmentation is the process of partitioning an image into disjoint and homogeneous regions, or equivalently, finding the boundaries between the regions. The desirable characteristics of a good image segmentation method have been described by Haralick and Shapiro in [63] with reference to grayscale images. "Regions of an image segmentation should be uniform and homogeneous with respect to some characteristic such as gray tone or texture. Region interiors should be simple and without many small holes. adjacent region of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segmentation should be simple, not ragged, and must be spatially accurate." Based on the above requirements, a formal definition of segmentation is given by [64, 65] in the following: Let I denote the set of all pixels and P a uniformity (homogeneity) predicate defined on groups of connected pixels, then the segmentation of I is a partitioning of the set I into a set of connected subsets (regions) $R_n, n = 1, ..., N$ such that

$$\bigcup_{n=1}^{N} R_n = I \quad \text{with} \quad R_n \bigcap R_m = \emptyset, \quad n \neq m \tag{3.1}$$

$$P\left(R_n\right) = \text{true} \quad \forall n \tag{3.2}$$

$$P\left(R_n \bigcup R_m\right) = \text{false} \quad \forall n \neq m \tag{3.3}$$

where R_n and R_m are adjacent.

These conditions can be summarized as

1. the partition has to cover the entire image
2. each region has to be homogeneous with respect to the predicate P
3. two adjacent region cannot be merged into a single region that satisfies the predicate P

The homogeneity predicate measures similarity in the selected features representing each region such as color and/or texture information.

Segmentation is an extremely important operation in many applications of image processing and analysis. In general, automated segmentation is one of the most difficult tasks in the image analysis, because a false segmentation will cause degradation of the subsequent image processing steps such as measurement, classification and recognition, therefore impacting the interpreting and understanding of the image. As mentioned above, the essential goal of segmentation is to decompose an image into regions which are meaningful for a given application. Some of the practical applications of image segmentation include: medical imaging to locate tumors, measuring tissue volumes and computer-guided surgery; locating objects in satellite images such as roads and forests; face and fingerprint recognition; traffic control systems. General-purpose methods and techniques have been developed over the years for image segmentation. Due to the lack of general solution to the segmentation problem, these methods can only solve the image segmentation problem for a particular problem domain when combined with specific domain knowledge.

Segmentation techniques for grayscale image have been studies quite extensively. Most of these approaches are based on either discontinuity and/or uniformity of grayscale values in an image region. Methods based on discontinuity partition an image by detecting isolated points, lines and edges according to sudden changes in grayscale levels. Methods based on uniformity include thresholding, clustering, region growing, and region splitting and merging. These methods are discussed in survey papers such as [63–66].

In [64], various existing segmentation techniques before the 1980s were categorized into three groups: edge detection, region extraction, and characteristic feature thresholding or clustering. Both parallel and sequential edge detection techniques were presented. Methods for region merging, region splitting and combination of the two were introduced. The threshold selection schemes based on gray level histogram and local properties, as well as structural and syntactic techniques were also described.

In [63] segmentation techniques were grouped into six major classes: measurement space guided spatial clustering, single linkage region growing methods, hybrid linkage region growing methods, spatial clustering methods, and split-merge methods. These methods were compared based on region merge error, blocky region boundary and memory usage. Several thresholding techniques were evaluated in [66] based on uniformity and shape measures. It categorized global thresholding methods into two groups: point-dependent techniques, and region-dependent techniques. Other segmentation methods such as fuzzy set approaches, neural network based methods, Markov Random Field (MRF) based methods, and surface based approaches were reviews in [65]. It also included range image and magnetic resonance image (MRI) in addition to most commonly used pixel intensity based image.

Color image segmentations has attracted more attention due to the fact that color images provide more information than grayscale image and techniques for grayscale image can be extended to segment color images using separate color components and their transformations. In [67], the problem of using edge-based and region-based segmentation methods to color images with complex textures was analyzed. Properties of several color representations and segmentation methods in different color spaces were discussed in [68]. The authors in [69] propose to use color and texture features and clustering technique to segment images. The paper also indicates that image segmentation based on color and texture features performs well in low-resolution and compressed images. In the paper by Dorin et.al [70], a high quality edge image is produced by extracting all the significant colors. In Chapter 1 we reviewed previous work done in segmenting food images.

We have investigated various approaches to segmenting food items in an image such as connected component labeling, active contours, normalized cuts, semi-automatic methods and multilevel approach [1,25,37–39]. Since we are only interested in food objects in an image, it is more efficient to first detect regions where potential food items are located instead of segmenting the entire image. Once these regions of interest are detected, suitable segmentation techniques can be used for each of

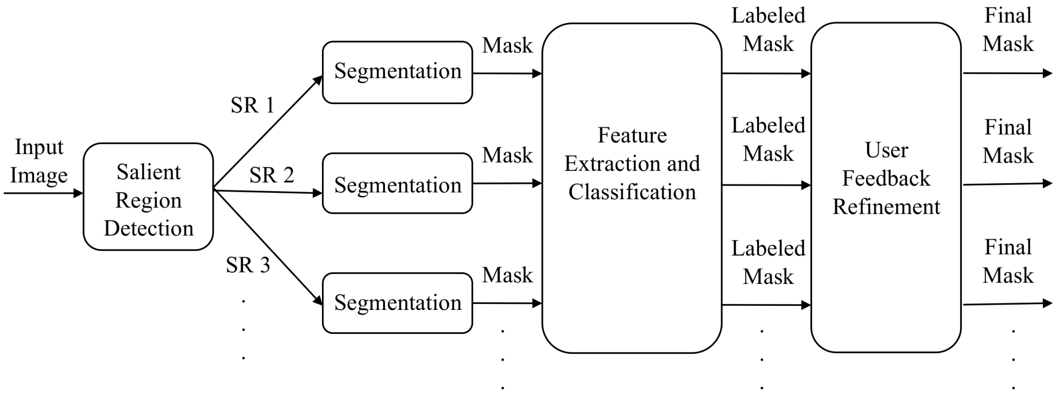


Fig. 3.1. Proposed Segmentation Approach.

these regions to find the precise boundaries of the food items. Each of these segments are classified into a particular food label using features extracted from that segment. The TADA system also includes a review step where a user can provide feedback information such as confirmation and adjustment for the location and identification of foods in the meal image. A refinement step can then be used to generate the final segmentation based on the user feedback information. Our proposed segmentation approach is illustrated in Figure 3.1. A complete description of the approach and various segmentation methods examined, as well as evaluation of proposed techniques are discussed in the following sections.

3.1 Connected Component Labeling

Region-based segmentation uses a combination of thresholding and region growing [71]. Suppose an image $f(x, y)$ is composed of light objects on a dark background in such a way that object and background pixels have intensity levels grouped into two dominant values. One simple way to segment objects from the background is to select a threshold T that separates these modes. Then any point (x, y) for which

$f(x,y) \geq T$ is known as an *object point*; otherwise, the point is known as a *background point*. The thresholded image $g(x,y)$ is defined as

$$g\left(x,y\right) = \begin{cases} 1 & \text{if} f\left(x,y\right) \geq T \\ 0 & \text{if} f\left(x,y\right) < T \end{cases} \tag{3.4}$$

We chose T by trail and error.

The method of connected component labeling scans each pixel in an image, generally from top to bottom and left to right, to identify regions containing connect pixels. These regions contain adjacent pixels that share the same membership criterion such as pixel intensity, color or texture. Connected component labeling is used on binary or graylevel images where different measures of connectivity are possible. Assume we have a binary input image $I = (0,1)$ and use a 4-connected neighborhood for our pixels adjacent relationship. The connected components labeling method scans the image along each row until it encounters a pixel p, which denotes a pixel to be labeled and has pixel intensity $I = 1$. When this occurs, it examines the four neighbors of p which have already been encountered during the scan. These four neighbors are to the left of pixel p, to the right of it, above and below it. Based on such information, p is labeled as the following:

- If all four neighbors are 0, p is assigned with a new label, or
- if only one neighbor has pixel intensity $I = 1$, p is assigned with its label, or
- if more than one of the neighbors have pixel intensity $I = 1$, p is assigned with one of the labels and a note is made of the equivalences.

Upon the completion of the scan, pixels with the same label are sorted into same classes, and a unique label is given to each class.

In our implementation, in collaboration with Anand Mariappan, we investigated a two step approach to segment food items using connected components [25]. In the first step the color image is converted to grayscale and thresholded to form a binary image. The goal here is to separate the plate from the tablecloth. The plate was empirically found assuming it was brighter than the table cloth (similar process can be used if the plate is darker than the tablecloth). For segmenting the food items on the plate, the binary image is searched in 8-point connected neighbors for the low intensity value (i.e. 0) in the thresholded image. Since we used a fixed threshold, pixels corresponding to the food items might be labeled as the plate. As a result, we need to refine the estimates of the food locations. Next, the RGB image is converted to the YCbCr color space. Using the chrominance components, Cb and Cr, the mean value of the histogram corresponding to the plate was found. Pixel locations which were not segmented during the first step were compared with the mean value of the color space histogram of the plate to identify potential food items. These pixels were given a different label from that of the plate, then 8-point connected neighbors for the labeled pixels were searched to segment the food items.

3.2 Active Contours

The technique known as active contours 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. Two main approaches are generally used in active contours based on the implementation of *snakes* and *level sets*. Snakes explicitly move predefined snake points based on an energy minimization scheme, while level set approaches move contours explicitly as a particular level of a function. 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 based on the image gradient is often used to find boundaries of image regions and to attract the contours to these boundaries. Instead of searching for geometrical boundaries, region-based active contours usually use statistical information of the image intensity to evolve the contours.

3.2.1 Snakes

The first model of active contour was proposed by Kass, et al. [72] where a snake model, a controlled continuity spline, was described. We can define a contour parameterized by arc length s as

$$C\left(s\right) \equiv \left\{\left(x\left(s\right), y\left(s\right)\right) : 0 \leq s \leq L\right\} : \Re \to \Omega \tag{3.5}$$

where L denotes the length of the contour C and Ω the entire domain of an image $I(x, y)$. The corresponding discrete domain approximation is

$$C\left(n\right) = \left\{\left(x\left(n\right), y\left(n\right)\right) : 0 \leq n \leq N\right\} \tag{3.6}$$

An energy function $E(C)$ can be defined on the contour as

$$E\left(C\right) = E_{int} + E_{ext} \tag{3.7}$$

where E_{int} and E_{ext} denote *internal* and *external* energy functions. The internal energy constrains shape of the contour, e.g. to encourage smoothness. An example of the internal energy is a quadratic functional given by

$$E_{int} = \sum_{n=0}^{N} \alpha \left|C'\left(n\right)\right|^2 + \beta \left|C''\left(n\right)\right|^2 \tag{3.8}$$

where α controls the elasticity of the contour, and β controls stiffness of the contour. The external energy encourages matching to suitable image features such as strong edges, and can be defined as

$$E_{ext} = \sum_{n=0}^{N} E_{img}\left(C\left(n\right)\right) \tag{3.9}$$

where $E_{img}(x, y)$ denotes a scalar function defined on the image plane so that the local minimum of E_{img} attracts the snakes to edges. A common example is a function of image gradient, given by

$$E_{img}\left(x, y\right) = \frac{1}{\lambda \left|\nabla G_\sigma * I\left(x, y\right)\right|} \tag{3.10}$$

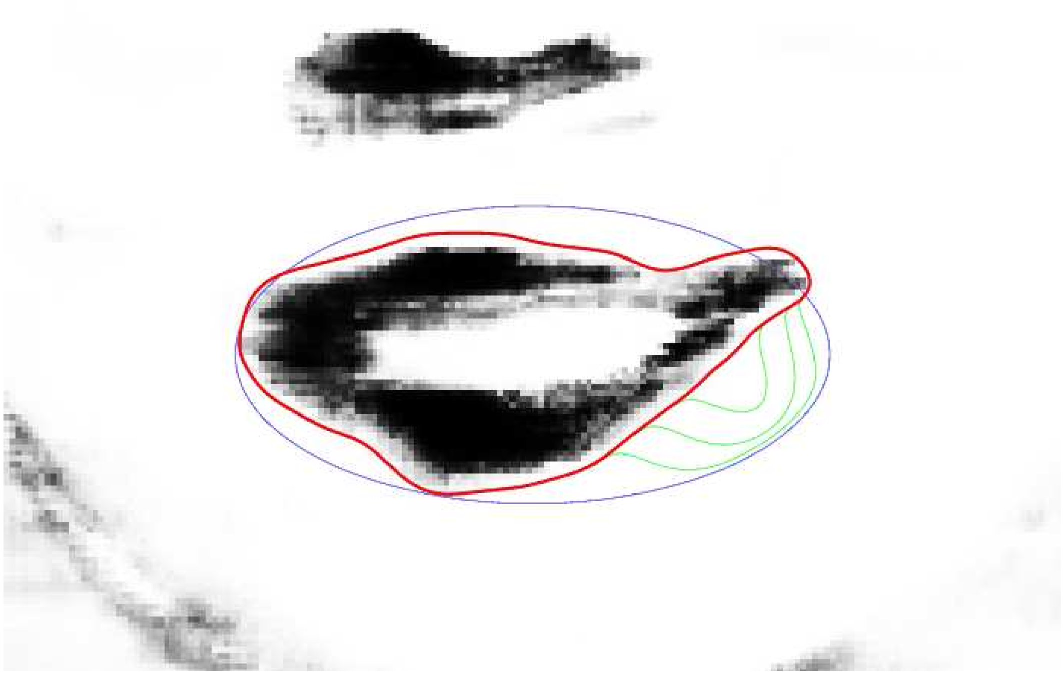


Fig. 3.2. An example of classic snakes.

where G_σ denotes a Gaussian smoothing filter with standard deviation σ, and λ is a suitable constant. The goal is to find the contour C that minimizes the total energy E with the given set of weights α, β, and λ. Figure 3.2 shows an example of classic snakes [72], where the blue curve indicates the initial contour position, three green curves shows the intermediate positions, and the red curve shows the final position after 4 iterations to located the lips.

The classic snakes provides an accurate location of the edge only if the initial contour is given sufficiently near the edges because only the local information along the contour is used. However, estimating a proper position of initial contours without prior knowledge is a difficult problem. Also, the classic snakes cannot detect more than one boundary simultaneously because the snakes maintain the same topology during the evolution stage. As a result, snakes cannot merge from multiple initial contours or split to multiple boundaries. The level set methods provide a solution for solving the snakes problem.

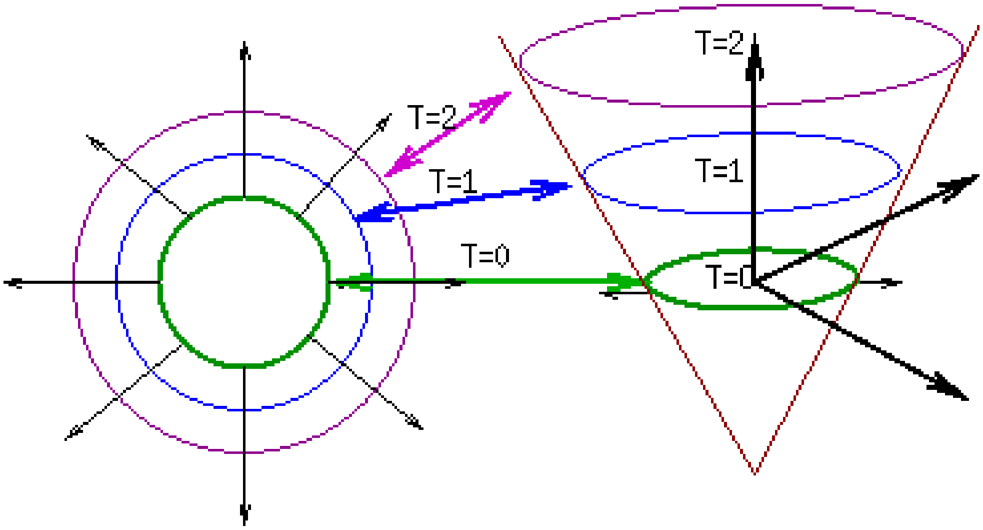


Fig. 3.3. Level set evolution and the corresponding contour propagation.

3.2.2 Level Set Methods

The level set method was proposed by Osher and Sathian [73] to implement active contours. A contour is represented implicitly via a two-dimensional Lipschitz-continuous function $\phi(x, y) : \Omega \rightarrow \Re$ defined on the image plane. The function $\phi(x, y)$ is called *level set function*, and the zero level set of $\phi(x, y)$ is defined as the contour, such that

$$C \equiv \{(x, y) : \phi(x, y) = 0\}\,, \qquad \forall\,(x, y) \in \Omega \tag{3.11}$$

where Ω denotes the entire image plane. Figure 3.3 shows an example of the evolution of the level set function $\phi(x, y)$ and the propagation of the corresponding contours C. As the level set function $\phi(x, y)$ increases from its initial stage $T = 0$, the corresponding set of contours C also propagates toward outside. As a result of this definition, the evolution of the contour is equivalent to the evolution of the level set function, i.e. $\partial C/\partial t = \partial \phi\,(x, y)\,/\partial t$. The advantage of using the zero level is that a contour can be defined as the border between a positive area and a negative area, so the contours can be identified by checking the sign of $\phi(x,y)$. The initial level set function $\phi_0(x,y) : \Omega \rightarrow \Re$ may be given by the signed distance form the initial contour as

$$\begin{aligned} \phi_0\left(x,y\right) &\equiv \left\{\phi\left(x,y\right) : t=0\right\} \\ &= \pm D\left(\left(x,y\right), N_{x,y}\left(C_0\right)\right), \qquad \forall\left(x,y\right)\in\Omega \end{aligned} \tag{3.12}$$

where $\pm D(a,b)$ denotes a signed distance between x and y, and $N_{x,y}(C_0)$ denotes the nearest neighbor pixel on initial contours C_0 from (x,y).

The deformation of the contour is generally represented in numerical form as a partial differential equation. A formulation of contour evolution using the magnitude of the gradient of $\phi(x,y)$ was first proposed by Osher and Sethian [73], given by

$$\frac{\partial\phi\left(x,y\right)}{\partial t} = \left|\nabla\phi\left(x,y\right)\right|\left(\nu+\varepsilon\kappa\left(\phi\left(x,y\right)\right)\right) \tag{3.13}$$

where ν denotes a constant speed term to push or pull the contour, $\kappa(\cdot)$ denotes the mean curvature of the level set function $\phi(x,y)$ given by

$$\begin{aligned} \kappa\left(\phi\left(x,y\right)\right) &= \operatorname{div}\left(\frac{\nabla\phi}{\|\nabla\phi\|}\right) \\ &= \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{\left(\phi_x^2+\phi_y^2\right)^{3/2}} \end{aligned} \tag{3.14}$$

where ϕ_x and ϕ_{xx} denote the first-order and second-order partial derivatives of $\phi(x,y)$ with respect to x, and ϕ_y and ϕ_{yy} denote the same with respect to y. The role of the curvature term is to control the regularity of the contours similar to the internal energy term E_{int} does in the classic snakes model, and ε controls the balance between the regularity and robustness of the contour evolution.

Another form of contour evolution was proposed by Chan and Vese [74]. The length of the contour $|C|$ can be approximated by a function of $\phi(x,y)$ [75] as

$$\begin{aligned} |C| \approx \mathrm{L}_\varepsilon\left\{\phi\left(x,y\right)\right\} &= \int_\Omega \left|\nabla H_\varepsilon\left(\phi\left(x,y\right)\right)\right| dxdy \\ &= \int_\Omega \delta_\varepsilon\left(\phi\left(x,y\right)\right)\left|\nabla\phi\left(x,y\right)\right| dxdy \end{aligned} \tag{3.15}$$

where $H_{\varepsilon}(\cdot)$ denotes the regularized form of the unit step function (often referred to as the *heaviside* function) $H(\cdot) : \Omega \rightarrow \Re$ given by

$$H\left(x,y\right) = \begin{cases} 1, & \text{if} \phi\left(x,y\right) \geq 0 \\ 0, & \text{if} \phi\left(x,y\right) < 0 \end{cases} \quad \forall\left(x,y\right) \in \Omega \tag{3.16}$$

and δ_{ε} denotes the derivative of $H_{\varepsilon}(\cdot)$. Since the unit step function produces either 0 or 1 depending on the sign of the input, the derivative of the unit step function produces non-zero only when $\phi(x,y) = 0$, i.e. on the contour C. Consequently, the integration shown in Equation 3.15 is equivalent to the length of the contours on the image plane. The associated Euler-Lagrange equation [76] obtained by minimizing $L_{\varepsilon}(\cdot)$ with respect to ϕ and parameterizing the descent directions by time t is given by

$$\frac{\partial \phi\left(x,y\right)}{\partial t} = \delta_{\varepsilon}\left(\phi\left(x,y\right)\right)\kappa\left(\phi\left(x,y\right)\right) \tag{3.17}$$

The contour evolution motivated by the equation above can be interpreted as the motion by mean curvature minimizing the length of the contour. Therefore, Equation 3.13 is considered as the motion motivated by PDE, while Equation 3.17 is considered as the motion motivated by energy minimization.

An outstanding characteristic of level set methods is that contours can split or merge as the topology of the level set function changes. Therefore, level set methods can detect more than one boundary simultaneously, and multiple initial contours can be placed. This flexibility and convenience provide a means for an autonomous segmentation by using a predefined set of initial contours. On the other hand, the computational cost of level set methods is high because the computation should be done on the same dimension as the image plane Ω. Thus, the convergence speed is relatively slower than other segmentation methods, particularly local filtering based methods. The use of multiple initial contours increases the convergence speed by co-operating with neighbor contours quickly. Level set methods with faster convergence, called *fast marching methods* [77], have been studied intensively for the last decade. Because of these attractive properties, our active contour model uses the level set method.

3.2.3 Edge-Based Active Contours

Edge-based active contours are closely related to the edge-based segmentation. Most edge-based active contour models consists of two parts: the regularity part, which determines the shape of contours, and the edge detection part, which attracts the contour towards the edges.

Geometric active contour models were proposed by Caselles et al. [78] adding an additional term, known as the *stopping function*, to the speed function shown in Equation 3.13. Malladi et al. [79] proposed a similar model given by

$$\frac{\partial \phi\left(x,y\right)}{\partial t} = g\left(I\left(x,y\right)\right)\left(\kappa\left(\phi\left(x,y\right)\right)+\nu\right)\left|\nabla \phi\left(x,y\right)\right| \tag{3.18}$$

where $g(\cdot) : \Omega \rightarrow \Re$ denotes the stopping function, i.e. a positive and decreasing function of the image gradient. A simple example of the stopping function is given by

$$g\left(I\left(x,y\right)\right) = \frac{1}{1+\left|\nabla I\left(x,y\right)\right|^{n}} \tag{3.19}$$

where n is given as 1 in [79]. The contour moves in the normal direction with a speed of $g\left(I\left(x,y\right)\right)\left(\kappa\left(\phi\left(x,y\right)\right)+\nu\right)$, therefore stops on the edges, where $g(\cdot)$ vanishes. The curvature term $\kappa(\cdot)$ maintains the regularity of the contours, while the constant term ν accelerates and keeps the contour evolution by minimizing the enclosed area.

The *Geodesic active contour* model was proposed by [80] after the geometric active contour model. Yezzi et al. [81] also proposed a similar active contour model. Based on the principle of classic dynamic systems, solving the active contour problem is equivalent to finding a path of minimal distance, called *geodesic curve* given by

$$\frac{\partial C}{\partial t} = \left(g\left(I\left(x,y\right)\right)\kappa\left(\phi\left(x,y\right)\right) - \nabla g\left(I\left(x,y\right)\right)\cdot \mathcal{N}\right)\mathcal{N} \tag{3.20}$$

where $\mathcal{N}$ denotes the inward unit normal given by

$$N = -\frac{\nabla \phi}{\left\|\nabla \phi\right\|} \tag{3.21}$$

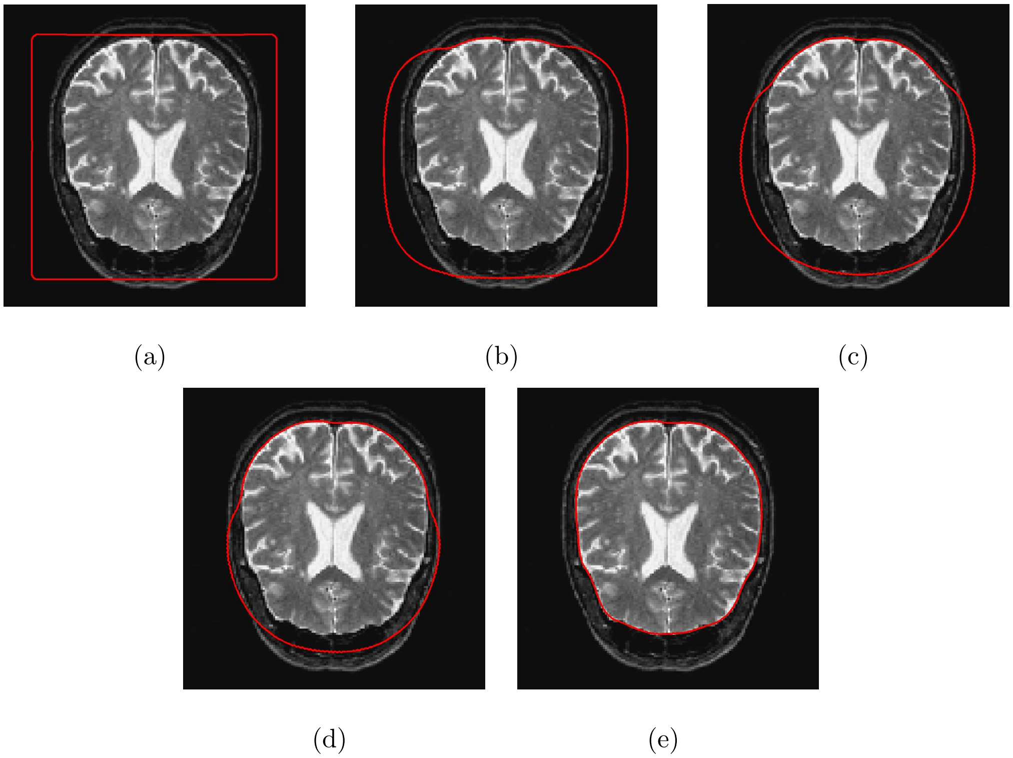


Fig. 3.4. Segmentation with geodesic active contours.

From the relation between a contour and a level set function and the level set formulation of the steepest descent method, solving this geodesic problem is equivalent to searching for the steady state of the level set evolution equation [80] given by

$$\frac{\partial\phi\left(x,y\right)}{\partial t}=g\left(I\left(x,y\right)\right)\left(\kappa\left(\phi\left(x,y\right)\right)+\nu\right)\left|\nabla\phi\left(x,y\right)\right|+\nabla g\left(I\left(x,y\right)\right)\cdot\nabla\phi\left(x,y\right)\quad(3.22)$$

We can notice the geodesic active contour model shown in Equation 3.22 is identical to the geometric active contour model shown in Equation 3.18 except for the additional term $\nabla g\left(I\left(x,y\right)\right)\cdot\nabla\phi\left(x,y\right)$. Figure 3.4 shows an example of image segmentation using geodesic active contours. Geodesic active contour have been the most popular methods among the edge-based active contour models and their applications have been extended to multispectral images by Sapiro.

Color snakes is a geodesic active contour model particularly for multispectral images propose by Sapiro [82, 83]. In order to detect the edges in multispectral images, a special gradient function based on Riemannian geometry, known as the *color gradient* function, is used instead of the traditional image gradient function. A simple example of the color gradient function is given by

$$g\left(I\left(x,y\right)\right)=\frac{1}{1+\left(\lambda_{+}-\lambda_{-}\right)} \tag{3.23}$$

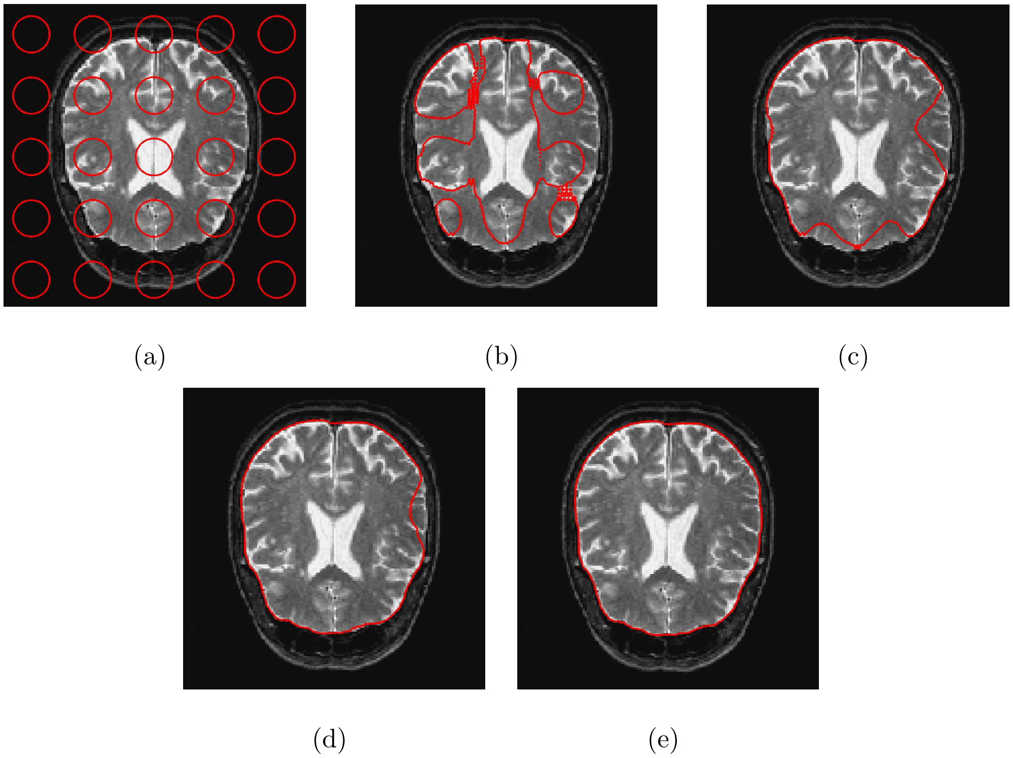
where λ_{+} and λ_{-} represent the maximal and minimal rate of changes on the multi-spectral image $I(x, y)$, respectively.

Due to the structure of the speed functions and the stopping functions in Equation 3.13 and Equation 3.22, edge-based active contour models have a few disadvantages compared to the region-based active contour models. Because of the constant term ν, edge-based active contour models evolve the contour towards only one direction, either inside or outside. Therefore, an initial contour should be placed completely inside or outside of ROI, and some level of a prior knowledge is still required. Also, edge-based active contours inherit some disadvantages of the edge-based segmentation methods due to the similar technique used. Since both edge-based segmentation and edge-based active contours rely on the image gradient operation, edge-based active contours may skip the blurry boundaries, and are sensitive to local minima or noise as edge-based segmentation methods often do.

3.2.4 Region-Based Active Contours

Most region-based active contour models consist of two parts: the regularity part, which determines the smooth shape of contours, and the energy minimization part, which searches for uniformity of a desired feature within a subset. A nice characteristic of region-based active contours is that the initial contours can located anywhere in the image as region-based segmentation relies on the global energy minimization rather than local energy minimization. Therefore, less prior knowledge is required than edge-based active contours.

The *Piecewise-constant active contour* model was proposed by Chan and Vese [74] using the *Mumford-shah segmentation* model [84]. Piecewise-constant active contour model moves deformable contours minimizing an energy function instead of searching



(a) (b) (c)

(d) (e)

Fig. 3.5. Segmentation with Chan-Vese active contour without edges.

edges. A constant approximates the statistical information of image intensity within a subset, and a set of piecewise-constant approximate the statistics of image intensity along the entire domain of an image. The energy function measures the difference between the piecewise-constant and the actual image intensity at every image pixel. The level set evolution equation is given by

$$\frac{\partial \phi\left(x,y\right)}{\partial t} = \delta_{\varepsilon}\left(\phi\left(x,y\right)\right)\left[\nu\kappa\left(\phi\left(x,y\right)\right) - \left\{\left(I\left(x,y\right)-\mu_1\right)^2 - \left(I\left(x,y\right)-\mu_0\right)^2\right\}\right] \quad (3.24)$$

where μ_0 and μ_1 denote the mean of the image intensity within the two subsets, i.e. the outside and inside of contours, respectively. The final partitioned image can be represented as a set of piecewise-constants, where each subset is represented as a constant. This method has shown fastest convergence speed among region-based active contours due to the simple representation. Figure 3.5 shows an example of image segmentation using Chan-Vese active contour without edges.

The *Piecewise-smooth active contour* model was proposed by Tsai et al. [85]. The same segmentation principles used for piecewise-constant model partitions an image, but a smoothed partial image instead of constants represent each subset. The evolution equation is given by

$$\frac{\partial \varphi\left(x,y\right)}{\partial t} = \delta_{\varepsilon}\left(\varphi\left(x,y\right)\right) \left[\begin{array}{c} \nu\kappa\left(\varphi\left(x,y\right)\right) - \left\{ \begin{array}{c} \left(I\left(x,y\right) - \mu_1\left(x,y\right)\right)^2 \\ -\left(I\left(x,y\right) - \mu_0\left(x,y\right)\right)^2 \end{array} \right\} \\ -\omega\left(\left|\nabla \mu_1\left(x,y\right)\right|^2 - \left|\nabla \mu_0\left(x,y\right)\right|^2\right) \end{array} \right] \tag{3.25}$$

where $\mu_0(x,y)$ and $\mu_1(x,y)$ denote the smoothed images within the outside and inside of contours, respectively.

Although traditional region-based active contours partition an image into multiple sub-regions, those multiple regions belong to only two subsets: either the inside or outside of contours. Chan and Vese proposed *multi-phase active contour* model [76], which increases the number of subsets that active contours can find simultaneously. Multiple active contours evolve independently based on the piecewise-constant model shown in Equation 3.24 or the piecewise-smooth model shown in Equation 3.25, the multiple subsets are defined by a group of disjoint combination of the level set functions. For example, N level set functions define maximum 2^N subsets of the entire region. An example of subsets defined by 4-phase active contour is

$$\left[\begin{array}{c} \Omega_0 \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{array} \right] \equiv \left\{ \left(x,y\right) : \left[\begin{array}{c} \phi_2\left(x,y\right) < 0, \phi_1\left(x,y\right) < 0 \\ \phi_2\left(x,y\right) < 0, \phi_1\left(x,y\right) > 0 \\ \phi_2\left(x,y\right) > 0, \phi_1\left(x,y\right) < 0 \\ \phi_2\left(x,y\right) > 0, \phi_1\left(x,y\right) > 0 \end{array} \right] \right\} \tag{3.26}$$

where $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$ denote the four subsets defined by two level set functions $\{\phi_1, \phi_0\}$, i.e. two active contours. The level set evolution equation for this case is given by

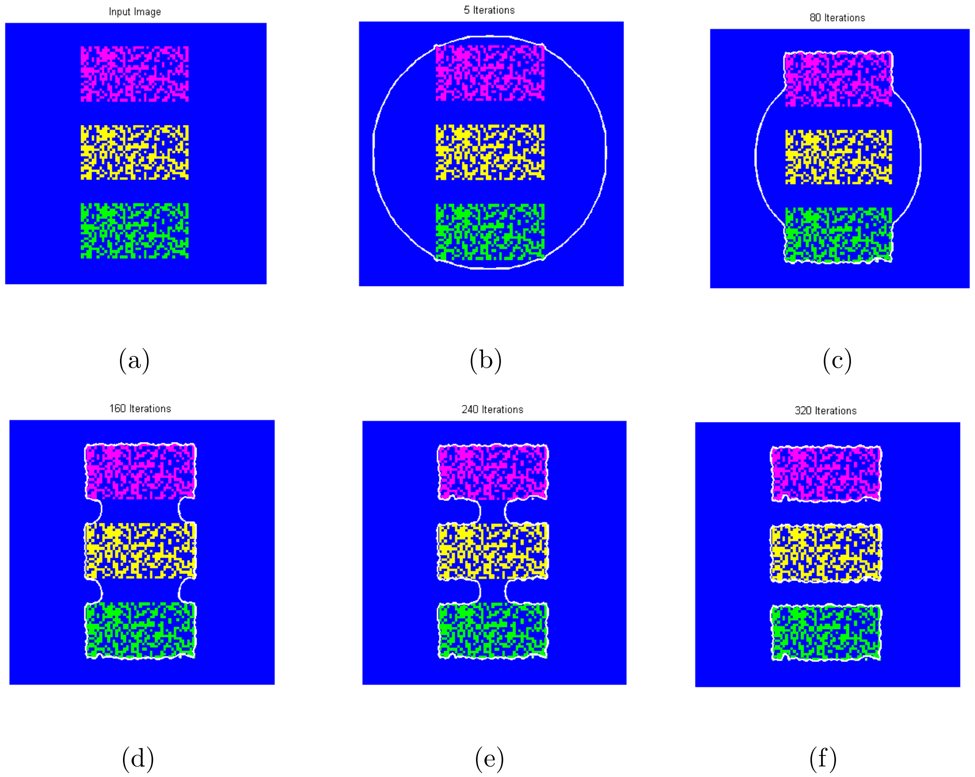
$$\begin{aligned} \frac{\partial\phi_1(x,y)}{\partial t} &= \delta_\varepsilon\left(\phi_1\left(x,y\right)\right)\left\{\nu\kappa\left(\phi_1\left(x,y\right)\right) - \left[\begin{array}{c} \left\{\begin{array}{c} \left(I\left(x,y\right)-\mu_3\right)^2 - \\ \left(I\left(x,y\right)-\mu_2\right)^2 \end{array}\right\} H_2 + \\ \left\{\begin{array}{c} \left(I\left(x,y\right)-\mu_1\right)^2 - \\ \left(I\left(x,y\right)-\mu_0\right)^2 \end{array}\right\} \left(1-H_2\right) \end{array}\right]\right\} \\ \frac{\partial\phi_2(x,y)}{\partial t} &= \delta_\varepsilon\left(\phi_2\left(x,y\right)\right)\left\{\nu\kappa\left(\phi_2\left(x,y\right)\right) - \left[\begin{array}{c} \left\{\begin{array}{c} \left(I\left(x,y\right)-\mu_3\right)^2 - \\ \left(I\left(x,y\right)-\mu_1\right)^2 \end{array}\right\} H_1 + \\ \left\{\begin{array}{c} \left(I\left(x,y\right)-\mu_2\right)^2 - \\ \left(I\left(x,y\right)-\mu_0\right)^2 \end{array}\right\} \left(1-H_1\right) \end{array}\right]\right\} \end{aligned} \tag{3.27}$$

where $H_n \equiv H_\epsilon(\phi_n(x,y))$ and $\{\mu_0, \mu_1, \mu_2, \mu_3\}$ denote the mean of image intensity within each corresponding subsets $\{\Omega_0, \Omega_1, \Omega_2, \Omega_3\}$. Multi-phase active contours provide a means to integrate segmentation and pattern classification tasks. An image is partitioned into multiple sub-regions and they simultaneously identify those regions into classes. Depending on whether training samples are provided or not, supervised or unsupervised segmentation can actually perform supervised or unsupervised pattern classification. This provides a way to the autonomous pattern classification technology reducing the number of procedures and processing time.

The same segmentation principle can be extended to multispectral images by taking the mean of energy functions measured at each band [86]. The level set evolution equation of 2-phase active contour model is given by

$$\frac{\partial\varphi\left(x,y\right)}{\partial t} = \delta_\varepsilon\left(\varphi\left(x,y\right)\right)\left[\nu\kappa\left(\varphi\left(x,y\right)\right) - \frac{1}{N}\sum_{n=1}^{N}\left\{\begin{array}{c} \omega_{1n}\left(I_n\left(x,y\right)-\mu_{1n}\right)^2 \\ -\omega_{0n}\left(I_n\left(x,y\right)-\mu_{0n}\right)^2 \end{array}\right\}\right] \tag{3.28}$$

where $\mu_i = \left[\mu_{i1}, \mu_{i2}, ..., \mu_{in}, ..., \mu_{iN}\right]^T$ denote the mean value of the vector valued image intensity $I(x,y)$ within the corresponding subset Ω_i. $\omega_i > 0$ are parameters for each band. We adopted this method and applied to color images in RGB color space.



(a) (b) (c)

(d) (e) (f)

Fig. 3.6. Segmentation with Chan-Vese active contour on vector-valued image.

Figure 3.6 shows an example of applied the method to the RGB channels of a vector-valued image.

Due to the global energy minimization, region-based active contours generally do not have any restriction on the placement of the initial contours. That is, region-based active contour can detect interior boundaries regardless of the position of initial contours. The use of pre-defined initial contours provides a method of autonomous segmentation. They are also less sensitive to local minima or noise than the edge-based active contours. However, due to the assumption of uniform image intensity, most methods are applicable only to images where each subset is representable by a simple expression, e.g. single Gaussian distribution or a constant. If a subset consists of multiple distinctive sub-classes, these methods would produce over-segmented or under-segmented results.

We use the region-based active contours approach in some of the controlled diet studies done by the nutritionist where simple types of food are given to test subjects for evaluation. In addition, the review tool in the mpFR application allows a participant to use a pen drawing tool to create a rough contour of any food object where initial analysis fails to locate and/or identify it. Such user feedback information contains lists of x, y coordinates associated with the pen drawing tool. These points are connected using linear interpolation method to create an initial contour needed to apply the active contours model. Therefore, final segmentation mask for each food objects is generated by refining the segmentation results from the the automatic image analysis as illustrated in Figure 3.1. In particular, we use the Chan-Vese model [86] for color images in the RGB color space as describe in Equation **??**. $N = 3$ denotes the three color components, μ_{1n} and μ_{0n} corresponds to the mean value of each color component inside and outside of the contour. $\omega_{1n} > 0$ and $\omega_{0n} > 0$ specifies weighting parameters for each color component.

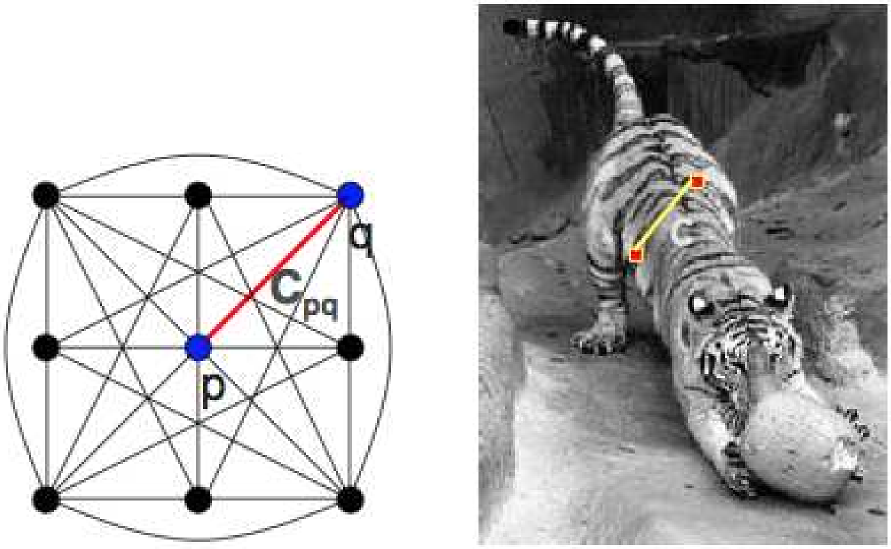
The active contours model works well when the food items are separated from each other, however, it sometimes fails to distinguish multiple food items that are connected. Due to the restriction of the current model to separate only two regions, images with more than one object region cannot, therefore, be captured by the model. Multiphase models in combination with other segmentation techniques such as region competition have been proposed to solve the multiregion segmentation problem; however, the initialization problem associated with active contour models is still a challenge. As a result, we investigated other techniques to extract multiple objects in an image, in particular methods based on graph theory.

3.3 Graph-Theoretical Methods

The goal of graph theory based approaches is to partition a graph describing the entire image into a set of connected components that correspond to image regions. When an image is set equivalent to a graph, the set of points in an arbitrary feature space are represented as weighted undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where the nodes of the graph are points in the feature space, and an edge is formed between every pair of nodes. **V** is a set of *vertices* or *nodes*, each node represents one image element. The image elements can be individual pixels, small regions, or other types of image features. Usually only elements within a small neighborhood are connected. This provides spatial coherence and has computational advantages. **E** is a set of *edges* linking neighboring nodes together. The *weight* of the edge is proportional to the similarity between the vertices it joins together. The edge weight is also known as pairwise *similarity*, or pairwise *affinity*. In general, given a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ graph based segmentation methods attempt to find groups of nodes in **G** that are strongly connected to one another, but weakly connected to the rest of the graph. Figure 3.7 shows a fully-connected graph and an example of the the similarity measure between two nodes. There are generally two methods for grouping the nodes. The *splitting* methods partition a graph by removing redundant edges while the *region growing* methods join components based on the attributes of nodes and edges. Next, we describe some graph partitioning approaches.

3.3.1 Review of Graph Partitioning Approaches

The most efficient graph-based segmentation methods use fixed thresholds and local measures to find regions. For example, the approach in [87] describes a method by breaking large edges in a *Minimum Spanning Tree* (MST) of the graph. Given a connected and undirected graph, a spanning tree of that graph is a subgraph connecting all the vertices together like a tree. Multiple spanning trees can exist in a single graph. Each edge is assigned with a weight, which is used to assign a weight to a spanning tree by summing the weights of all edges in the spanning tree. A *minimum spanning tree* can then be defined as a spanning tree whose weight is less than or equal to the weight of other spanning trees. Such a tree can be constructed efficiently using Kruskals method, where the process starts with the completely disconnected



(a)

(b)

Fig. 3.7. Images as graphs. (a) is a fully-connect graph, every pixel is considered to be a node, C_{pq} is measures the similarity between every pair of pixels, p, q, (b) shows an example of the similarity measure between two nodes.

graph, edges are added in increasing order of weight as long as doing so does not introduce a cycle. The process stops when all the vertices are connected.

Felzenszwalb and Huttenlocher [88] proposed the local variation method that partitions the image so that for any pair of regions, the variation between the regions should be larger than the variation within the regions. They introduced a simple but effective modification of Kruskal's algorhtim. During processing, each MST C_i is associated with a threshold

$$T\left(C_i\right) = w\left(C_i\right) + k/\left|C_i\right| \tag{3.29}$$

where $w(C_i)$ called the *local variation* of C_i is the maximum weight in the spanning tree C_i. $|C_i|$ is the number of pixels in C_i and $k > 0$ is a constant. To process edge

(x_k, x_l) whose two endpoints are in two separate MSTs, C_i and C_j, these MSTs can be merged by adding the edge (x_k, x_l) only if

$$w\left(x_k, x_l\right) \leq \min\left(T\left(C_i\right), T\left(C_j\right)\right) \tag{3.30}$$

Note that, as the size of C_i increases, Equation 3.29 and 3.30 dictate an increasingly tight affinity upper bound $T(C_i)$ of an edge merging C_i with another region. Sorting the edges according to weight causes the method to connect relatively homogeneous regions first. The merging process is very sensitive to the local variation in the merged regions. Because of the increasingly tight bound as seen in 3.29, a large homogeneous region C_i can only be merged by edges having weights that are larger than the largest affinity $w(C_i)$ in the MST C_i. This encourages growth of small regions C_i due to this loose bound. As a result, the approach tends to produce narrow regions near the “true segment boundaries, but can be computed very efficiently.

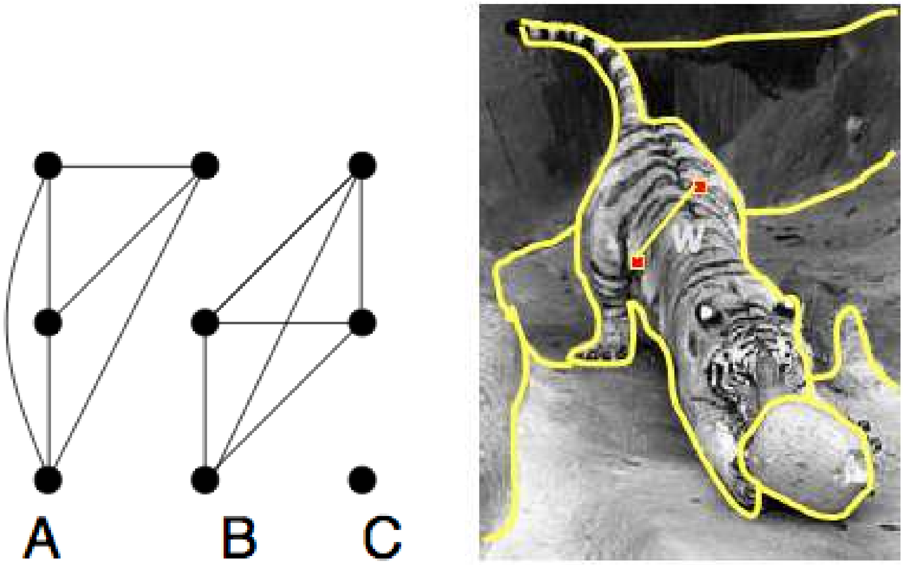
Another method [89] is based on computing the *minimum cut* in the graph representing an image. The criterion for a cut is designed to minimize similarity between split regions. This approach shows advantage of capturing non-local properties of the image. A *cut* through a graph is defined as the total weight of the links that must be removed to divide the graph into two separate components.

$$cut\left(A, B\right) = \sum_{i \in A, j \in B} w\left(i, j\right) \tag{3.31}$$

The goal is to find the cut through the graph that has the overall minimum weight

$$MinCut\left(A, B\right) = \min_{A,B}\left(cut\left(A, B\right)\right) \tag{3.32}$$

The cut should correspond to the subset of edges of least weight that can be removed to partition the graph. Since weight encodes similarity, this should be equivalent to partitioning the graph along the boundary of least similarity. The method can be computed efficiently, however, it has a preference for short boundaries and sometimes picks a trivial partition. Therefore, solution needs to be constrained to avoid such trivial cuts. Figure 3.8 shows an example of breaking graph into segments by delete links that cross between segments.



(a) (b)

Fig. 3.8. Segmentation by Graph Cuts. (a) shows a graph with deleted links that cross between segments, (b) shows an example of the graph cut method.

The S-T min-cut method [90] reduces the problem of trivial cuts by introducing two special nodes called *souce (s)* and *sink (t)*. s and t are linked to some image nodes by links of very large weight so that they will never be selected in a cut. Then the goal is to seek a minimum cut separating s and t. That is, we seek a partitioning of the graph into two sets of nodes F and G, with $G = V - F, s \in F, andt \in G$, such that the linkage

$$L\left(F,G\right) = \sum_{x_i \in F, x_j \in G} a\left(x_i, x_j\right) \tag{3.33}$$

is minimized. Due to the constraints, the computation of S-T min-cut problem is much simpler than the general graph partition problem. An S-T graph can be generated to fine an efficient solution to the S-T min-cut problem. Given two sets of disjoint pixels S and T, a weighted directed graph is formed as follows. Two directed edges $\langle x_i, x_j \rangle$ and the reverse edge $\langle x_j, x_i \rangle$ are included for each edge (x_i, x_j) in the undirected graph. Both of these edges are weighted by the affinity $a(x_i, x_j)$. Two additional nodes s (source) and t (sink) are created. Finally, for each $x_i \in S$ and $x_j \in T$, two infinitely weighted directed links $\langle s, x_i \rangle$ and $\langle x_j, t \rangle$ are included. The sets S and T must satisfy the following:

1. Each S and T generated must be sufficiently large, otherwise, we get a trivial cut that contains only the image nodes in either the set for S and the set for T
2. Both sets should be fully contained within 'natural' segments. This is because the sets themselves will never be partitioned due to infinite weights.
3. Enough pairs of S and T should be generated to identify most of the salient segments in the image.

Estrada et al [91] introduces a suitable generation process based on spectral properties of the affinities matrix. However, the process is computationally heavy because several hundred min-cut problems need to be solved for different combination of S and T.

3.3.2 Normalized Cut

The min-cut method often favors cutting small sets of isolated nodes in the graph due to the minimum cut criteria. Figure 3.9 illustrates an example. Assuming the edge weights are inversely proportional to the distance between the two nodes, the cut that partitions nodes n_1 and n_2 have very small value. Furthermore, any cut that partitions individual nodes on the right region will have smaller cut value than the cut that partitions the nodes into the left and right regions. To avoid such bias, Shi and Malik [92] proposed the normalized cut criterion that computes the cut cost as a fraction of the total edge connections to all the nodes in the graph that penalizes large segments. This normalized cut measure is represented as

$$NCut\,(A, B) = \frac{cut\,(A, B)}{assoc\,(A, V)} + \frac{cut\,(A, B)}{assoc\,(B, V)} \tag{3.34}$$

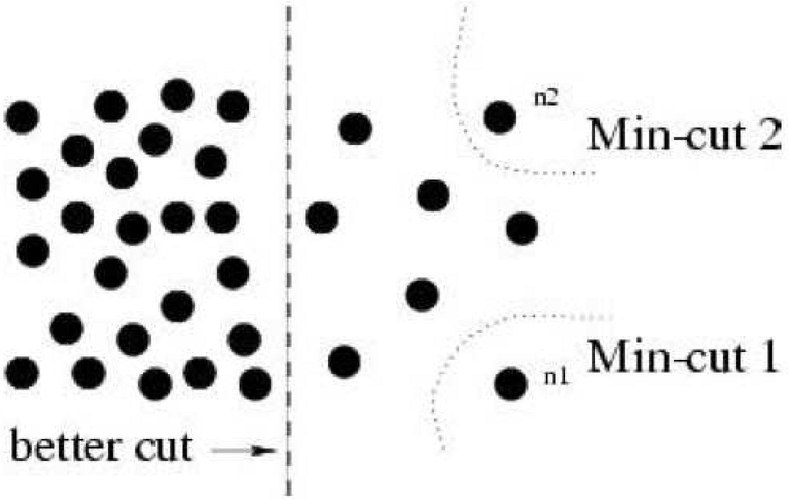


Fig. 3.9. An example where minimum cut gives a bad partition.

where the term $assoc(A, V)$ is the total weight of the connections between the region A and the rest of the nodes in the graph

$$assoc\left(A, V\right) = \sum_{i \in A, j \in V, (v_i, v_j) \in E} w\left(i, j\right) \tag{3.35}$$

Solving for the optimal NCut exactly is NP-complete, however, we can obtain an approximate solution. Let $\mathbf{W}$ be the *affinity matrix* such that $W(i, j)$ contains the weight of the edge linking nodes i and j. We also define a diagonal matrix $\mathbf{D}$, such that $D(i, i) = \sum_j w_{i,j}$ and $D(i, j) = 0$. Then it turns out that we can minimize $NCut(A, B)$ by

$$\min_{A,B} NCut\left(A, B\right) = \min_{\mathbf{y}} \frac{\mathbf{y^T}\left(\mathbf{D} - \mathbf{W}\right)\mathbf{y}}{\mathbf{y^T D y}} \tag{3.36}$$

The above equation can be minimized by solving the generalized eigenvalue problem if $\mathbf{y}$ is relaxed to take real values,

$$\left(\mathbf{D} - \mathbf{W}\right)\mathbf{y} = \lambda \mathbf{D y} \tag{3.37}$$

The above equation can be solved by converting to standard eigenvalue problem,

$$\mathbf{D}^{-\frac{1}{2}}\left(\mathbf{D} - \mathbf{W}\right)\mathbf{D}^{-\frac{1}{2}}\mathbf{z} = \lambda \mathbf{z}, \qquad \text{where} \quad \mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y} \tag{3.38}$$

It can be shown that the second smallest eigenvector is the solution to the Normalized Cut problem.

The recursive two-way NCut grouping method consists the following steps. Given an image $\mathbf{I}$, a weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ can be constructed where each node of the graph is a pixel of the image $\mathbf{I}$. Let N be the number of nodes in $|\mathbf{V}|$.

Step 1

An $N \times N$ symmetric similarity matrix $\mathbf{W}$ is constructed as the following:

$$w_{i,j} = \exp \frac{-\left\|\mathbf{F}(i) - \mathbf{F}(j)\right\|_2^2}{\sigma_I^2} * \begin{cases} \exp \frac{-\|\mathbf{X}(i)-\mathbf{X}(j)\|_2^2}{\sigma_X^2} & \text{if } \left\|\mathbf{X}(i) - \mathbf{X}(j)\right\|_2 < r \\ 0 & \text{otherwise} \end{cases} \tag{3.39}$$

where $\mathbf{X}(i)$ represents the spatial location of node i and $\mathbf{F}(i)$ is a feature vector that takes on values such as:

- $\mathbf{F}(i) = \mathbf{I}(i)$ representing the intensity value when segmenting gray scale images,
- $\mathbf{F}(i) = [v, u \cdot s \cdot sin(h), v \cdot s \cdot cos(h)](i)$, for segmenting color images where h, s, v represent color images using HSV components,
- $\mathbf{F}(i) = [|\mathbf{I} * f_1|, ..., |\mathbf{I} * f_n|](i)$, where the f_i represents individual Difference of Gaussian filter at various scales and orientations for texture segmentation.

Define $d_i = \sum_j w_{i,j}$ as the total connection from node i to all other nodes. An $N \times N$ diagonal matrix $\mathbf{D}$ can be constructed with d on its diagonal.

Step 2

Solve the generalized eigensystem,

$$(\mathbf{D} - \mathbf{W})\,\mathbf{y} = \lambda \mathbf{D}\mathbf{y} \tag{3.40}$$

and get an eigenvector with the second smallest eigenvalue.

Step 3

The eigenvector from Step 2 can be used to bipartition the graph. Ideally, the eigenvector should only take on two discrete values, thus the signs can be used to partition the graph ($\mathbf{A} = \{\mathbf{V}_i | \mathbf{y}_i > 0\}$, $\mathbf{B} = \{\mathbf{V}_i | \mathbf{y}_i \leq 0\}$). Unfortunately, because $\mathbf{y}$ contain real values, a splitting point needs to be chosen with the following options,

- Choose 0
- Choose the median
- Choose a splitting point which minimized the NCut(A,B).

The splitting point which minimizes NCut value also minimizes

$$\frac{\mathbf{y}^T\left(\mathbf{D}-\mathbf{W}\right)\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \tag{3.41}$$

where $\mathbf{y} = (1+\mathbf{x}) - b(1-\mathbf{x})$, $b = k/(1-k)$ and

$$k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i} \tag{3.42}$$

where $\mathbf{x}$ is an N dimensional indicator vector, $x_i = 1$ if node i is in A and -1, otherwise. otherwise. We need to try different values of splitting points to find the minimal NCut. Generally, the optimal splitting point is around the mean of resulting eigenvectors.

Step 4

To generate multiple segmentations, we need to repeat the bipartition recursively. The recursion stops if NCut value is larger than a pre-selected threshold, or the total number of nodes is smaller than the threshold value.

Figure 3.10 shows the second smallest to the ninth smallest eigenvalues of the system. Various image features such as intensity, color, texture, contour continuity, motion can be combined under one uniform framework. We used intensity and color as the image features for using normalized cut on food images. Methods to optimize the normalized cut criterion such as cue combination and the number of clusters are discussed in [93] to obtain meaningful segmentations.

Alternatively, one can use all the top eigenvectors to simultaneously obtain a K-way partition instead of the recursive 2-way cut described above.

1. We start with a simple clustering method, such as k-means, to obtain an over-segmented image with k regions

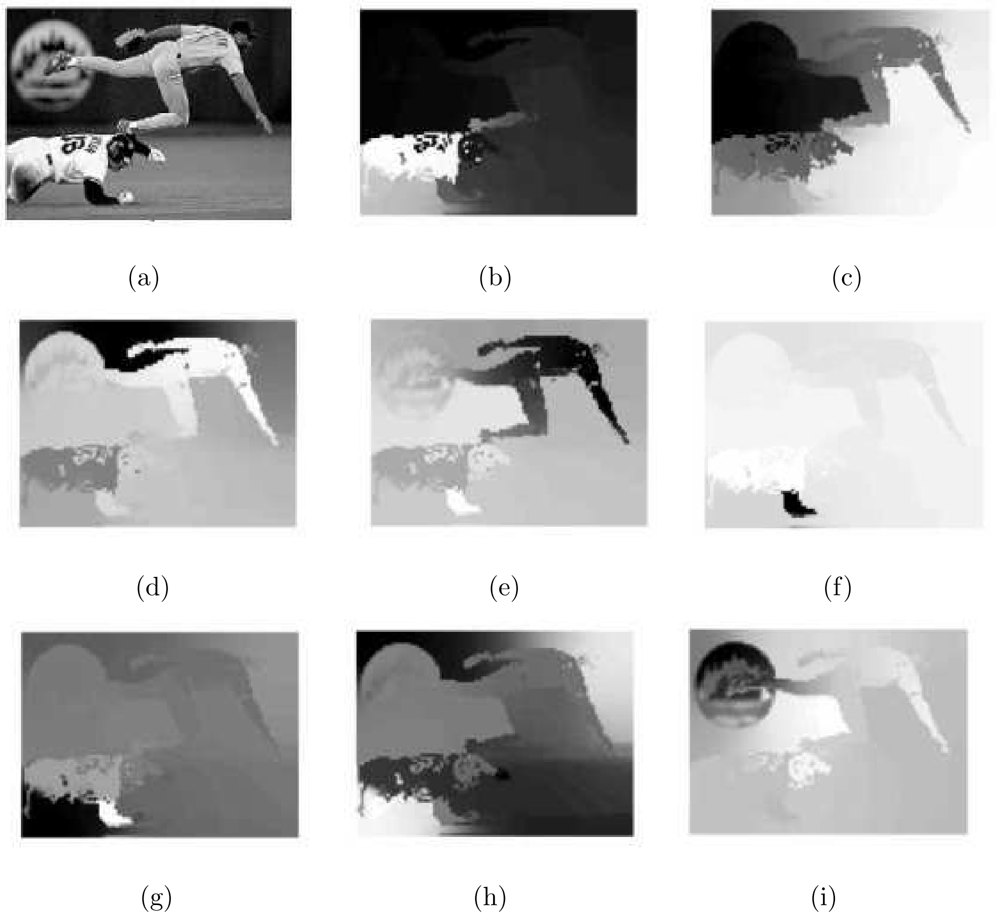


Fig. 3.10. (a) is a gray level image, (b)-(i) show the eigenvectors corresponding to the second smallest to the ninth smallest eigenvalues of the system.

2. - Apply greedy pruning to iteratively merge two segments at a time until only k segments are left, minimizing the k-way NCut criterion

$$NCut_k = \frac{cut\,(A_1, V - A_1)}{assoc\,(A_1, V)} + \frac{cut\,(A_2, V - A_2)}{assoc\,(A_2, V)} + ... + \frac{cut\,(A_k, V - A_k)}{assoc\,(A_k, V)} \tag{3.43}$$

   - Apply global recursive cut from the initial segments to build a condensed graph. Based on this graph, recursively bipartition the graph according to NCut criterion.

3.4 Semi-Automatic Segmentation

In this type of segmentation, the user outlines the regions with mouse clicks and methods are used so that the path that best fits the edge of the image is constructed.

3.4.1 Related Work

Many popular image editing programs contain semi-automatic object extraction tools. The most popular tool for extracting foreground semi-automatically in image editing programs such as Adobe Photoshop [94] is *Magic Wand*. Magic Wand starts with a small user-specified region. It then performs region growing such that all selected pixels fall within user-adjustable tolerance of the color statistics for a specified region. For natural images, finding the correct tolerance threshold is often problematic. The methods work well for images which contain few colors, such as drawings. For "natural" images that contain many colors, such as photographs, the results are unusable or the interaction required is far from being feasible.

Intelligent Scissors [95] can be used to select contiguous areas of similar color in a fashion similar to Magic Wand. Intelligent Scissors creates a selection boundary by assisting the user to create a set of connected line segments around the objects. Nodes are joined with mouse clicks using curve shapes that attempt to follow color weights. Although this method works with sub-pixel accuracy, a satisfactory segmentation is only achieved with very simple images that have clear edges.

Bayes Matting models color distributions probabilistically to achieve alpha mattes [96] base on [97]. The method uses a shrinked shape of the object and a subset of the background as input. The user uses a "brush" to coarsely redraw the shape of the input with the brush stroke having to contain both foreground and background. The method then tries to compute opacity values over the pixels marked with the brush. The main disadvantage is that for complicated objects the user must specify quite detailed shape information for the method to work properly.

Knockout-2 is a proprietary plug-in for Photoshop [98]. According to [96] the results are sometimes similar or less quality than Bayes Matting. Adobe Photoshop contains a tool called *extract*, which requires a little less user interaction. Instead of two strokes, only one thick brush strokes has to be drawn by the user, which has to cover the edge of the the object. Extract produces similar results to Knockout-2.

GrowCut [99] is a method based on cellular automaton. The classification of a pixel is partly determined by the classification of its neighbors. Doing this over many iterations, the selection will become more and more stable. Due to the large number of iterations required, this process takes more than one minute even for moderate complex images that are far from the higher resolution images captured by modern digital cameras.

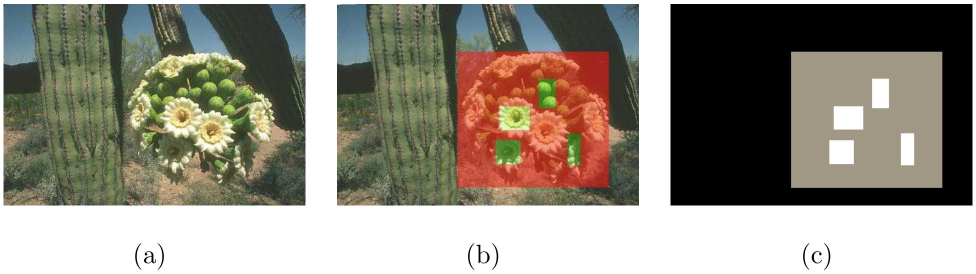
GrabCut [100] is a two step approach. The first step is automatic segmentation that relies on the work of *GraphCut* [**?**], which is based on a powerful optimization technique that can be used in ways similar to Bayes Matting, including trimaps and probabilistic color models. This method can achieve robust segmentation even in a camouflage, when foreground and background color distributions are not well separated. The second step is manual post-editing based on the idea of building a graph where each pixel is a graph node with edges connecting to its 8 neighboring pixels. The edges are weighted to form a max-flow/min-cut problem to compute the segmentation. The user only need to select the region of interest. The manual post-processing tools include a *background brush*, a *foreground brush*, and a *matting brush* to smooth region boundaries or re-edit segmentation error manually. *GrabCut* is quite robust compared to methods mentioned earlier but can only select one object at a time. The method minimizes a global cost function which cannot distinguish between fine local details and noise. Therefore, it fails for highly detailed regions and noisy pictures.

3.4.2 Simple Interactive Object Extraction

Simple Interactive Object Extraction (SIOX) [101] is an method for extracting foreground objects from color images and video with little user interaction. It has been implemented as “foreground selection” tool in the GIMP [102], and as part of the tracer tool in Inkscape [103]. SIOX defines *foreground* as a set of spatially connected pixels that the user is interested. The rest of the image is considered as background.

The input for the SIOX method is a color image in CIELAB space and an initial confidence matrix M_i. A confidence matrix has the same dimensions as the image. Each element of the matrix contains a floating point number that lies in the interval [0, 1] and corresponds to one pixel in the image. A value of 0 means the corresponding image pixel belongs to the background, and a value of 1 means the corresponding image pixel belongs to the foreground. Any value between 0 and 1 describes a certain tendency that the corresponding pixel belongs to either foreground or background, with 0.5 indicating no preference. In the following, confidence values of 1 mean known foreground, values of 0 known background, and values of 0.5 unknown. This notion of a confidence matrix has a few advantages. The confidence matrix along with the original picture can easily be passed between different processing steps. Its elements can easily be interpreted as probabilities or as values of a gray-scale image. The latter interpretation allows for standard image operations, such as convolutions or morphological operators, without having to alter the original image. The confidence values can directly be mapped to transparency values. The input confidence matrix for SIOX may contain known foreground, known background, and unknown elements. However, It must contain at least known background. M_i is either specified by the user or generated by an automatic classifier. The method contains the following steps:

1. Color signatures S_B and S_F are created. S_B represents the specified known background and S_F the known foreground (either it has been specified or the signature is calculated as a difference signature between the signature of the entire image and S_B).



(a)

(b)

(c)

Fig. 3.11. (a) Original input image, (b) A user-provided selection (red: region of interest, green: know foreground), and (c) Corresponding confidence matrix (black: known background, gray: unknown, white: known foreground).

2. Each unknown pixel of the image is classified as either foreground or background using a nearest-neighbor search in S_F and S_B. This produces a new confidence matrix M_o.

3. Noise is filtered by using a erode/dilate morphological operation and a blur operation on the matrix M_o to remove artifacts and optionally close holes up to a specific size.

4. Connected components are identified with high confidence in M_o which are large enough or correspond to user markings. Values of all other connected components are set to 0.

5. The confidence matrix M_o is then used. This is usually done by mapping the elements of M_o directly to the transparency values of the pixels contained in the image.

Figure 3.11 shows a sample input for the method and the corresponding confidence matrix.

A color signature is defined as a set of representative colors, not necessarily a subset of the input colors. A color signature is constructed by clustering a set of pixels into equally-size. The centroids of the clusters are defined as the representative colors. A color signature is created from a set of pixels as follows: Given a set of color pixels, all colors are regarded as points in a d-dimensional color space. This color space is subdivided recursively, starting with the whole space. For each step i, the points in the current box B of the subdivision are projected onto the axis a along dimension i mod d. Two extreme projections p, q are determined, and if $\|p - q\|$ is larger than a given threshold $l_{i \quad mod \quad d}$, B is split into two with a plane orthogonal to a at $(p+q)/2$. This is repeated until all boxes have at least one dimension that is smaller than the threshold for that dimension. Through experiments, the triple (0.64, 1.28, 2.56) for the box width in dimension i was found to be a good set of threshold values using genetic methods.

In a second pass, all center points of the boxes resulting from pass one are taken and are used as input points for the same method. To improve noise robustness, only the center points of such boxes B are considered that contain at least t points for a fixed threshold t. These points are representative points and therefore become part of the signature. A good value for t is the number of specified pixels divided by 1000. As observed by [104], this clustering method produces a good distribution and a representative signature with few points.

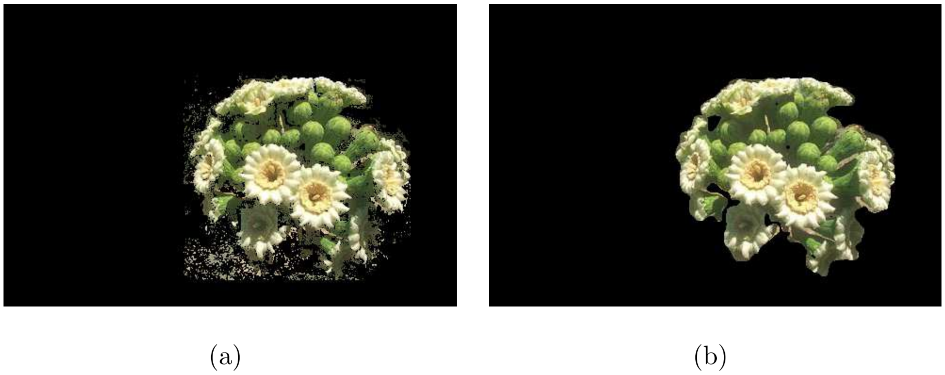
A color signature is built for the set of pixels having confidence 0 and another one is built for pixels of confidence 1. If the confidence matrix does not contain any pixels with confidence 1, the foreground signature is found by *color signature subtraction* which is defined as follows. Two color signatures S_1 and S_2 are subtracted into a resulting signature $R = S_1 \backslash S_2$ by comparing the representative colors contained in S_1 and S_2 using the Euclidean distance. For each element in S_2, the element in S_1 with minimum distance is marked. R is a subset of S_1 that contains only those representative colors of S_1 that have not been marked. S_2 must not contain more elements than S_1. In order to build a foreground signature when only known background is given, the background signature is subtracted from the signature of the entire image.

Pixels with confidence value 0.5 are classified using nearest neighbor search. If the Euclidian distance of a pixels color is closer to an element of the foreground signature than to all elements of the background signature, it is classified as foreground. Otherwise it is classified as background. If a color has equal minimal distances to both signatures, the pixel is considered foreground. The practical reason behind this is that it is usually easier to erase wrongly classified foreground than to reconstruct wrongly classified background in most image editing tools.

Simple foreground/background classification based on the distances to the color signatures will usually select some individual pixels in the background with a foreground color and vice versa, resulting in tiny holes in the foreground object. Again, the incorrectly classified background pixels are eliminated by a standard "erode" morphological operation while tiny holes are filled by a standard "dilate" morphological operation [71] directly done on the confidence matrix. A breadth-first search using the confidence matrix is performed to identify all spatially connected regions that were classified as foreground. Either the biggest region or all regions with an area greater than a threshold are considered as the final foreground object(s). The user can specify a smoothness factor to define how much smoothing should be applied to the confidence matrix. More smoothing reduces small classification errors. Less smoothing is appropriate for high-frequency object boundaries. The values of the confidence matrix are directly used as transparency values (α values) for each corresponding pixel. Figure 3.12 shows a sample result before and after post-processing.

3.4.3 GIMP Tool

For still image object segmentation, the user specifies the known background and known foreground regions manually. In the following, the user-specified regions are called *trimap*. As discussed in the previous section, the known foreground is optional, but it improves the robustness of the segmentation. To provide this information, the user makes several selections with a mouse. The outer region of the first selected area specifies the known background while the inner region defines the unknown region.



(a) (b)

Fig. 3.12. Color classification result. (a) result before post-processing, (b) result after post-processing.

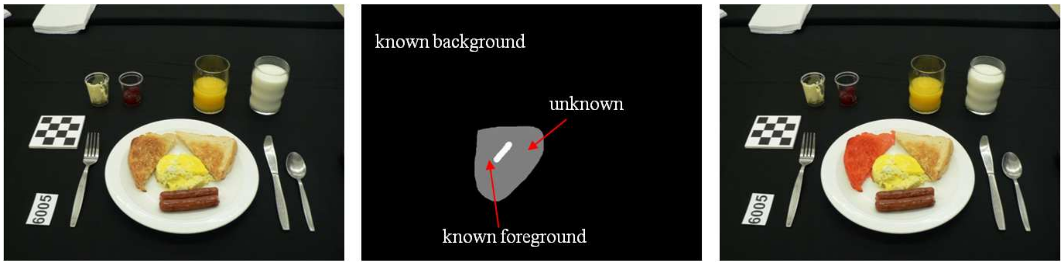
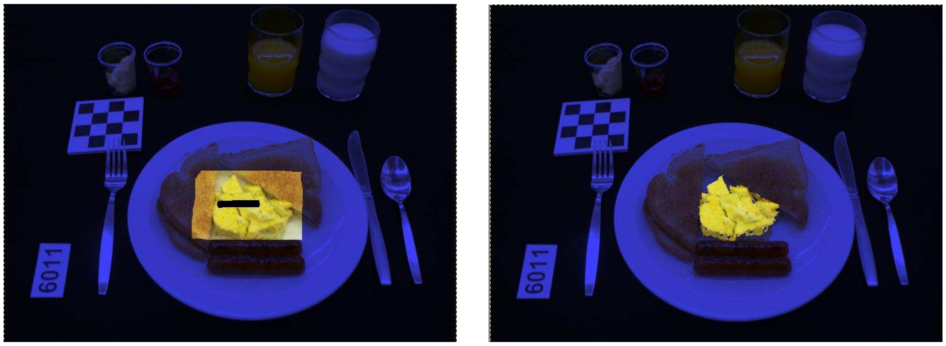


Fig. 3.13. An example of SIOX tool in Gimp used with a food image.

Using additional selections, the user may specify one or more known foreground regions or additional background regions to refine the region of interest. Internally, the *trimap* is mapped into a confidence matrix. Figure 3.13 illustrates an example of the approach used with a food image.

Using this interaction style, SIOX has been integrated into the core of the open-source project GIMP (GNU Image Manipulation Program) [102]. A freehand selection tool is used to specify the region of interest. It contains all foreground objects to be extracted and few background pixels. The pixels outside the region of interest form the known background while the inner region defines the unknown region. The known background is visualized as dark area. The user then uses a foreground brush to mark representative foreground regions. Internally, this input is mapped into a confidence matrix, where each element of the matrix represents a pixel in the input



(a) (b)

Fig. 3.14. (a) Select region of interest, (b) Result from segmentation.

image. The values of the elements lie in the interval [0, 1] where a value of 0 specifies known background, a value of 0.5 specifies unknown, and a value of 1 specifies known foreground. Once the mouse button has been released, the selection is shown to the user. The user can refine the selection by either adding additional foreground markings or by adding background markings using the background brush. The final selection mask is created when the user pressing the “Enter” key. The object can then be manipulated independently. Figure 3.14 shows an example of using the tool to extract an object in a food image.

3.5 Multilevel Segmentation

Assigning predefined class labels to every pixel in an image is a highly unconstrained problem. Human vision system has the remarkable abilities to group pixels of an image into object segments without knowing *a priori* which objects are present in that image. Designing well-behaved models capable of making more informed decisions using increased spatial support is an open problem for segmentation and classification systems. It is necessary to work at different spatial scale on segments that can model either entire objects, or at least sufficiently distinct parts of them. Recent developments in this area have shown promising results. In [105], the authors present a framework for generating and ranking plausible objects hypotheses by solving a sequence of constrained parametric min-cut problems and ranking the object hypotheses based on mid-level properties. A multiple hypothesis framework is proposed in [106] for robust estimation of scene structure from a single image and obtaining confidences for each geometric label. Sivic et. al. [107] use a probability latent semantic analysis model to discover the object categories depicted in a set of unlabeled images. The model is applied to images using vector quantization on SIFT-like region descriptors.

Given a large, unlabeled collection of images, for each pixel in the test images, our goal is to predict the class of the object containing that pixel or declare it as "background" if the pixel does not belong to any of the specified classes. The output is a labeled image with each pixel label indicating the inferred class (object). We exploit the fact that segmentation methods are not stable as one perturbs their parameters, thus obtaining a variety of different segmentations. Many segmentation methods such as Normalized Cuts [92], use the number of segments as one of the input parameters of the segmentation method. Since, the exact number of segments in an image is not known a priori, a particular choice of the number of segments results in either an under segmented or over segmented image. Further, for a particular choice of number of segments, some objects may be under segmented, while others may be over segmented. That is, some of the segments may contain pixels from more than one class while more than one segments may correspond to a single class. The probabilistic classifier used in our system, for classification of these segments, provides the K most probable candidate classes along with their probability estimates. Consequently, probability estimates achieve smaller values for over-segmented or under-segmented classes. To overcome these problems associated with the unknown number of segments in images, we aim to generate a pool of segments for each image to achieve high probability of obtaining "good" segments that may contain potential objects. Since we are not relying on any particular segmentation to be correct, the choice of segmentation method is not critical. Figure 3.15 describes an overview of our approach. Given

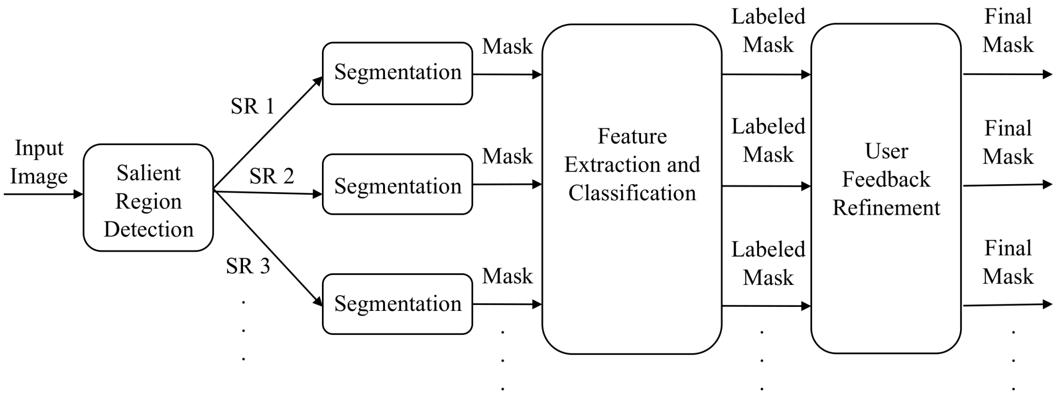


Fig. 3.15. Proposed Segmentation Approach.

an input image, we first use salient region detection to identify potential regions in the image containing objects of our interest. The proposed multilevel segmentation approach is then used with each salient region SR1, SR2, SR3 . . . , this step results in a number of segmentations to be classified based on the selected features. Normalized Cuts is used to generate segments in each of the salient regions as indicated in Figure 3.15. The results from the classifier are then used as feedback to the select the "optimal" parameters for the Normalized Cuts. In Section 3.2.4 we describe the use of region-based active contours model to refine the segmentation based on user feedback information. Here, we focus our discussion on the multilevel segmentation approach without user interaction.

3.5.1 Salient Region Detection

Our proposed segmentation method includes an initial step to identify regions of interest. Unique to our application, we are interested in regions of an image containing food objects. The region of interest detection is useful to our task of assigning correct label to each pixel by rejecting non-food objects such as the tablecloth, utensils, napkins, and thus reducing the number of pixels to be processed.

Knowledge-based methods are used to determine these regions of interest. The first step is to remove the background pixels from our search space. The images from our user studies contain uniformly colored tablecloths, therefore, we can generate a foreground-background image by labeling the most frequently occurring color in the CIE L*a*b* color space as the background pixel color. Another foreground-background image is formed by identifying strong edges present in each RGB channel of the image. In particular, we use the Canny operator to extract the edgesc̃itefind a citation for the Canny operator. Edge pixels are linked together into lists of sequential edge points, one list for each edge contour. These edge lists are transferred back into a 2D image array [108]. We combine background and edge images and remove undesired noise, such as holes, gaps, and bulges with morphological operations. We then label connected components in the binary image. Since food items are generally located in a plate, bowl, or glass that have distinctive shapes, our goal is to detect these objects. We first remove known non-food objects such as the fiducial marker, currently a checkerboard pattern, used as both a geometric reference and color reference. To determine which components contain potential food items, we use the Canny edge filter [109] on each component and obtain the normalized edge histogram. The criteria for identifying components that contain food objects is the uniformity of the edge histogram. We compute the euclidean distance between the normalized edge histogram of each salient region and a uniform distribution. Based on this criteria, a threshold is selected to determine salient region.

3.5.2 Multiscale Segmentation

By using multiscale segmentations, recent studies have achieved promising segmentation results under non-trivial conditions. In [110], the authors use an algebraic multigrid to find an appropriate solution to the normalized cut measures and describe a process of recursive coarsening to produce an irregular pyramid encoding region based on grouping cues. Another method, proposed in [111] constructs multiscale edges defining pairwise pixel affinity at multiple grids. Simultaneous segmentation through all graph levels is evaluated based on the average cuts criterion.

We adopted the approach proposed in [112], where multiple scales of the image are processed in parallel without iteration to capture both coarse and fine level details. The approach uses the Normalized Cut [92] graph partitioning framework. In the Normalized Cuts framework, segmentation quality depends on the pairwise pixel affinity graph. A larger graph radius generally makes the segmentation better. However, the advantage of graphs with long connections comes with a great computational cost. If implemented naively, segmentation on a fully connect graph G of size N would require at least $O(N^2)$ operations. Therefore, the ideal graph connection radius is a trade off between the computation cost and segmentation result.

In Normalized Cuts, an image is modeled as a weighted, undirected graph. Each pixel is a node in the graph with an edge formed between every pair of pixels. The weight of an edge is a measure of the similarity between the two pixels, denoted as $W_I(i, j)$. The image is partitioned into disjoint sets by removing the edges connecting the segments. The optimal partitioning of the graph is the one that minimizes the weights of the edges that were removed (the cut). The method in [92] seeks to minimize the Normalized Cut, which is the ratio of the cut to all of the edges in the set. Two simple, yet effective, local group cues are used to encode the pairwise pixel affinity graph. Since close-by pixels with similar intensity value are likely to belong to the same object, we can represent such affinity by:

$$W_I\left(i,j\right) = \exp\left[-\left(\frac{\left\|I_i - I_j\right\|_2^2}{\sigma_I^2} + \frac{\left\|X_i - X_j\right\|_2^2}{\sigma_X^2}\right)\right]. \tag{3.44}$$

where I_i and X_i denote pixel intensity and location. Image edges are also strong indicator of potential object boundary. The affinity between two pixels can be measured by the magnitude of image edges between them,

$$W_C\left(i,j\right) = \exp\frac{-\max_{x\in line(i,j)}\left\|Edge\left(x\right)\right\|^2}{\sigma_C^2} \tag{3.45}$$

where $line(i,j)$ is the line joining pixel i and j, and $Edge(x)$ is the edge strength at location x. We can combine these two grouping cues with the tuning parameter α by

$$W_{comb}\left(i,j\right) = \sqrt{W_I\left(i,j\right) \times W_C\left(i,j\right)} + \alpha W_C\left(i,j\right). \tag{3.46}$$

The graph affinity $W(i,j)$ exhibits very different characteristics at different ranges of spatial separation. Therefore, we can separate the graph links into scales according to their underlying spatial separation,

$$W_{full} = W_1 + W_2 \approx W_1 + {C_{1,2}}^T W_2 C_{1,2} = W_{reconstruction}, \tag{3.47}$$

where W_i contains affinity between pixels with certain spatial separation range and can be compressed using a recursive sub-sampling of the image pixels such as the use of interpolation matrix $C_{1,2}$ between two scales. This decomposition allows one to study behaviors of graph affinities at different spatial separations. The small number of short-range and long-range connections can have virtually the same effect as a large fully connected graph. This method is able to compress a large fully connected graph into a multiscale graph with $O(N)$ total graph weights. The combined grouping cues are used with the CIE L*a*b* color space. Selections of the Normalized Cut parameters to generate multiple segmentation hypothesis are discussed in Section 3.5.5.

3.5.3 Fast Rejection

Having a large pool of segments makes our overall methods more reliable, however many segments are redundant and not good. These segments are results of selecting inappropriate clustering number in the segmentation step reflecting accidental image grouping. We deal with these problems using a fast rejection step. We first filter out small segments (up to 500 pixels in area) in our implementation as these segments do not contain significant feature points to represent the object classes. We then assign label to segments in each salient region which belong to background as detected previously. The number of segments that passes the fast rejection step is indicative of how rich or cluttered a salient region is.

3.5.4 Feature Extraction and Classification

This section briefly describes the features used in our experiments. We have proposed a framework that combines global and local features with late decision fusion [42]. Global features are the features that incorporate statistics of the overall distribution of visual information in the object. For the global features we considered three types of color features namely *color statistics, entropy statistics, and predominant color statistics* and three types of texture descriptors namely *Entropy Categorization and Fractal Dimension estimation (EFD), Gabor-Based Image Decomposition and Fractal Dimension Estimation (GFD), and Gradient Orientation Spatial-Dependence Matrix (GOSDM).* Finally, four types of local feature are also used. Extracting local features consists of describing visual information from a neighborhood around points of interest in the segment. The local features used in this thesis are: *SIFT, Haar wavelets, Steerable filters, and color statistics.*

Color features: *Color statistics* includes the 1^{st} and 2^{nd} order moment estimates of the *R, G, B, Cb, Cr, a, b, H, S, V* channels for the entire segment. For *Entropy statistics* each segment is first divided into smaller blocks ($N \times N$ pixels) and then 1^{st} and 2^{nd} order moment statistics of the entropy in the *R, G, B* channels are estimated for each block. The average values for all the blocks is used as the final entropy features. *Predominant color statistics* describes the distribution of four most representative colors (in RGB space) for an object [113].

Texture features: *EFD* is an extension of multifractal analysis framework [114, 115]. We select the entropy of the image as a measure to define a point categorization. The Fractal dimension is, then, estimated for every point set according to this categorization. This approach attempts to characterize the variation of roughness of homogenous parts of the texture in terms of complexity. The Fractal dimension is estimated using the Box-counting method [116]. *GFD* is another variant of multifractal theory, in this case the image is decomposed into primitives in its spatial frequency dimension. For each filtered response the fractal dimension is estimated. Finally, our third texture descriptor, *GOSDM* consists of a set of gradient orientation spatial-dependence matrices to describe textures by determining the probability of occurrence of quantized gradient orientations at a given spatial offsets.

As previously stated, we also use local features to describe visual information from smaller portions of the segment/object. These include the *SIFT* descriptor introduced by Lowe in 2004 [117], *Haar wavelets*, which capture the distribution of gradients within the neighborhood around the point of interest, and *Steerable filters* which refer to randomly oriented filters synthesized using a linear combination of the basis filters [118]. The feature vectors consists of 1st and 2nd order moment statistics of the response of the filtered patch. Finally, *local color statistics* features are also considered. The 1st and 2nd of the *R, G, B, Cb, Cr, a, b, H, S, V* channels around each point of interest are estimated to capture local color information. The Differential-of-Gaussians approach (DoG) [117] is used for detecting the points of interest.

The above features are independently classified for each of the 12 feature channels ($l = 1, \cdots, 12$). Global features are classified using Support Vector Machines(SVM) [119]. Radial Basis Function are used as kernels of our SVM implementation. Local features are represented by the frequency histogram of visual words obtained by assigning each descriptor of the segment to the closest visual word. Visual words are formed from the training set by using hierarchical k-means clustering. For each of the four local features, the signature of the segment is defined as follows:

$$\phi_j = \{(t_1, m_1), ..., (t_i, m_i), .., (t_N, m_N)\} \tag{3.48}$$

where ϕ_j represents the signature of the object for the jth local feature channel, t_i represents the frequency term and m_i is the *medoid* of the ith cluster. N is the number of visual words. These signatures are applied to a nearest neighbor search method to select the final class for each channel.

Both classification methods used, namely SVM for global features and nearest neighbors for local features, output K ($K = 4$ for these experiments) candidate decision categories for each segment (S_q), and each feature channel l, ($c_k^{(l)}(S_q)$). The final decision, $C_k(S_q)$, is made by applying a majority vote rule on $c_k(S_q) = [c_k^{(1)}(S_q), \cdots, c_k^{(L)}(S_q)]$. This means that the top K categories are selected as final candidates for segment S_q.

In our framework, not only the classifier's decision is used as "side" information for segmentation, but also the confidence score from the classifier. Independent of which classification method is used for each individual feature, the confidence score is estimated in the same fashion. The confidence score describes the classifier's confidence that its inferred label is correct. The confidence score $\psi_l(S_q, c)$, for assigning segment S_q to class c in the feature channel l is defined as:

$$\psi_l(S_q, c) = \frac{1}{T}\sum_{i=1}^{T} exp(-d(S_q, S_c^i)), \text{for each } c \in C_K(S_q), \tag{3.49}$$

where $d(S_q, S_c^i)$ represents the distance between normalized feature vector of the query segment S_q and the normalized feature vector of the i^{th} nearest neighbor training segment belonging to class c. T is set to 5 in our experiments. The final confidence score for all feature channels of each candidate class c is defined as:

$$\Psi(S_q, c) = \frac{1}{L}\sum_{i=1}^{L} \psi_l(S_q, c), \tag{3.50}$$

where $L = 12$ is the total number of feature channels, and $\psi_l(.,.)$ represents the confidence score per feature channel, and $\Psi(.,.)$ the final confidence score of the classifier to label segment S_q with label c.

As a result of this, each pixel in the segment is mapped to four confidence scores corresponding to the four candidate categories predicted by the classifier. The next section describes the approach followed to achieve segmentation stability and robustness by using the confidence scores.

3.5.5 Optimal Segmentation

Each segmentation hypotheses vary in the number of segments and class labels, thus making errors in different regions of the image. Our challenge is to determine which parts of the hypotheses are likely to be correct and combine the hypotheses to accurately locate the objects and determine their class labels. Since the "correct" number of segments Q that yield an "optimal" segmentation is unknown *a priori*, we would like to explore all possible parameter settings. We are still left with defining the optimal segmentation. In [93], the authors built upon stability-based approaches to develop methods for automatic model order selection. The approach includes the ability to detect multiple stable clusterings instead of only one and a simple means of estimating stability that does not require training a classifier.

We propose an iterative stability framework for joint segmentation and classification. To produce multiple segmentations, we vary the number of segments Q in two ways, depending on the size of the salient region. For our image database, $Q = 3$ is used as the initial number of segments for regions less than 250-pixels in length or breadth of a bounding box, and $Q = 7$ for larger regions. Figure 3.16 shows multiple segmentations for each salient region of the input image shown in Figure 2.1. Let $S^m_{(i,j)}$ denote the segment corresponding to the pixel $I(i,j)$, for the m^{th} iteration of segmentation and classification steps. $C_K(S^m_{(i,j)})$ denotes the set of K candidate class labels for segment $S^m_{(i,j)}$. The set of K candidate class labels for pixel $I(i,j)$, after M iterations is denoted by $C^M_K(I(i,j))$. Each of the candidate classes $c^M_k(I(i,j))$ is estimated based on the cumulative confidence scores $\Psi^M(I(i,j),c)$ defined in Equation 3.51.

$$c^M_k(I(i,j)) = \underbrace{argmax}_{c \notin C^M_{(k-1)}} \Psi^M(I(i,j),c), \text{ where,}$$

$$\Psi^M(I(i,j),c) = \frac{\sum_{m=1}^{M} \sum_{c_i \in C_K(S^m_{(i,j)})} 1_{(c_i=c)} \Psi(S^m_{(i,j)},c)}{\sum_{c_i \in C_K(S^m_{(i,j)})} 1_{(c_i=c)}} \tag{3.51}$$

In our experiments, we accumulate confidence scores for the top four candidate class labels. In each iteration, every pixel is assigned with the class label that has the highest cumulative confidence scores up till the current iteration. The pixel label becomes stable when the cumulative confidence scores does not show more improvement. The iteration process stops when the percentage of pixels labels being updated is less than 5% for each segment. In general, we achieve the “optimal” results after four iterations. The output is a labeled map with each pixel assigned to the best class label. The iterative stability measure depends on the classifier’s confidence of the assigned label for each segment being correct, thus the performance of classification plays an important role. The correct class label of the segment requires accurate detection of the object boundary so that features extracted from the segment can be closely matched to features of objects in the training images. Therefore, a high confidence score of the classifier not only implies strong visual similarity between the identified object and its training data, but also accurate boundary detection from the segmentation. It is unlikely that the classifier will have 100% accuracy even with perfect segmentation because some foods are inherently difficult to classify due to their similarity in the feature space. Examples of these are illustrated in Section 4.2.6.

3.6 Evaluation Methods

In previous sections, we have examined various bottom-up image segmentation methods. Researchers in this field have frequently point out the need for standard quality measures that would allow both evaluation and comparison of available segmentation procedures. According to [120], evaluation methods can be broadly divided into two categories: **analytical methods** and **empirical methods**. “The analytical methods directly examine and assess the segmentation methods themselves by analyzing their principles and properties. The empirical methods indirectly judge the

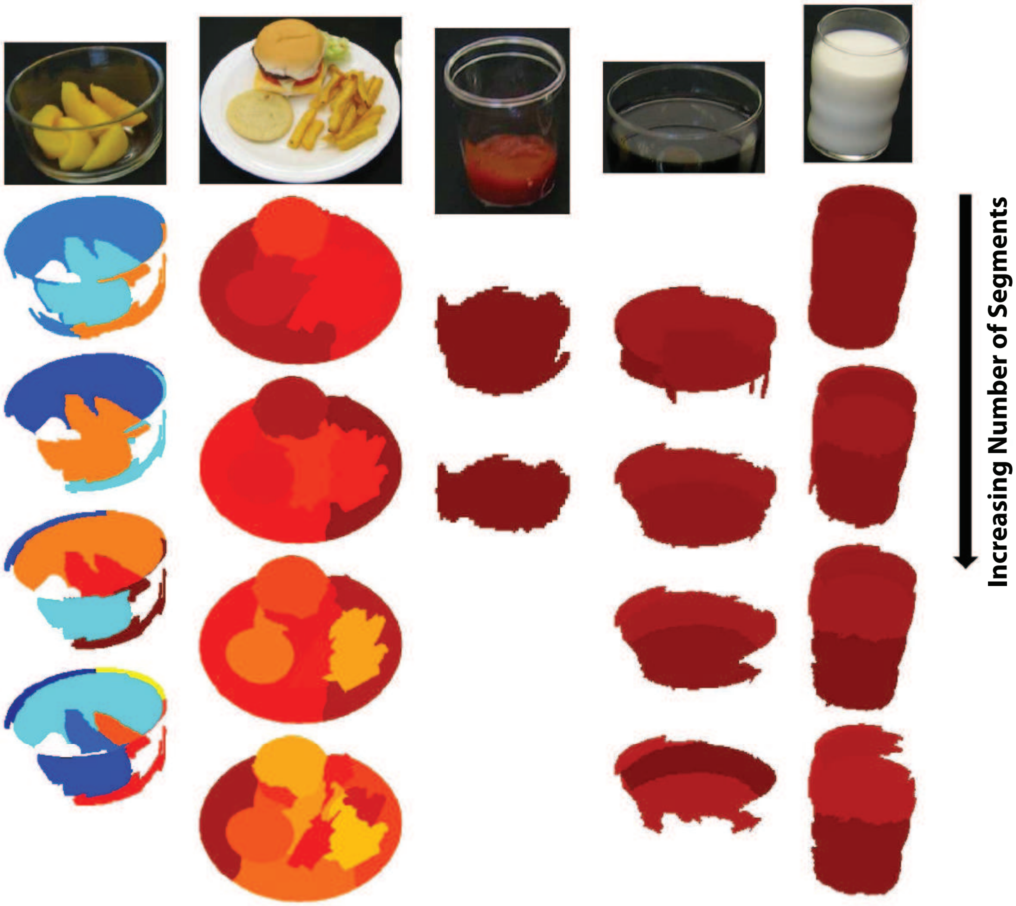


Fig. 3.16. Multiple Segmentation Results of Salient Regions.

segmentation method by applying them to test images and measuring the quality of segmentation results."

Although using analytical techniques for the evaluation of segmentation methods avoids implementation of these methods, they have not received much attention due to the difficulty to compare methods based solely on analytical studies. Empirical methods can be further classified into **goodness methods** and **discrepancy methods**. In the empirical goodness methods, some desirable properties of segmented images, often based on human intuition about what conditions should be satisfied by an "ideal segmentation," are measured by goodness parameters. There methods rate different segmentation techniques by computing some chosen goodness measure without requiring *a priori* knowledge of the reference segmentation. Different types of goodness measures have been proposed such as color uniformity, entropy, intraregion uniformity, inter-region contract, and region shape.

Empirical discrepancy methods are based on the availability of a **reference segmentation**, also known as **ground-truth**. The disparity between an actually segmented image and the ideal segmented image (ground-truth) can be used to assess the method performance. Both images are obtained from the same input image. Methods in this group take the difference, measured by various discrepancy parameters, between the actually segmented image and the reference image into account. Our evaluation methods fall into this category. In particular, we are interested in evaluating our proposed segmentation methods based on quantitative measures of segmentation quality as well as their performance in the context of object classification task.

3.6.1 Quantitative Evaluation

Martin et al. [121] proposed a set of measures - GCE, LCE, and BCE - to compute the overall distance between two segmentations as the sum of the local inconsistency at each pixel. The measures were designed to be tolerant to refinement. In another word, the consistency error should be low if subsets of regions in one segmentation consistently merge into some region in the other segmentation. To compute the consistency error for a pair of images, they first defined a measure of the error at each pixel p_i

$$E\left(S_1, S_2, p_i\right) = \frac{\left|R\left(S_1, p_i\right) \backslash R\left(S_2, p_i\right)\right|}{\left|R\left(S_1, p_i\right)\right|} \tag{3.52}$$

where $R(S_j, p_i)$ was the region in segmentation j that contained pixel P_i, $\backslash$ denoted set difference, and $|\cdot|$ denoted set cardinality. This measure is equal to 0 if all the pixels in S_1 were contained in S_2, therefore achieving the tolerance to refinement. Because this measure was non-symmetric, every pixel must be accounted for in each direction, resulting twice computation.

Given the error measures at each pixel, two segmentation error measures were defined

$$GCE\left(S_1, S_2\right) = \frac{1}{n} \min \left(\sum_i E\left(S_1, S_2, p_i\right), \sum_i E\left(S_2, S_1, p_i\right) \right) \tag{3.53}$$

and

$$LCE\left(S_1, S_2\right) = \frac{1}{n} \sum_i \min\left(E\left(S_1, S_2, p_i\right), E\left(S_2, S_1, p_i\right)\right) \tag{3.54}$$

The Global Consistency Error (GCE) assumed that one of the segmentations must be a refinement of the other, and forces all local refinements to be in the same direction. The Local Consistency Error (LCE) allowed for refinements to occur in either direction at different locations in the segmentation.

Martin et al. showed that when paris of human segmentations of the same images were compared, both the GCE and LCE were low; conversely, when random pairs of human segmentations were compared, the resulting GCE and LCE are high. Histograms of GCE and LCE were computed over the complete test image set from the Berkeley Segmentation Database [122], both for pairs of human segmentation of the same image, and for random pairs of human segmentations, as the baseline to evaluate the quality of the segmentation produced by Normalized Cuts.

It is interesting to examine the two metrics reported on two extreme cases: a completely under-segmented image where the segmentation contains only one region containing the entire image, and a completely over-segmented image in which every pixel was assigned a different label. From the definition of GCE and LCE, it can be seen that both measures result as 0 on both of these extreme cases regardless of the types of segmentation they are compared to. This is due to the tolerance of these measures of refinement. Any segmentation is a refinement of a completely under-segmented image, and a completely over-segmented image is a refinement of any other segmentations.

The authors noted these problems and proposed alternative ways to evaluate paris of segmentations. Three of the proposed metrics were region based and one was a variation of the LCE metric, namely Bidirectional Consistency Error

$$BCE\left(S_1, S_2\right) = \frac{1}{n}\sum_i \max\left(E\left(S_1, S_2, p_i\right), E\left(S_2, S_1, p_i\right)\right) \tag{3.55}$$

which no longer tolerated to refinement. The other two metrics were based on mutual information, one used the affinity matrix itself, and the other compared the pairwise assignment of pixels to regions in the two segmentations.

Martin also proposed the use of precision/recall curves based on the region boundaries to evaluate segmentation consistency. Recall is defined as the proportion of boundary pixels in the ground-truth that were successfully detected by the automatic segmentation; while precision is defined as the proportion of boundary pixels in the automatic segmentation that correspond to boundary pixels in the ground-truth. Precision and recall are sensitive to over and under-segmentation, therefore are proper measures of segmentation quality. Low precision scores means over-segmentation, and low recall scores are lead by under-segmentation. High values in both precision and recall scores can only be achieved when the boundaries in both segmentations agree in spatial locations and level of details.

Precision and recall were estimated in [123] by computing a minimum cost matching between boundary pixels located in two segmentations. The cost of matching two pixels b_i and b_x was proportional to their similarity in spatial location and orientation:

$$w_{b_i \to b_x} = \sqrt{\left(\Delta x\right)^2 + \left(\Delta y\right)^2} + \alpha\frac{|\Delta\theta|}{\pi/2} \tag{3.56}$$

where $\sqrt{\left(\Delta x\right)^2 + \left(\Delta y\right)^2}$ specifies the Euclidean distance between two pixels, α is the scaling parameter for the orientation term, and the orientation difference $|\Delta\theta|$ is limited to the range $[0, \pi/2]$. The matching cost for each pixel was computed using bipartite matching method, implemented using Andrew Goldberg's CSA package [124]. Due to different number of boundary pixels in the two segmentations, the number of outlier nodes is considered such that the total number of nodes is the same for both segmentations. The cost of matching was set to be high for a boundary pixel in either segmentation to an outlier node. Therefore, a boundary pixel would only match to an outlier node if no suitable match was found within the boundary pixels in the other segmentation.

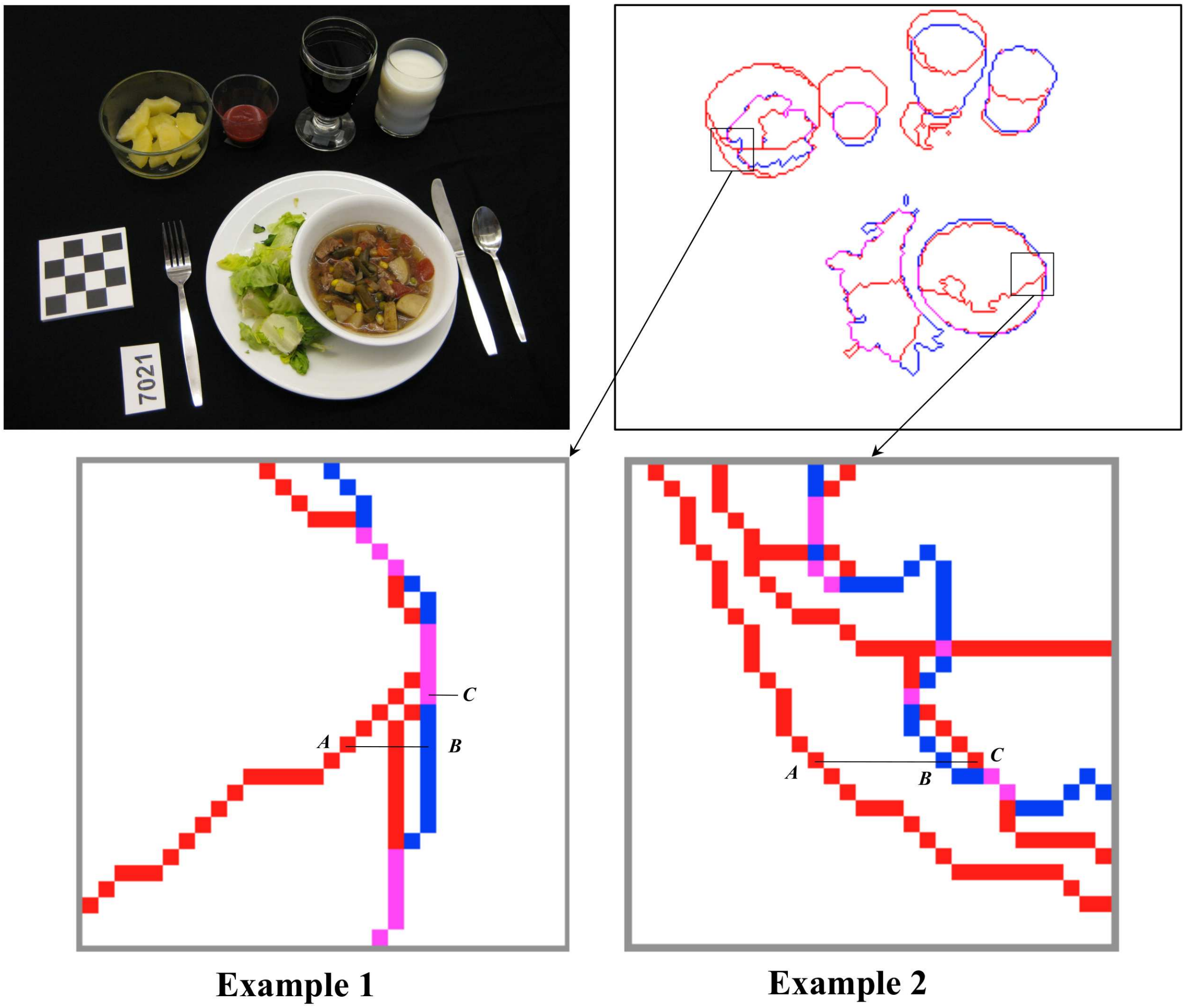
Estrada et al. [125] proposed a different matching strategy. Given two segmentations S_1 and S_2, a suitable match for each boundary pixel in S_1 was found by examining its neighborhood within a radius of d for boundary pixels in S_2. A pixel b_i in S_1 was matched to a pixel b_x in S_2 when the following conditions were satisfied,

- No other boundary pixel b_j in S_1 exists between b_i and b_x (no intervening contours constraints).

- The closest boundary pixel in S_1 for pixel b_x is in the specified direction of b_i. If b_x has several closest neighbors, at least one must point in the specified direction of b_i. In practical implementation, this means the directions from b_x to b_i, and from b_x to one of its closest neighbors must be within 25 degrees of each other (same side constraint).

In the case where more than one pixel in S_2 satisfies the listed conditions for pixel b_i, we select the nearest one. These two conditions imply that unless b_i is part of the closest boundary in S_1 to b_x's boundary in S_2, b_i should not be matched to a boundary pixel b_x . It is necessary to enforce the direction condition because double matches are possible when a boundary in S_2 is next to both sides by boundaries in S_1.

Figure 3.17 illustrates the two matching conditions. Blue pixels indicate boundary from human segmentation, red pixels show boundary from automatic segmentation, and purple pixels indicate overlap (exact match). **Example 1** shows an instance of the first matching condition. Pixel **A** in S_1 cannot be matched to pixel **B** in S_2 because there are other boundary pixels (*red*) between them. This condition prevents any other boundary pixels in S_1 from matching to the same boundary pixels in S_2 when overlap pixels such **C** exist. **Example 2** shows an application of the second matching condition. In this case, pixel **A** cannot be matched to boundary pixel **B** because **A** is not on the same side as boundary pixel C, which is the pixel closest to the boundary pixel **B** in S_2.



Example 1

Example 2

Fig. 3.17. An Illustration of the Matching Conditions. The top row shows an image from our food image database, and the overlay of the ground-truth segmentation (*blue*) and automatic segmentation (*red*) (*purple* indicates overlap pixels). **Example 1** illustrates the first matching condition, which avoids matching across multiple boundaries. **Example 2** illustrates the second matching condition, which ensures a band of matches pixels in S_1 all of which are within distance d of a single boundary in S_2.

Precision can be defined as

$$\text{Precision}\left(S_1, S_2\right) = \frac{\left|matched\left(S_1\right)\right|}{\left|S_1\right|} \tag{3.57}$$

where $matched(S_1)$ is the set of boundary pixels in S_1 that have a suitable match in S_2 within a distance d, and $|\cdot|$ represents set cardinality. Similarly, recall can be defined as

$$\text{Recall}\left(S_2, S_1\right) = \frac{\left|matched\left(S_2\right)\right|}{\left|S_2\right|} \tag{3.58}$$

Given the matching method described above, precision and recall can be estimated for any pair of segmentations. In particular, performance of any segmentation method can be evaluated by comparing its output segmentations against the human segmentations (ground-truth), and compute precision/recall statistics over a set of test images. If multiple human segmentations of the same image is available, automatic segmentation can compare against the union of the boundaries from all human segmentations of the same image. This was suggested in [123] to compare the level of detail between different human segmentations.

Another advantage of using precision/recall curves its ability to characterize the performance of a segmentation method over a range of input parameters. This allows selection of the optimal parameters for any desired recall/precision value, as well as compare different methods equally. We use the precision/recall curves to evaluate the performance of our proposed segmentation methods on the food image database collected from nutritional studies conducted by the Department of Foods and Nutrition at Purdue University.

3.6.2 Classification Based Evaluation

Researchers such as Borra and Sarkar [126] argued the importance of evaluating segmentation or grouping performance in the context of a particular task. This translated to measuring how much a particular method contributes to the success of higher-level procedure such as object recognition.

Our proposed multilevel segmentation method described in Section 3.5 provides the benchmark for classification based segmentation evaluation. In particular, the performance of the proposed segmentation and classification method is evaluated by the confusion matrix for a set of food classes, averaged over multiple instances of training and testing datasets. This is equivalent to the use of an intersection/union metric, defined as the number of correctly labeled pixels of that class, divided by the number of pixels labeled with that class in either the ground truth labeling or the inferred labeling. Equivalently, the accuracy is given by

$$\text{seg. accuracy} = \frac{\text{true pos.}}{\text{true pos.} + \text{false pos.} + \text{false neg.}} \tag{3.59}$$

4. EXPERIMENTAL RESULTS

In this chapter, we present results of testing our proposed methods for different cases and datasets. The evaluation of these methods are different for color correction, food classification and image segmentation. The results of the proposed methods are evaluated by human observers where ground-truth data is available such as food classification and image segmentation. For color correction, perceptual quality is highly subjective and can vary across observers. Therefore, we adopt quantitative approach to evaluate color correction accuracy by estimating statistic errors. We also use quantitative evaluation for image segmentation using precision/recall scores.

4.1 Image Analysis Results

4.1.1 Color Correction

To test our proposed color correction methods described in Section 2.1, images containing the color fiducial marker and sample food items were taken under various illumination conditions generated by the Macbeth SpetraLight II. Two different sizes of color fiducial markers were tested. The 4×5 marker was used in our nutritional studies due to its smaller size for convenient carrying. The other marker of size 7×8 is larger in size and incorporates more colors. We show both visual quality and pixel value differences between reference images and corrected images. Two methods were considered, namely our 3D LUT describe previously method and Choi's method [49]. Our proposed color correction method uses an image of the color fiducial marker captured under a known illumination as the reference colors. We then construct a transformation based on the reference illumination to correct colors of an image taken under unknown illumination. This method is realized use a 3D LUT. The

Choi's method defines a conversion vector base on the Macbeth color board, that can be applied to map colors from a source illumination to a target illumination.

Figure 4.1 shows examples of using the color fiducial marker to correct colors in sample images taken under unknown illumination to match colors in the reference image using the 3D look-up table (LUT). Visual comparison shows strong similarity between the reference image and color corrected images. Figure 4.2 shows comparison of using a 4 × 5 color fiducial marker to correct colors in sample images taken under unknown illumination to match colors in the reference image using 3D LUT interpolation and Choi's method [49]. Figure 4.3 shows comparison of using a 7 × 8 color fiducial marker to correct colors in sample images taken under unknown illumination to match colors in the reference image using 3D LUT interpolation and Choi's method [49]. A visual comparison between the color corrected images using the 3D LUT interpolation and Choi's method for both sizes of the fiducial marker indicates closer match to the reference image using the 3D LUT interpolation.

We also calculate the statistics of pixel value differences between the reference images and corrected images for both methods to verify our visual comparison. We observe from Table 4.1 that the 3D LUT performed favorably over Choi's method due to smaller error when compared to reference images. By incorporating more color patches on the fiducial marker, the error in pixel value differences was also reduced. Since Choi's method uses a global shift to pixel values in the target image, pixel value difference between spatial neighbors were preserved. Therefore, the color corrected images seem more nature. However, illumination change from the reference image to the target image is often not uniform, a single conversion vector may not be sufficient to accurately correct colors of different regions in the image. The 3D LUT interpolation method operates on each pixel in the target image to create the closest match to the pixel colors in the reference image. As a result, this method produced more accurate color correction in the target image.

Table 4.1
Testing error statistics of LUT and Choi's methods.

Type\Statistic	Mean	Median	1-Std. Dev.
4 × 5 color checkerboard (LUT)	18.2 (31.1%)	19.3	16.5
4 × 5 color checkerboard (Choi)	26.4	22.5	18.8
7 × 8 color checkerboard (LUT)	13.3 (43.16%)	10.1	12.4
7 × 8 color checkerboard (Choi)	23.4	17.3	13.9

4.1.2 Feature Extraction and Classification

Given a segmented image, each segment needs to be classified into candidate food object. As we discussed in Section 2.3, we extract color and texture features and use SVM for classification. These features include 1$^{\mathrm{st}}$ and 2$^{\mathrm{nd}}$ moment statistics of several color components *R, G, B, Cb, Cr, a, b, H, S, V*, and Gabor filters to measure local texture properties. We considered 19 food items from 3 different meal events (a total of 63 images) from one diet studies conducted at Purdue University. All images were acquired in the same room with the same lighting conditions. For each of the 19 categories we considered 50% of available images for training and 50% for testing. For the training set, we extracted features from the ground-truth segmentation. For the testing, we extracted features from automatic segmented images. There were a total of approximately 500 food segments that needed to be categorized.

We repeated the experiments ten times randomizing the training and testing sets for each experiment. In Table 4.2 we show the average correct classification for each food category and its corresponding top two misclassified categories. On average the correct classification achieved for all the categories is 56.2%. The non-food segments had a strong influence on the misclassification accuracy since we did not train the classifier with any specific non-food object due to the large variation within this class. Figure 4.4 shows examples of non-food items. We have addressed this issue in recent work done by Marc Bosch [40–42] as well as incorporating more robust features into our classifier. A model of semantic context is also incorporated into the classifier to correct misidentified food items based on likelihood of food combinations.

Some foods are inherently difficult to classify due to their similarity in the feature space we use. For the set of features used here, color is the dominant discriminate feature for many food objects due to their lack of texture or similarity in extracted texture feature. Examples of such errors are shown in the top misclassified categories in Table 4.2. We have also observed from our experiments that the performance of image segmentation plays a crucial role in achieving correct classification results. High classification rate can be achieved when using the ground-truth segmentation. We can achieve up to 95.5% [1] when using ground-truth segmentation. Since color is the dominant feature here, the average value of pixel intensity in the selected color components may be very different than the training data when the segment to be classified contain parts of other objects. To address this problem, we need to improve segmentation accuracy as well as to examine more robust features.

4.2 Image Segmentation Results

Several controlled diet studies were conducted by the Department of Foods and Nutrition at Purdue University whereby participants were asked to take pictures of their food before and after meals [23]. These meal images were used for our experiments. Currently, we have collected more than 8000 food images. To assess the accuracy of our various methods, it is important to develop ground-truth data for the images. For each image, we manually extracted each food item in the scene using a Cintiq Interactive Pen LCD Display and Adobe Photoshop. Given a meal image, we traced the contour of each food item and generated corresponding mask

Table 4.2
Classification Accuracy for Each Food Category and the Top 3 Misclassified Categories.

Food Category	Correct Rates	Top 1 Errors	Top 2 Errors
Catalina Dressing	45%	33% (Ketchup)	14% (Strawberry Jam)
Chocolate Cake	55%	40% (Coke)	5% (Fries)
Coke	59%	37% (Chocolate Cake)	4% (Hamburger)
Eggs (Scrambled)	59%	22% (Fries)	11% (Sugar Cookie)
Fries	58%	35% (Eggs)	7% (Sugar Cookie)
Garlic Bread	51%	36% (Toast)	7% (Sugar Cookie)
Hamburger	71%	21% (Garlic Bread)	8% (Fries)
Ketchup	67%	33% (Catalina)	-
Lettuce	71%	17% (Margarine)	12% (Pear)
Margarine	44%	34% (Milk)	13% (Sugar Cookie)
Milk	47%	25% (Margarine)	15% (Pear)
Orange Juice	61%	39% (Peach)	-
Peach (Canned)	68%	32% (Orange Juice)	-
Pear (Canned)	53%	24% (Margarine)	16% (Milk)
Sausage	67%	23% (Eggs)	10% (Ketchup)
Spaghetti	41%	23% (Fries)	10% (Garlic Bread)
Strawberry Jam	53%	29% (Ketchup)	12% (Catalina)
Sugar Cookie	62%	31% (Eggs)	7% (Margarine)
Toast	42%	37% (Garlic Bread)	13% (Fries)
Average	56.2%	-	-

images along with the correct food labels. As a control, different individuals were asked to ground-truth the images and the results were shown to graduate students in the Department of Foods and Nutrition at Purdue University for evaluation.

For visual comparison of outputs from different segmentation methods discussed in Chapter 3, we choose images collected from nutrient studies consisting of various meal occasions such as breakfast, lunch, dinner and snacks. We show results from several segmentation methods we examined earlier in our research including connected component labeling, normalized cuts and semi-automatic methods, as well as methods used in the our current segmentation approach shown in Figure 3.15 which includes salient region detection, multilevel segmentation and active contours for segmentation refinement.

4.2.1 Connected Component Labeling

Figure 4.5 show sample segmentation results from connected component labeling. The result is shown for images of food replicas (plastic food). We also used this method for some simple food images taken during the nutrient study, these results are shown in Figure 4.6. These images were taken under specific conditions including the use of a white plate and a check-board patterned tablecloth. The tablecloth was used as the fiducial marker for estimating the dimension and area of the food items. The white plate was used to assist the segmentation of the food items. This method works well for images taken under this specific condition, where assumption of the plate and tablecloth can be made based prior knowledge. In addition, food item must be homogeneous in color for this method to successfully capture the entire food object. These constraints places limitation of the performance of this method for use on a wide range of images.

We do not deploy connected component labeling in our current segmentation approach shown in Figure 3.15.

4.2.2 Active Contours

Figure 4.7 illustrates segmentation results generated by our region-based active contours method. The method uses the Chan-Vese [86] model to solve the level set evolution equation for color images using the RGB color components as discussed in Section 3.2. We use a simple rectangle as the initial contour, other geometric shape used as the initial contour also show similar results. Constrained by the need for an initial contour, this model works best for semi-automatic segmentation. One useful scenario in the dietary assessment application is when a user provides feedback on the automatic image analysis results. The user feedback refinement block in Figure 3.1 is such a scenario. In particular, when the image analysis fails to locate certain food items, the user can use a pen tool on mpFR to provide a rough contour of the food item. Given such initialization, the active contour model can then be used on the image to locate additional food items. Figure 4.8 shows two examples where one or more food items did not produce correct segmentation from the automatic image analysis. Using the pen tool on the mpFR, the user provided some initial points where these items were located. Initial contours were generated based on these user supplied points for curve evolution using our active contour model. Curves were successfully attached to the object boundaries through evolution to generate the final segmentations.

4.2.3 Normalized Cuts

We tested the performance of Normalized Cuts over a range of input parameters for a set of images collected from the 24-hr nutrient study including breakfast, lunch and dinner meals (see Section 3.3). For these images, we manually cropped the image to select regions of an image containing food items. We then used the Normalized Cuts method on grayscale versions of each cropped image. Figure 4.9, 4.10, 4.11 show sample results from this set of images. For these results, the optimal input parameters were selected from experiments based on perceptual quality comparison. Since Normalized Cuts takes only one input parameter which is the number of desired segment in an image, typical value range from 3 to 15 depending on the complexity of the image to be segmented.

4.2.4 SIOX

The Simple Interactive Object Extraction (SIOX) (see Section 3.4) is a semi-automatic method we examined when the user provides some initial information. In particular, we did our experiments using the SIOX tool contained in the open-source program GIMP [102]. Figure 4.12, 4.13 and 4.14 show sample results tested on food images. This underlying processing depends on user initialized foreground and background regions, therefore the performance of this method can be affected the initialization. In particular, when neighboring objects have similar color, the boundary of one object may contain part of the neighboring object. An example of this is shown in Figure 4.13 where part of the hamburger bun is labeled as regions for the fries. Due to the sensitivity to user initialization, we did not deploy the SIOX tool in the mpFR. Instead, we chose to use a simple pen tool for the user to provide a rough contour of the food item to be segmented and the use region-based active contours to refine the segmentation as illustrated in the user feedback refinement block in Figure 3.1.

4.2.5 Multilevel Segmentation

The goal of the multilevel segmentation method is to generate a pool of segments for each image to achieve high probability of obtaining “good” segments that may contain potential objects. Figure 3.15 describes an overview of our approach (also see Section 3.5). Given an input image, we first use salient region detection to identify potential regions in the image containing objects of interest. Our multilevel segmentation approach is then used on each salient region SR1, SR2, SR3 . . . , this step results in a number of segmentations to be classified based on the selected features. Normalized Cuts is used to generate segments in each of the salient regions as indicated in Figure 3.15. The results from the classifier are then used as feedback to the select the “optimal” parameters for the Normalized Cuts (see Section 3.5). Figure 4.15 shows an example of salient regions detected from a meal image.

Figure 4.16 shows the segmentations produced by different methods including human segmentation (ground-truth), our multilevel segmentation (classifier feedback included), and Normalized Cuts with a range of input parameters (number of segments between 3 and 13 and no classifier feedback). The range of input parameters were chosen based on experiments for this set of images. Input parameters that were much lower or higher than this range produced extreme under- or over-segmentations, that were not fair comparison to the proposed multilevel segmentation. Note for the multilevel segmentation approach the input parameters for the Normalized Cuts are selected based the classifier feedback. These segmentations provide a visual comparison of the segmentations generated by each method, and complement the results from segmentation evaluation in Figure 4.17 discussed later. It is encouraging that the precision/recall results for each method agree well with the visually perceived quality of the corresponding segmentations.

4.2.6 Quantitative Segmentation Evaluation

In additional to visual comparison of the proposed multilevel segmentation method to both human segmentations and segmentations produced by the Normalized Cuts method, we also examined quantitative evaluation by generating precision/recall scores for a range of input parameters (number of segments between 3 and 13) of the Normalized Cuts method, the optimal parameter, as well as human segmentation (see Section 3.6). The segmentation output of our proposed method is a labeled image where pixels within the same region have identical labels. To extract the region boundaries from the label images, we could simply mark as a boundary any pixels that have a neighbor with a different label. This would result in boundaries that are two pixels thick. This is not ideal because very small regions will disappear and replaced by clusters of boundary pixels, i.e. the region-structure within is lost. Also, thick boundaries are likely to introduce unwanted artifacts in the resulting precision/recall scores.

To eliminate these problems, we generate boundary images that are twice the resolution of the original segmentations. Boundaries in the higher-resolution images can lie between the pixels corresponding to the original regions in the low resolution segmentation. The procedure for generating the up-sampled boundaries includes up-sampling of each region in the original segmentation in the higher-resolution image. A final boundary map is formed by the union of the up-sampled boundaries of each regions.

Due to the computationally intensive nature of our multilevel segmentation approach, we did the evaluation on images that have been down-sampled by a factor of 4 from their original size in our image database. Since the ground-truth segmentations have the same size as the original images, we can either up-sample the automatic segmentation to the appropriate size, or down-sample the human segmentation. To reduce the computational cost of evaluating the error measures, we chose the latter option.

To obtain a meaningful benchmark, for each combination of parameters, we tested the methods on a set of 130 food images. The scores for the method for a particular combination of input parameters is the median of the precision and recall scores obtained for the individual image. The median precision and recall scores computed for different combinations of input parameters yield tuning curves characterizing the performance of the proposed methods.

Since the Normalized Cuts and multilevel segmentation methods take only one input parameter, which is the desired number of segments, we tested these methods for a number of output segments within a chosen range of input parameter that yield reasonable segmentations appropriate for our dataset. For the Normalized Cuts method (no classifier feedback), the range of input parameter is [3,13]. The multilevel segmentation method automatically selects the optimal input parameter from the classifier feedback.

We choose for comparison the turing curves resulting from various precision and recall scores. From these curves, the appropriate parameters can be selected that will yield a target value of either precision or recall for a given method. More importantly, they provide a direct way of comparing the quality of the segmentation produced by different methods across a wide range of input parameters. We can tell from these tuning curves if a method performs consistently better than others across its particular range of parameters, and then rank methods by performance for particular values of precision or recall.

Figure 4.17 offers quantitate evidence that our proposed multilevel segmentation method outperforms the Normalized Cuts (no classifier feedback) across the range of tested input parameters. Comparing the four sets of curves, the scores for all tuning curves fall sharply as d decreases. This is consistent with observation made by Martin et al. [122] and supports the use of a smaller matching radius during evaluation. Furthermore, such observation indicates that the matching distance d should be at least two pixels apart to obtain meaningful precision/recall scores.

4.2.7 Classification Based Segmentation Evaluation

Since performance of the multilevel segmentation depends on both segmentation and classification, we evaluated the segmentation performance in the context of object classification. For these experiments we considered images from two nutritional studies. This set of images have a total of 32 food classes from 200 images, each image consists of 6-7 food classes. We divided the dataset into training and testing, for each category approximately half of the images are training data and the other half are testing data. We use a minimum of 15 training samples per category. Examples of food objects used in our experiments are shown in Figure 4.18.

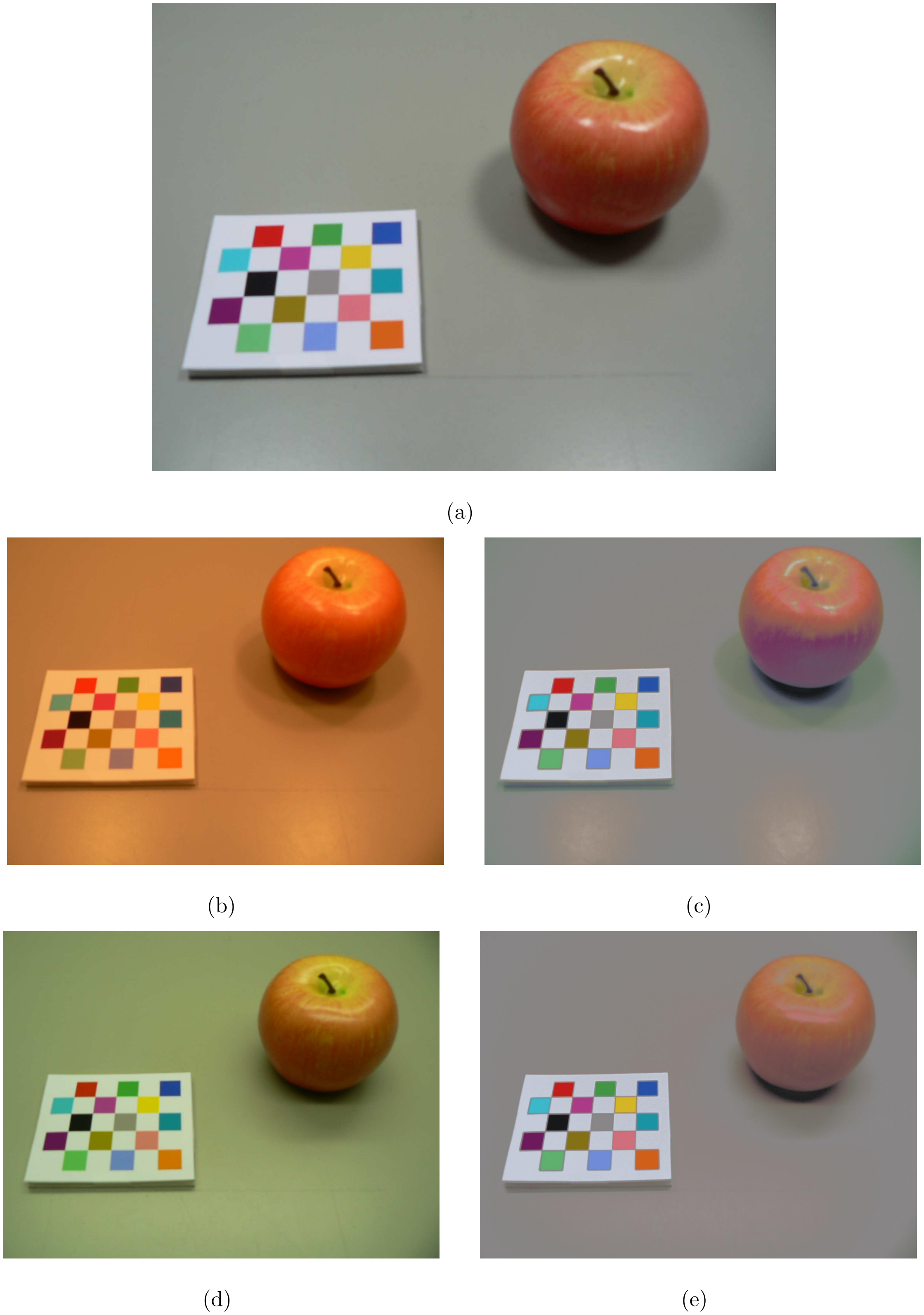
Let Δ denote the $N_f \times N_f$ confusion matrix (for these experiment, the number of classes $N_f = 33$, 32 food classes plus background). Then, $\Delta(i, j)$ is the number of pixels predicted as class j when they actually belonged to class i, divided by the total number of pixels actually belonging to class i. Thus, higher values on the main diagonal of the confusion matrix indicate good performance. Note this is a stricter measure of classification compare to conventional object classification. In conventional object classification, only the class labels identified in an image are used as accuracy measure compared to the ground-truth data, pixel locations of the identified classes are not taken into consideration. In this evaluation, the label for every pixel in an image is counted for accuracy. Therefore, identified objects as well as their locations are evaluated against ground-truth labels. Table 4.3 shows the diagonal entries of the confusion matrix, that is the average accuracy for all the classes. Final average classification accuracy for 32 food classes is 44%.

The above evaluation combines errors from both segmentation and classification. If an image is perfectly segmented, i.e. it completely agrees with the ground-truth, error in classification contributes to pixels being incorrectly labeled. If a region of the image is poorly segmented, it may not represent the visual characteristics of associated food class, therefore resulting in wrong class labels for pixels belong to that region. Some foods are inherently difficult to classify due their similarity in the feature space; others are difficult to segment due to faint boundary edges that camouflage the food item; as well as the non-homogeneous nature of certain foods. For example, yellow cake has an average classification accuracy of 11% because its appearance is very similar to that of cream cheese (Figure 4.18). Coke has an average classification accuracy of 16% due to its visual similarity to coffee and difficulty in segmenting it with the black background. BBQ chicken also has a low accuracy of 28% due to its non-homogeneous property and visual variation depending on the placement of the BBQ sauce on top of the chicken. (Figure 4.18). Since foods are generally served in certain combinations, we can explore contextual information in addition to visual characteristics to increase accuracy of classification [41].

Based on our evaluation, we have shown improved accuracy of segmenting food images using proposed multilevel segmentation approach compared to the Normalized Cuts method without classifier's feedback. Future development in both segmentation and classification will likely to further improve the performance of overall image analysis task.

Table 4.3
Segmentation Accuracy for Each Class and Average Performance.

Background	Apple Juice	Bagel	Barbecue Chicken	Broccoli
97%	29%	98%	28%	66%
Brownie	Cheeseburger	Chocolate cake	Coffee	Coke
13%	17%	14%	93%	16%
Cream Cheese	Scrambled Egg	French Dressing	French Fries	Fruit Cocktail
95%	35%	35%	73%	78%
Garlic Bread	Green Beans	Lettuce	Mac&Cheese	Margarine
29%	17%	34%	57%	31%
Mashed Potato	Milk	Orange Juice	Peach	Pear
49%	11%	48%	24%	24%
Pineapple	Pork Chop	Sausage Links	Spaghetti	Sugar Cookie
38%	11%	30%	43%	69%
Vegetable Beef Soup	White Toast	Yellow Cake		**Average**
63%	64%	13%		**44%**



(a)

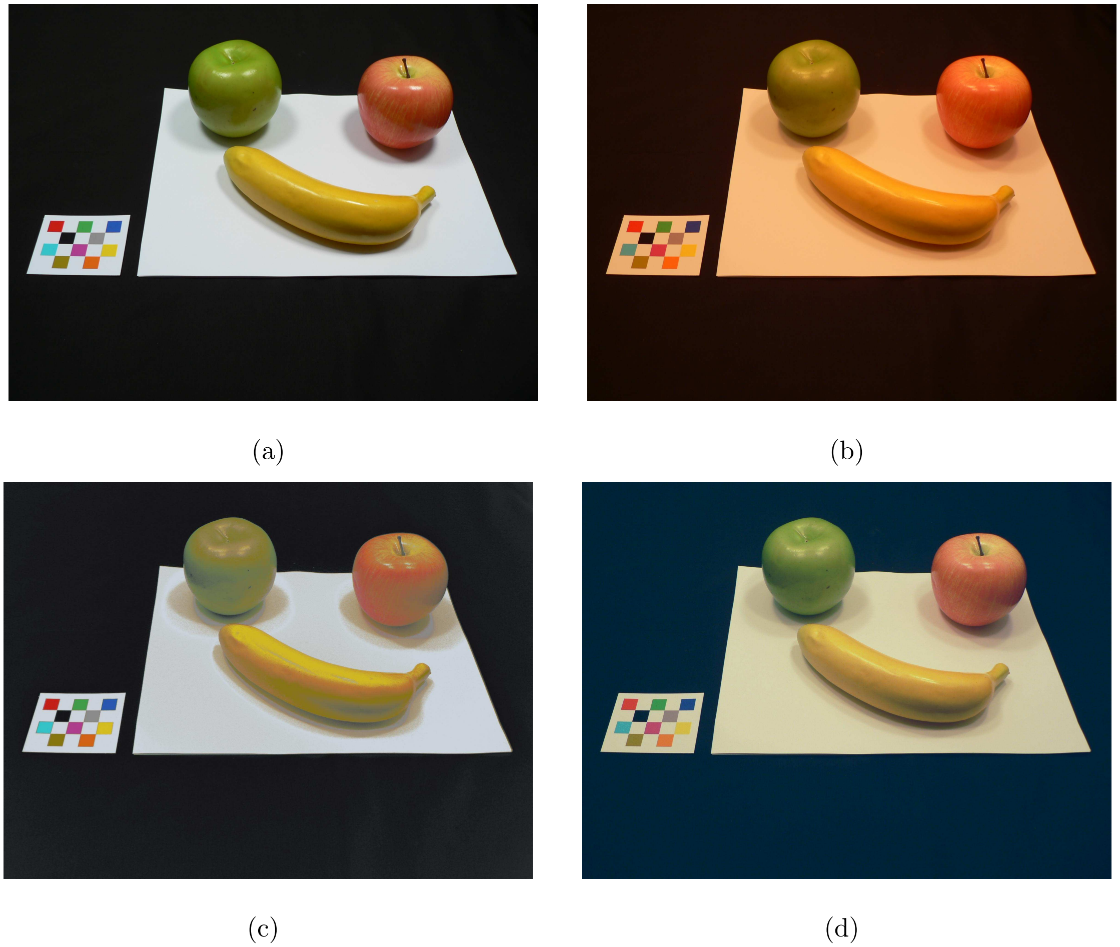
(b)

(c)

(d)

(e)

Fig. 4.1. Examples of using color fiducial marker to correct colors in images with unknown illuminations. (a) is the reference image containing the color fiducial marker, (b) is the same image taken under different illumination, (c) shows the color correct version of the test image, (d) is the same image taken under a second different illumination, (e) shows the color correct version of the test image.



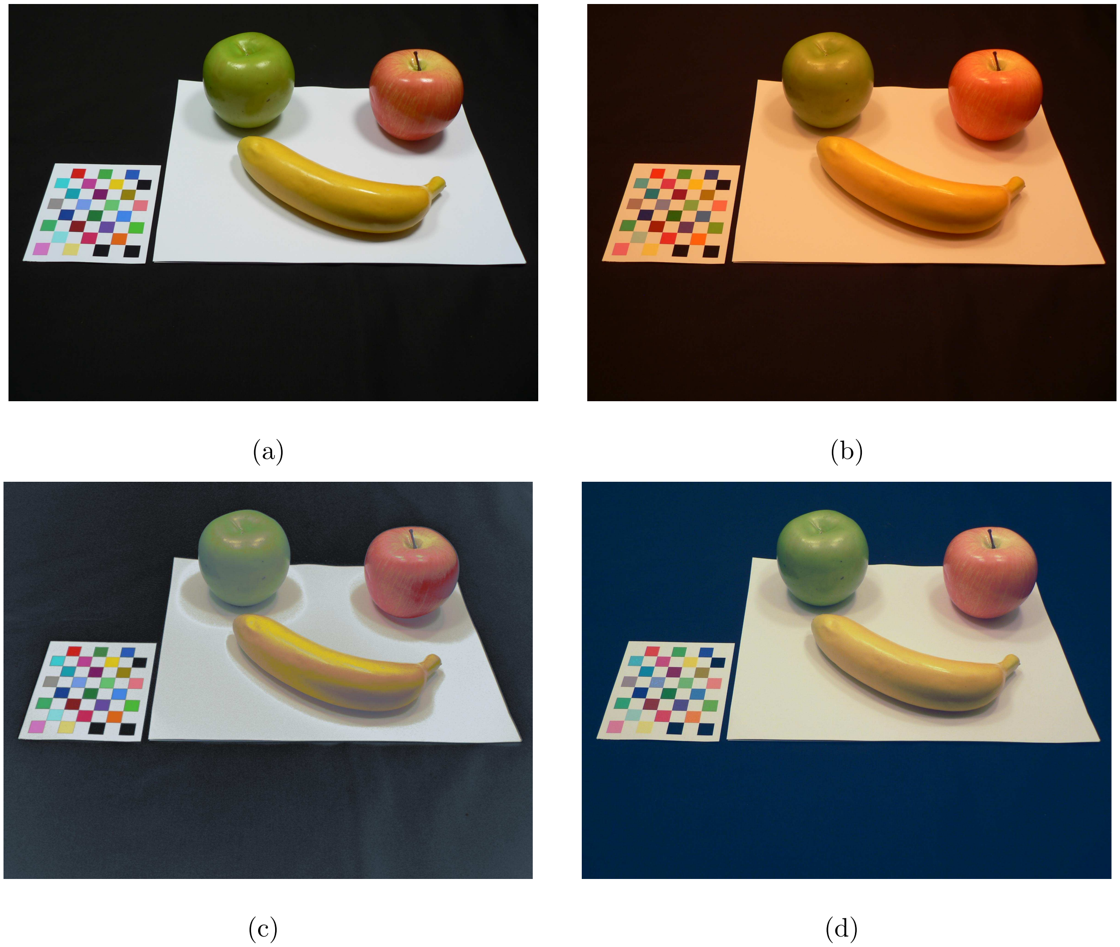
(a)

(b)

(c)

(d)

Fig. 4.2. Examples of using color fiducial marker to correct colors in images with unknown illuminations. (a) is the reference image containing the color fiducial marker, (b) is the same image taken under different illumination, (c) shows the color correct version of the test image using 3D LUT interpolation, (d) shows the color correct version of the test image using Choi's method.



(a)

(b)

(c)

(d)

Fig. 4.3. Examples of using a 7 × 8 color fiducial marker to correct colors in images with unknown illuminations. (a) is the reference image containing the color fiducial marker, (b) is the same image taken under different illumination, (c) shows the color correct version of the test image using 3D LUT interpolation, (d) shows the color correct version of the test image using Choi's method.

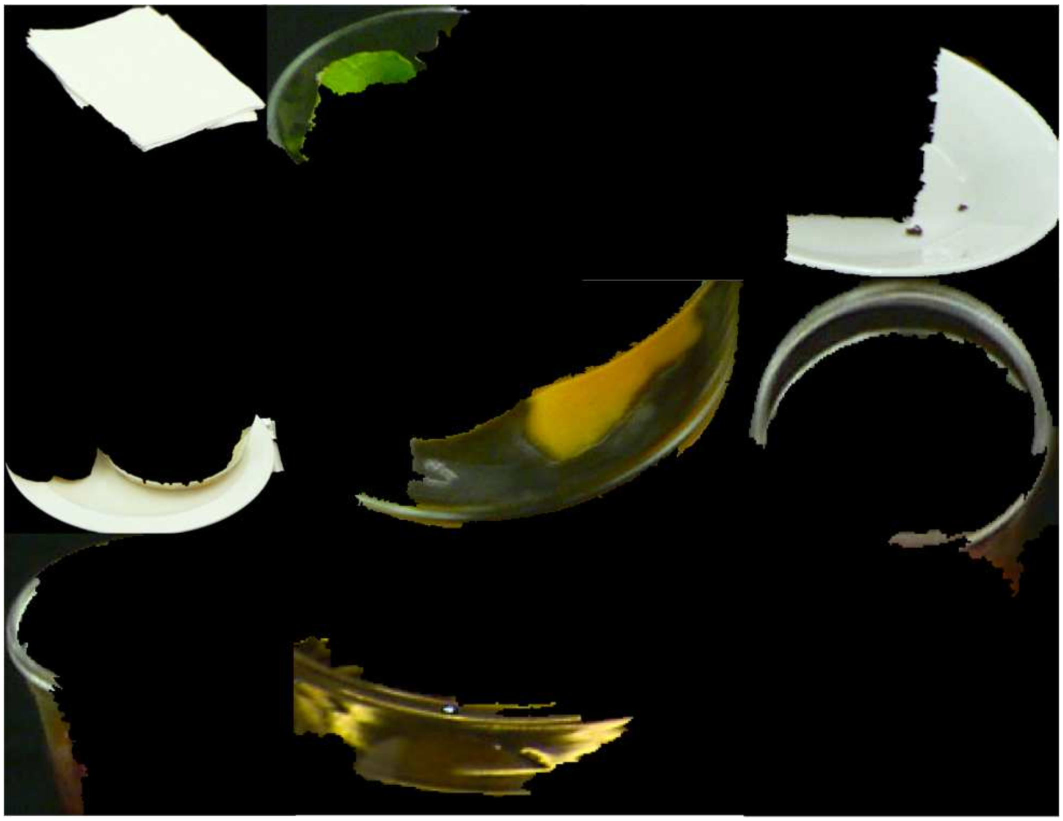
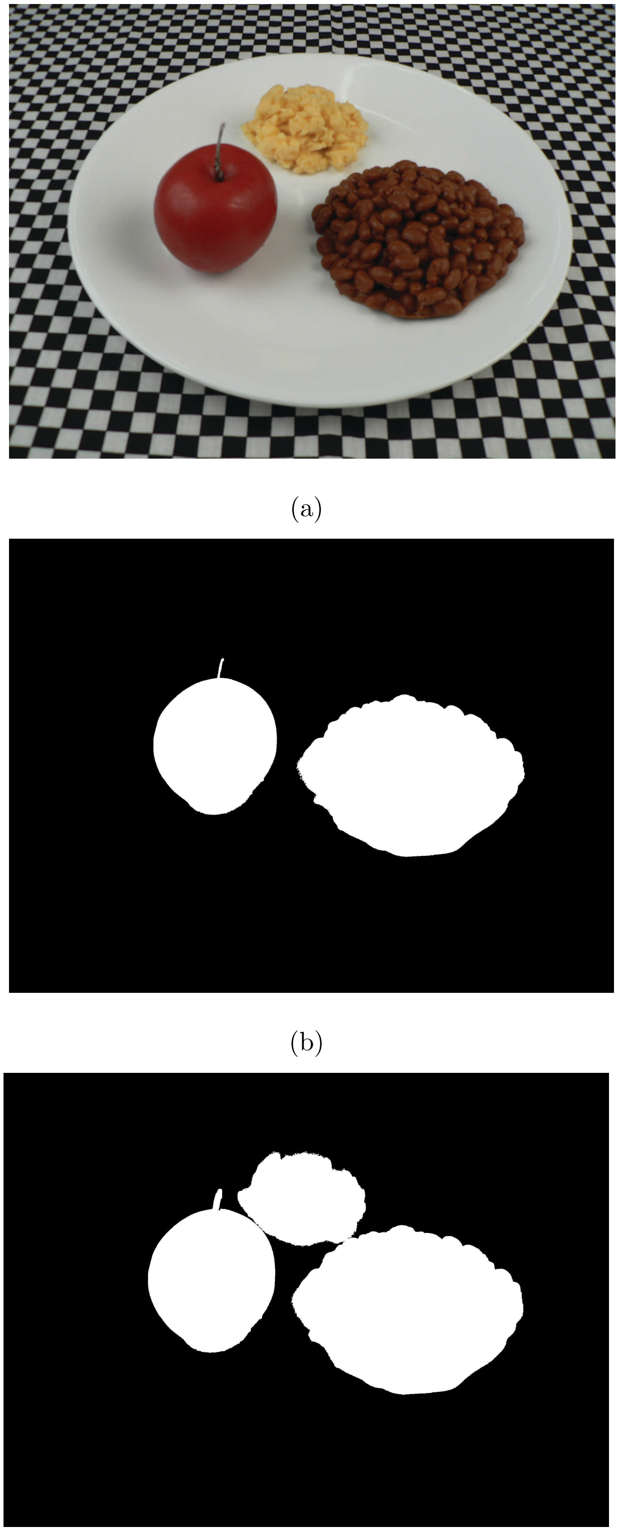


Fig. 4.4. Examples of non-food segments from our image analysis.

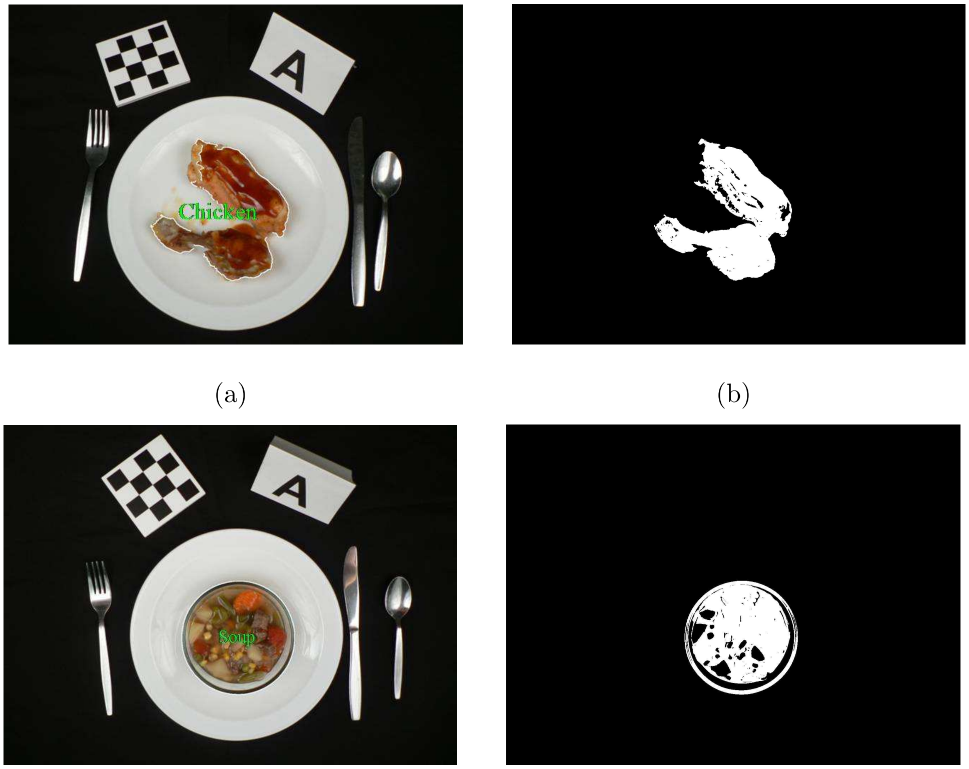


(a)

(b)

(c)

Fig. 4.5. Sample Results of Connect Component Labeling. (a) A Typical Image of a Meal, (b) Food Item Segmented Using a Fix Threshold (T = 127), (c) Additional Food Item Segmented Using Color Information.



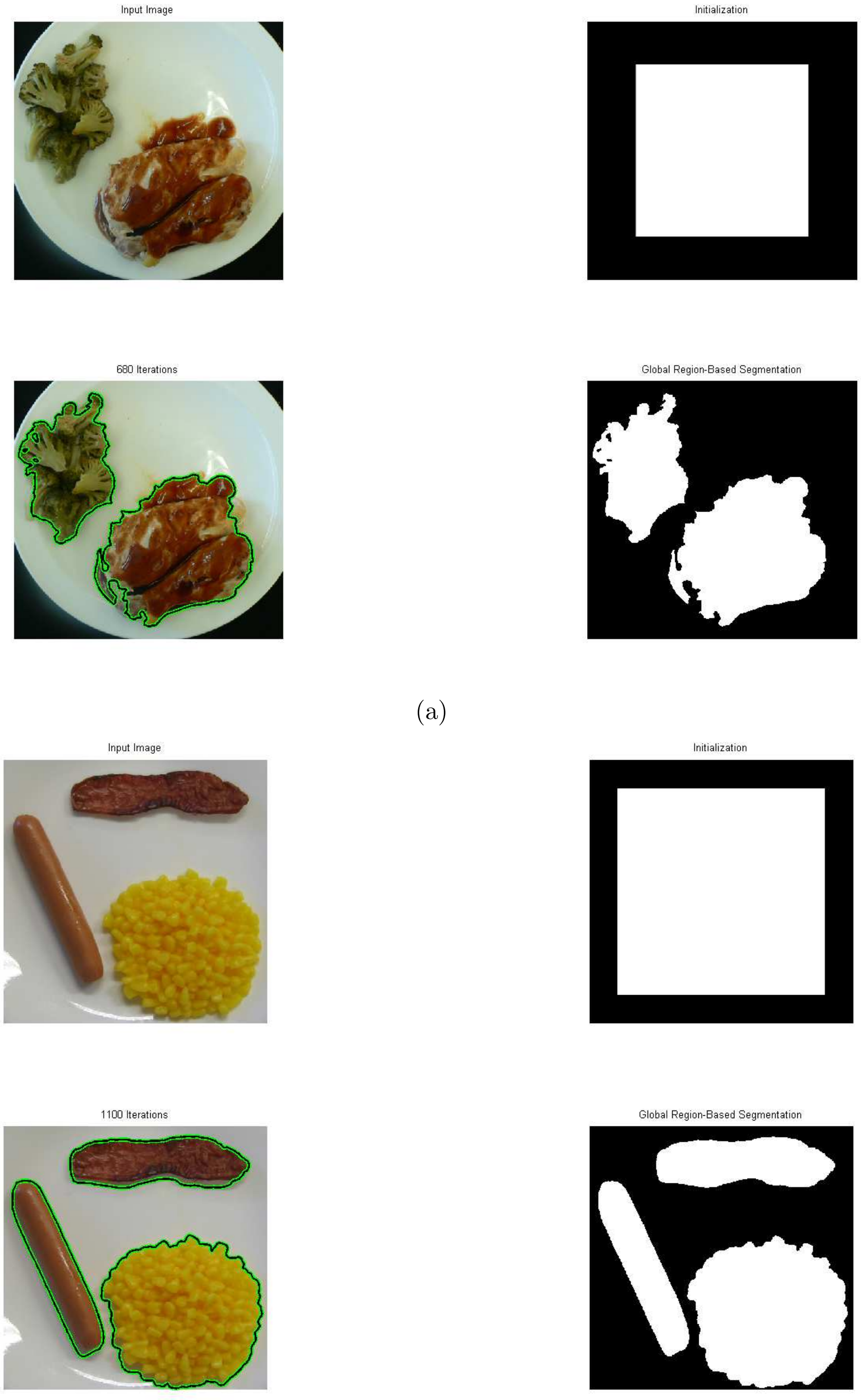
(a)

(b)

(c)

(d)

Fig. 4.6. Sample Segmentation of Real Food Image Using Connect Component Labeling. (a) BBQ Chicken, (b) Binary Image with White Region Showing the Segmented Item and Black the Background, (c) Vegetable Soup, (b) Binary Image with White Region Showing the Segmented Item and Black the Background,



(a)

(b)

Fig. 4.7. Sample Results of Active Contour Segmentation.(a) and (b) each contains the original image (upper left), initial contour (upper right), segmented object boundary (lower left), and binary mask (lower right).

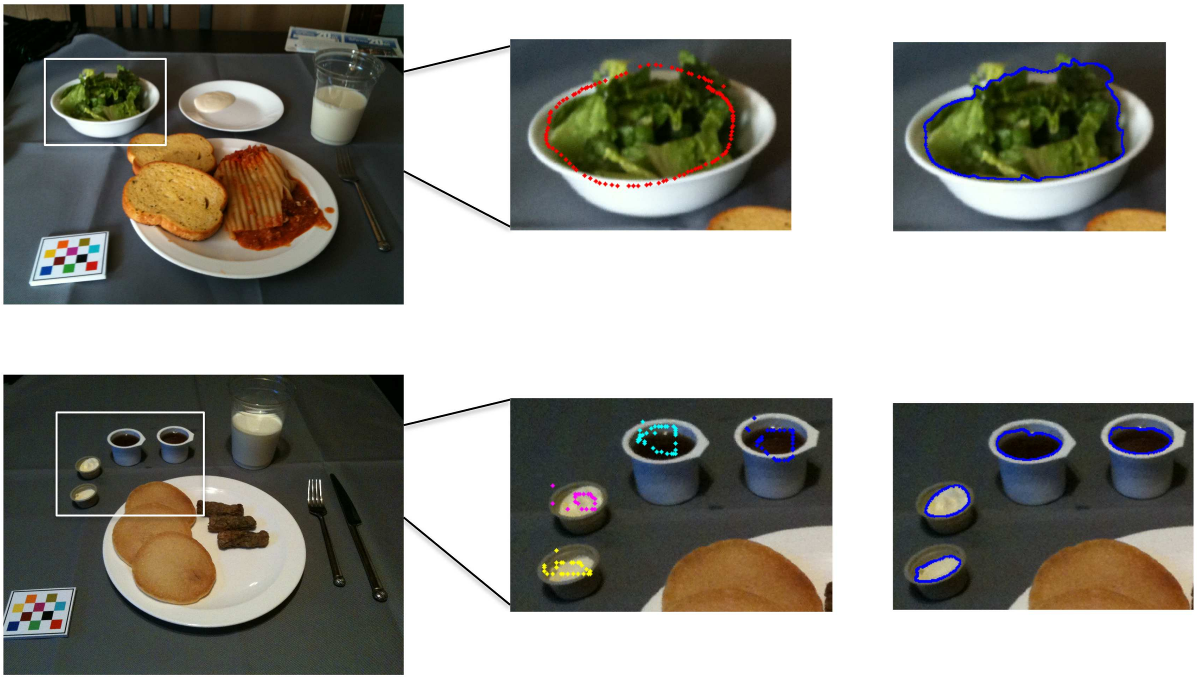


Fig. 4.8. Segmentation Results Using Active Contours Based on User Feedback. From Left to Right: Original Image, User Supplied Initialization Points, Final Segmentation Contours.

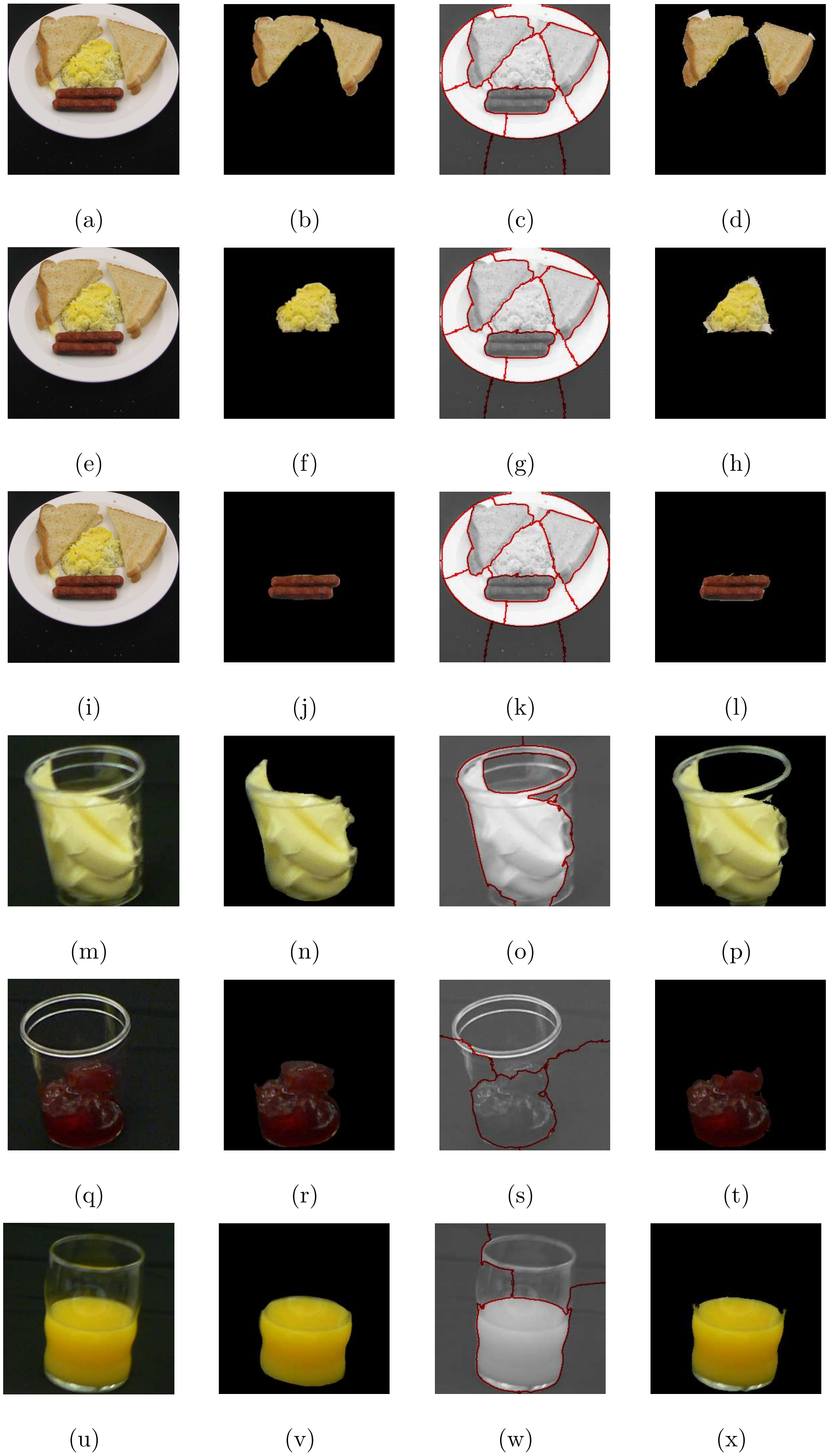
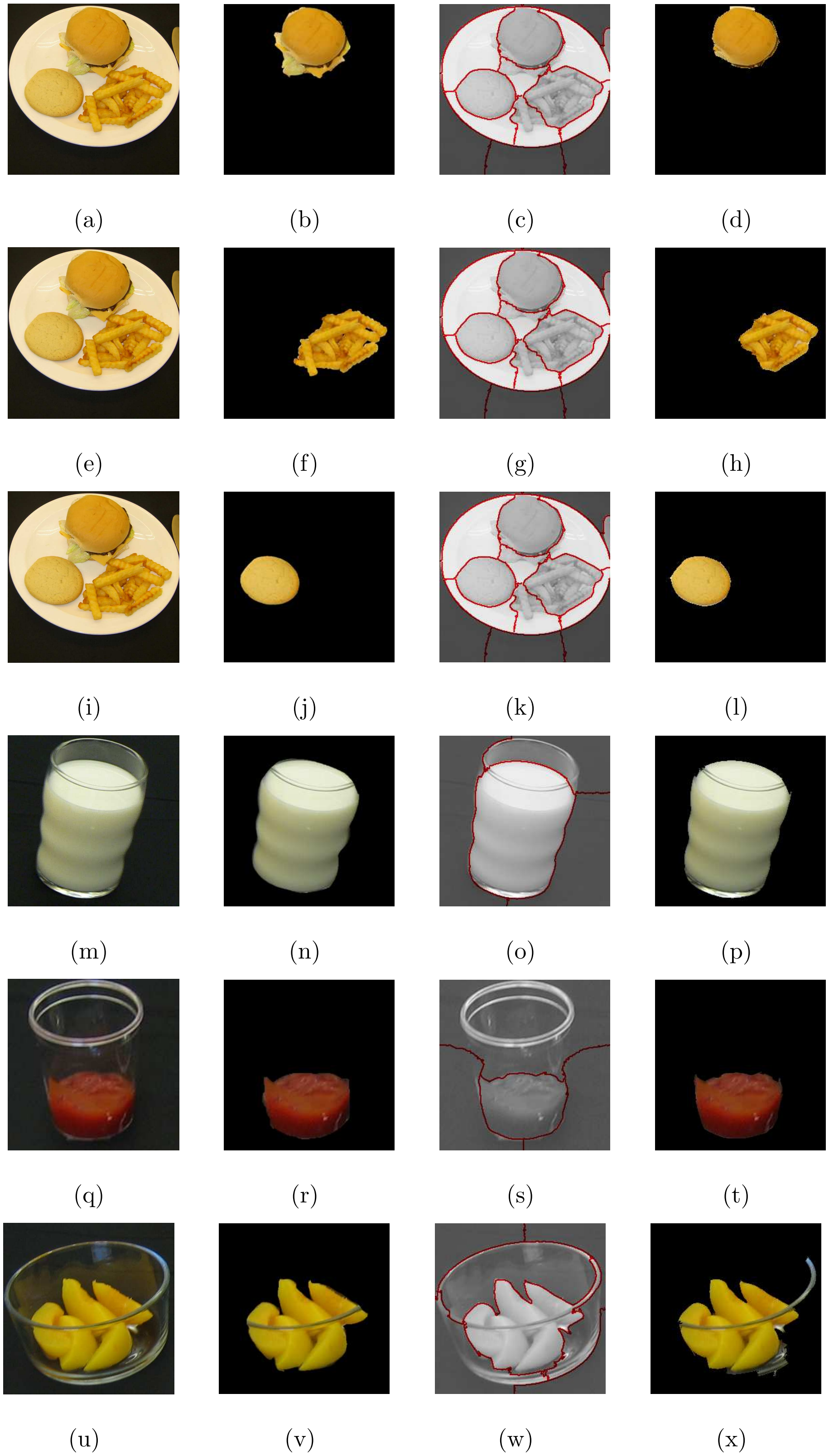


Fig. 4.9. Sample Results of Normalized Cut for Breakfast Menu. First column contains original images, second column show the ground-truth segmentation, third column show the segmented object boundary using Normalized Cut method, fourth column show the extract object from segmentation.


(a) (b) (c) (d)

(e) (f) (g) (h)

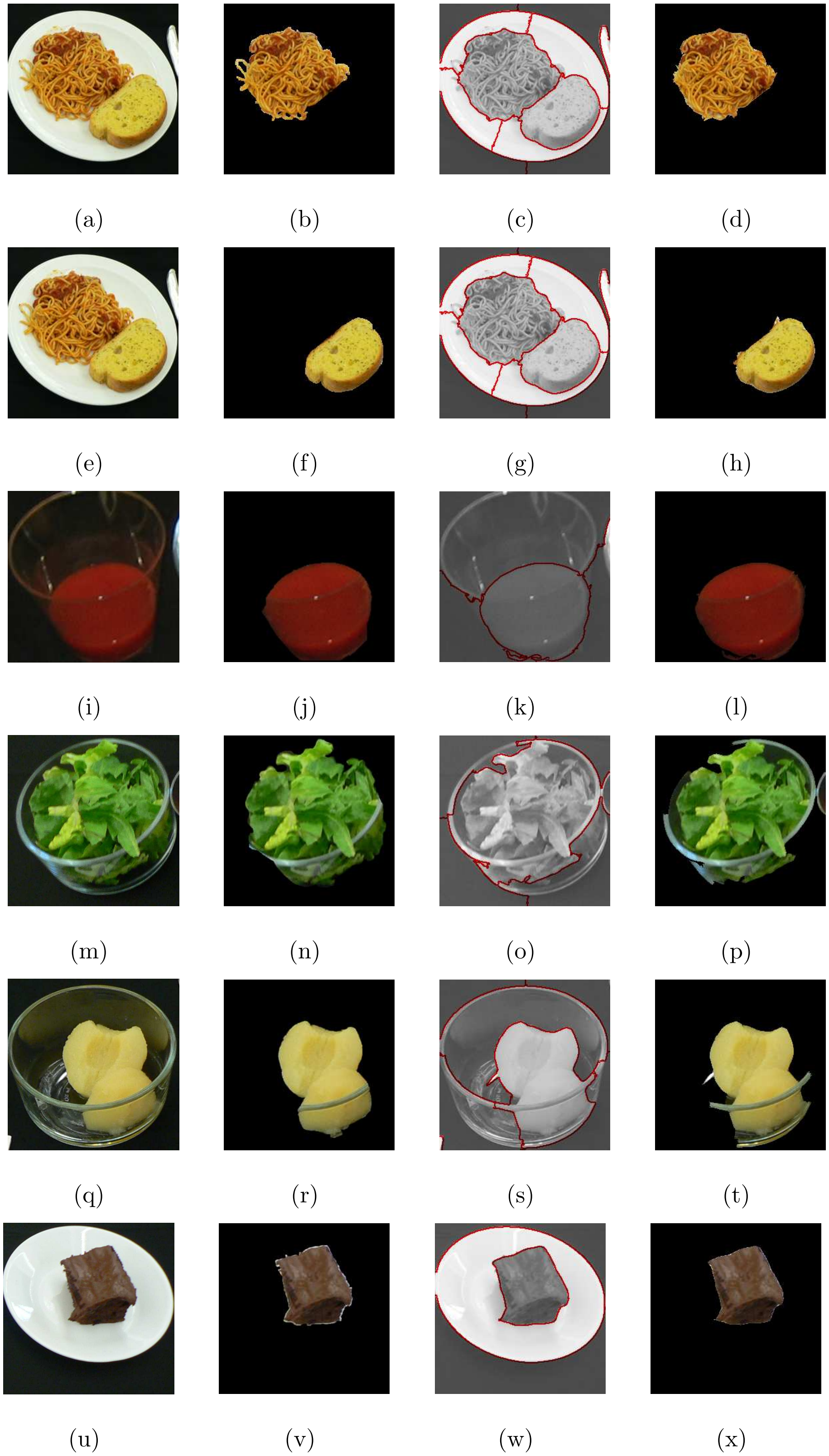
(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

(u) (v) (w) (x)

Fig. 4.10. Sample Results of Normalized Cut for Lunch Menu. First column contains original images, second column show the groundtruth segmentation, third column show the segmented object boundary using Normalized Cut method, fourth column show the extract object from segmentation.



(a) (b) (c) (d)

(e) (f) (g) (h)

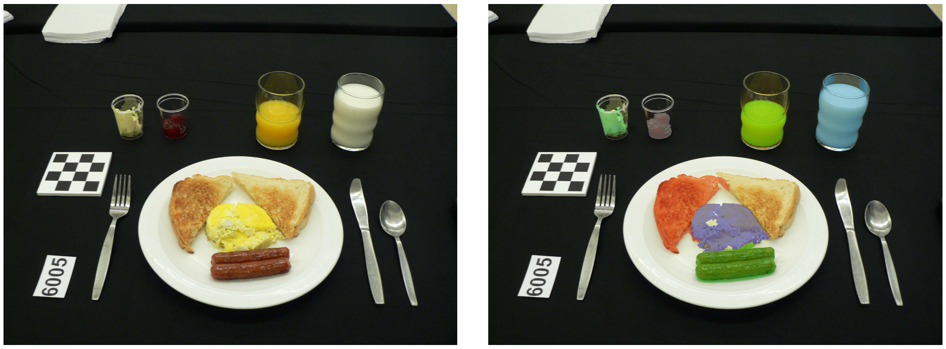
(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

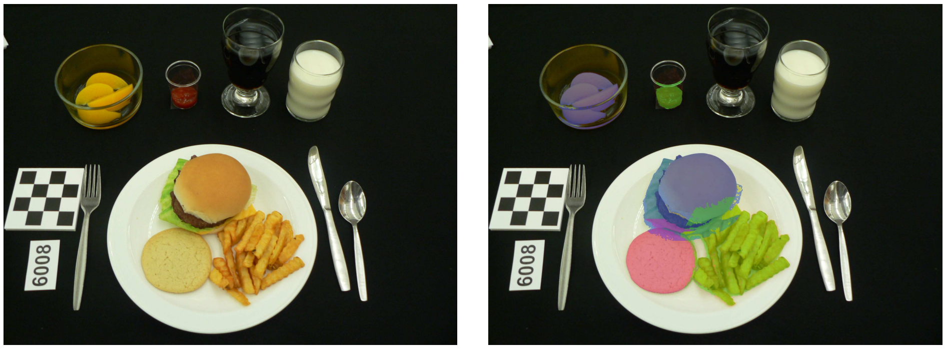
(u) (v) (w) (x)

Fig. 4.11. Sample Results of Normalized Cut for Dinner Menu. First column contains original images, second column show the ground-truth segmentation, third column show the segmented object boundary using Normalized Cut method, fourth column show the extract object from segmentation.



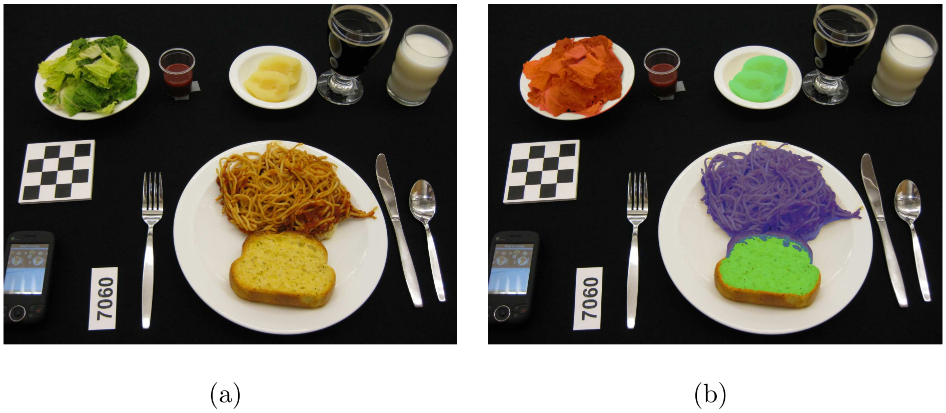
(a) (b)

Fig. 4.12. Sample Result Using The SIOX Tool. (a) original image, (b) overlay image.



(a) (b)

Fig. 4.13. Sample Result Using The SIOX Tool. (a) original image, (b) overlay image.



(a)

(b)

Fig. 4.14. Sample Result Using The SIOX Tool, (a) original image, (b) overlay image.

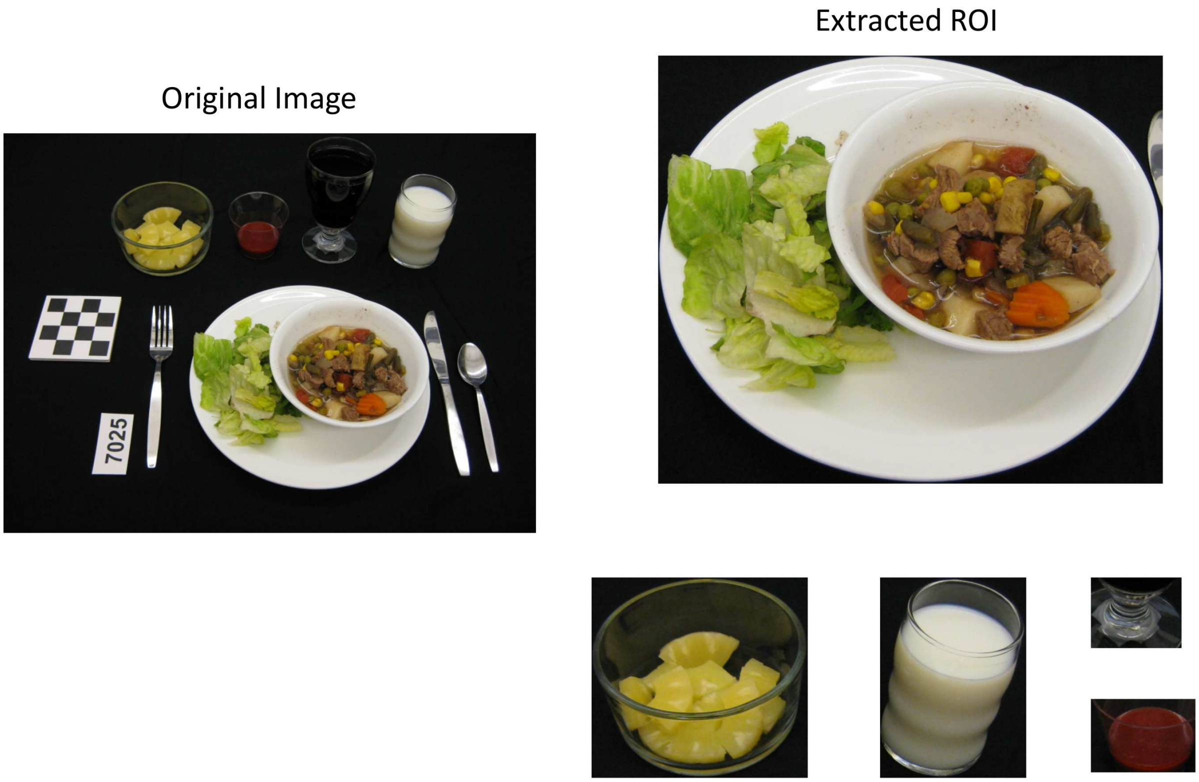


Fig. 4.15. An Example of Salient Regions Detected From a Meal Image.

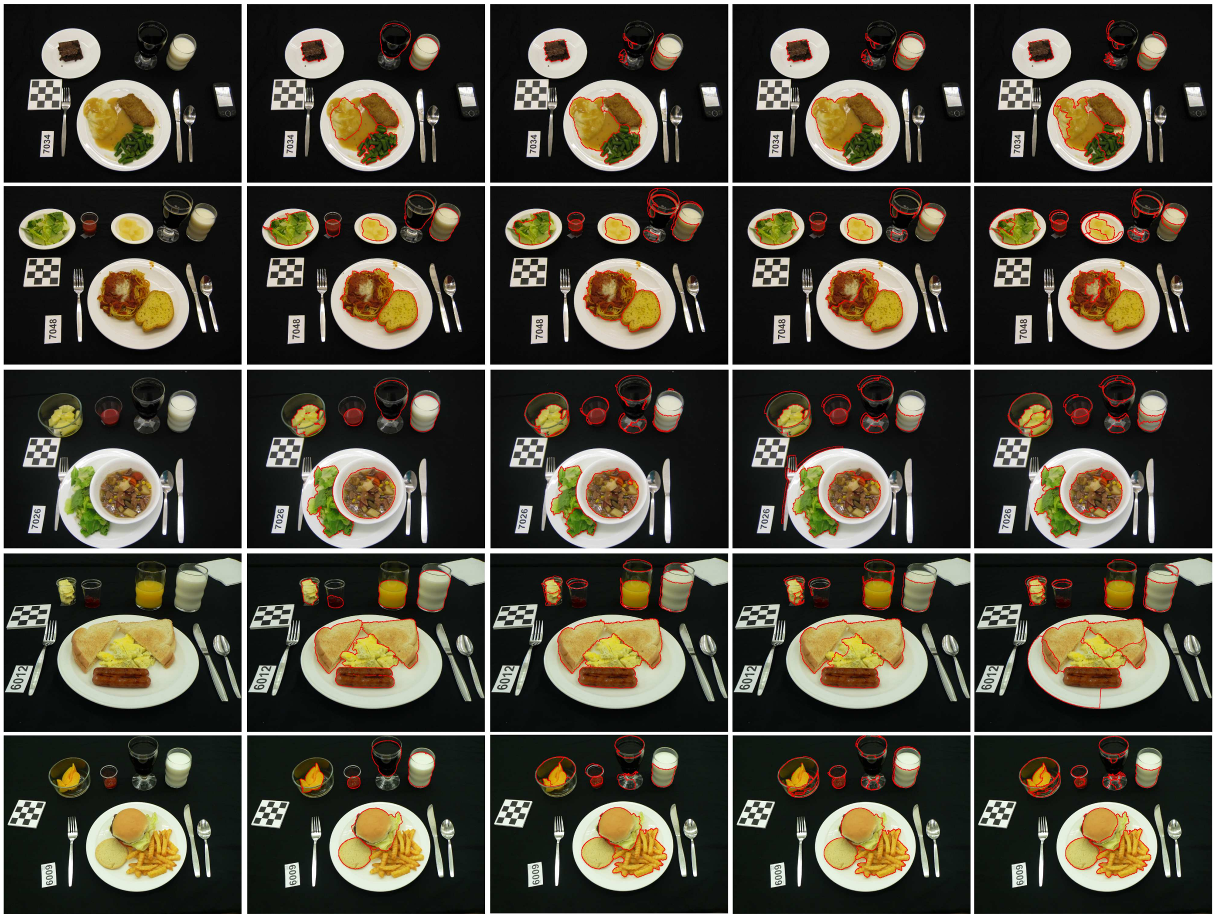
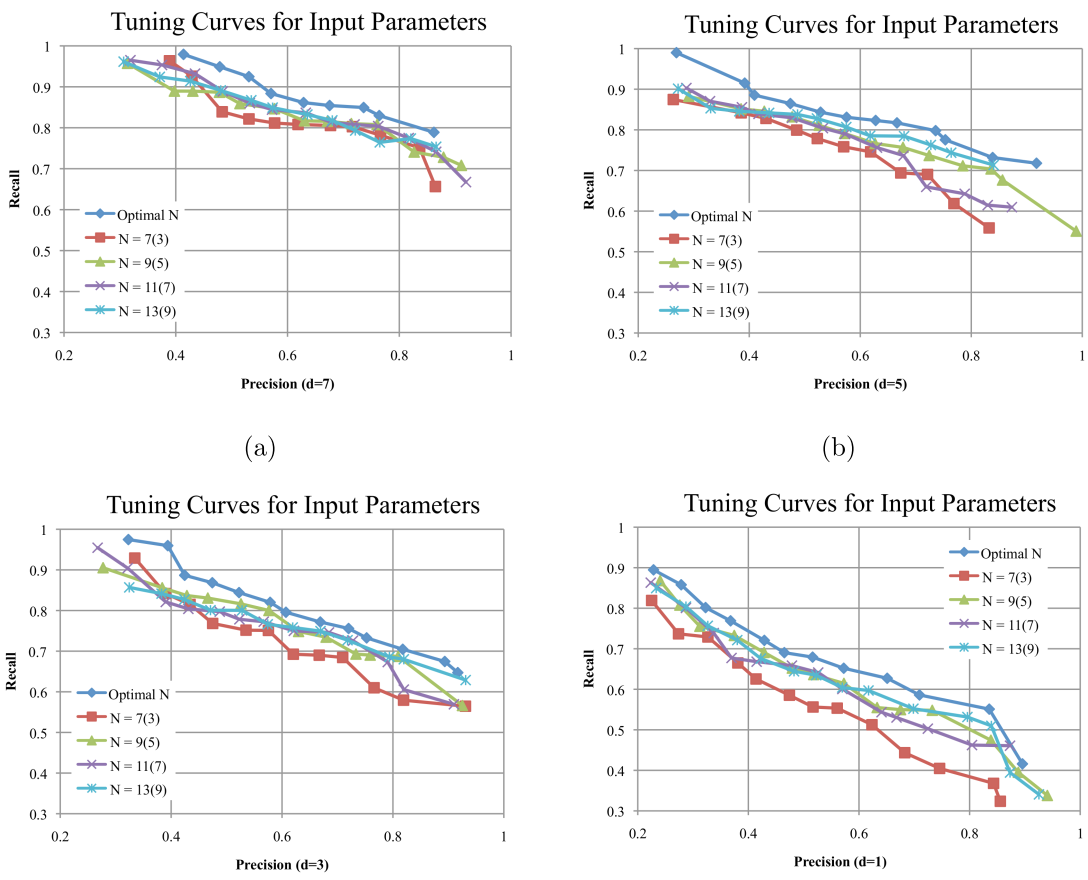


Fig. 4.16. Visual Comparison of the Multilevel Segmentation and Other Methods. From left to right: Original image, Ground-truth segmentation, Multilevel segmentation, Normalized Cuts with small input parameter ($N = 3$), Normalized Cuts with larger input parameter ($N = 13$).



(a)

(b)

(c)

(d)

Fig. 4.17. Tuning Curves for Normalized Cuts (no classifier feedback) and multilevel segmentation (with classifier feedback) methods for (a) $d = 7$, (b) $d = 5$, (c) $d = 3$, and (d) $d = 1$. For curves generated by Normalized Cuts, we choose input parameters most favorable for this method given the test images. The optimal input parameter is automatically chosen based on the proposed multilevel segmentation method.

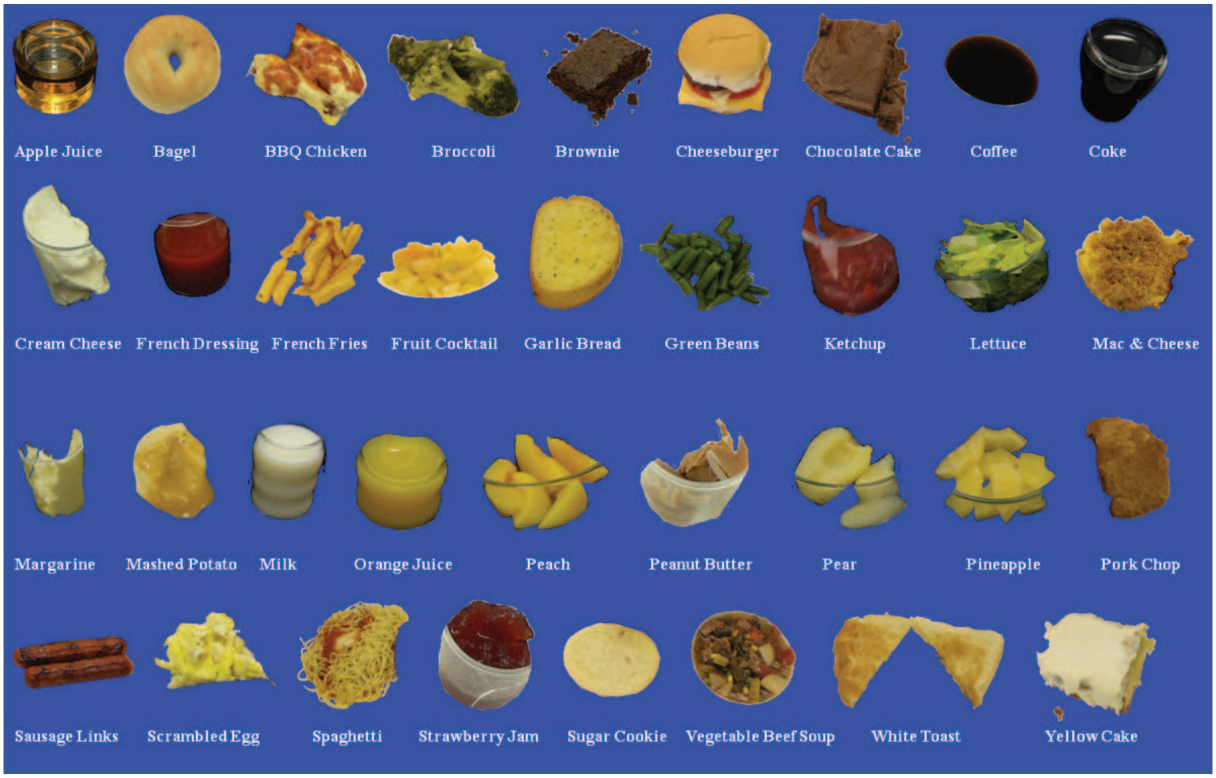


Fig. 4.18. Sample Images of Food Objects Used in Our Experiments.

5. CONCLUSION AND FUTURE WORK

We have developed a dietary assessment system using mobile devices. This system has been deployed on an iPhone and it is currently being used by dietitians and nutritionists in the Department of Foods and Nutrition at Purdue University for various nutrition studies.

5.1 Conclusions

The research in this thesis focuses on developing methods for image segmentation and particularly for the segmentation of food images. My main contributions are as follows:

- Developed a multiple hypothesis segmentation system to select optimal segmentations based on confidence scores assigned to each segment. This approach combined two ideas: a set of segmented objects could be partitioned into perceptually similar object classes based on global and local features; and perceptually similar object classes could be used to assess the accuracy of image segmentation. Both segmentation and classification accuracy were improved by generating multiple segmentations of each image using multiscale graph decomposition.

- Evaluated the quantitative performance of our multilevel segmentation approach based on comparing region boundaries with ground-truth data for consistency. A separate evaluation was performed in the context of object classification to assess pixel level accuracy for identifying each food class.

- Demonstrated the use of region-based active contour models to refine the image segmentation based on user feedback from the TADA mpFR. With such feedback, an initial curve, was deformed to the boundary of the object under constraints from the image.

- Developed a color fiducial marker used as the reference object in images captured under various illumination conditions. Constructed a transformation based on the reference illumination to correct the colors of the unknown image. The realization of the method was achieved through the use of a 3D LUT.
- Examined color and texture features for each segmented food region. The was part of the initial classification methods used in the TADA system.These features were classified using SVM.

5.2 Future Work

In our image analysis system, performance of the image segmentation directly affects the accuracy of food identification and automatic portion estimation. A recent study [127] has shown the use of image segmentation can improve object categorization accuracy due to the fact that by segmenting the object of interest, the noise introduced by the background around the object is minimized. An example is shown in [128] where flowers are segmented from the background to increase recognition accuracy. However, methods of unsupervised image segmentation have not been popular as preprocessing for recognition and categorization mainly because of the unsatisfactory quality of image segmentation methods. Here we discuss potential solutions to the problem.

- The goal of an unsupervised clustering method is to partition the data based on some criterion that does not use labeled examples. As a result, there are many open problems in this area that include choosing the appropriate grouping criterion and the number of clusters. In our proposed multilevel segmentation approach, we use the confidence scores associated with class labels assigned to each image partition as a measure for choosing segmentation parameters. This method relies on the performance of the object recognition task and its underly feature representation of object categories. Other criteria for selecting stable segmentation parameters should be explored to generate meaningful image partitions that will not be affected by performance of other image analysis tasks. Rabinovich, et. al [93] suggest the use of stability as a predictor of "goodness" for a specific set of segmentation parameters.

- One of the general assumption of image segmentation is that each sub-region has a uniform property, such as homogeneous image intensity. However, this assumption is not always true in real images. Another assumption of traditional segmentation methods for multispectral images is that a scalar expression can represent vector-valued data, such as the use of brightness or luminance to represent different colors. However, this is also not true in many cases. In [129], the region-based active contour model is used to partition an image based on the uniformity of statistical property of a subset, such as a multivariate Gaussian distribution. The authors [130] recognize the shortcoming of the method when the statistical property of a subset is non-uniform. As a result, they proposed the use of mixture density model to represent the non-uniform statistical property of a subset.

- Incorporating prior knowledge into the chosen segmentation method can drastically improve the accuracy of image segmentation. For our application, assumption is made to the location of food items as majority of the foods are placed in tableware such as plates, bowls and cups. Currently we use shape information of these "tablewares" to locate the proximity of possible foods. However, such information becomes unpredictable in free-living situation when different users have their own eating habits. Increased complexity of eating environment also makes it difficult to detect potential food objects using current methods.

5.3 Publications Resulting from this Work

Journal Papers:

1. **Fengqing Zhu**, Marc Bosch, Insoo Woo, SungYe Kim, Carol J. Boushey, David S. Ebert, and Edward 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, August 2010, pp. 756-766.

2. Bethany L. Six, TusaRebecca E. Schap, **Fengqing Zhu**, Anand Mariappan, Marc Bosch, Edward J. Delp, David S. Ebert, Deborah A. Kerr, and Carol J. Boushey, "Evidence-Based Development of a Mobile Telephone Food Record," *Journal of American Dietetic Association*, January 2010, pp. 74-79.

3. TusaRebecca E Schap, **Fengqing Zhu**, Edward J. Delp, and Carol J. Boushey, "Merging Dietary Assessment with the Adolescent Lifestyle," *Journal of Human Nutrition and Dietetics*, submitted.

Conference Papers:

1. **Fengqing Zhu**, Marc Bosch, Ziad Ahmad, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Challenges in Using a Mobile Device Food Record Among Adults in Free-living Situations," *mHealth Summit*, Washington DC, December 2011.

2. **Fengqing Zhu**, Marc Bosch, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Multilevel Segmentation for Food Classification in Dietary Assessment," *Proceedings of* 7^{th} *International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, September 4-6, 2011, pp. 337-342.

3. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Combining Global and Local Features for Food Identification and Dietary Assessment," *Proceedings of the International Conference on Image Processing*, Brussels, Belgium, September 2011.

4. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Food Texture Descriptors Based on Fractal and Local Gradient Information," *Proceedings of the* 19*th* *European Signal Processing Conference*, Barcelona, Spain, September 2011.

5. Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, **Fengqing Zhu**, Carol J. Boushey and Edward J. Delp, "Integrated Databases System for Mobile Dietary Assessment and Analysis," *Proceedings of the* 1*st* *IEEE International Workshop on Multimedia Services and Technologies for E-health in conjunction with the International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011.

6. **Fengqing Zhu**, Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, David S. Ebert, Carol J. Boushey and Edward J. Delp, "Segmentation Assisted Food Classification for Dietary Assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, Vol. 7873, Burlingame, California, January 2011.

7. JungHoon Chae, Insoo Woo, SungYe Kim, Ross Maciejewski, **Fengqing Zhu**, Edward J. Delp, Carol J. Boushey, and David 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, Burlingame, California, January 2011.

8. **Fengqing Zhu**, Marc Bosch, Carol J. Boushey and Edward J. Delp, "An Image Analysis System for Dietary Assessment and Evaluation," *Proceedings of the International Conference on Image Processing*, Hong Kong, China, September, 2010.

9. Anand Mariappan, Marc Bosch, **Fengqing Zhu**, Carol J. Boushey, David S. Ebert, Deborah A. Kerr, and Edward J. Delp, "Personal Dietary Assessment Using Mobile Devices," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VII*, Vol. 7246, San Jose, California, January 2009.

10. **Fengqing Zhu**, Anand Mariappan, Carol J. Boushey, Deborah A. Kerr, Kyle Lutes, David S. Ebert, and Edward J. Delp, "Technology-Assisted Dietary Assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VI*, Vol. 6814, San Jose, California, January 2008.

LIST OF REFERENCES

LIST OF REFERENCES

[1] F. Zhu, M. Bosch, I. Woo, S. Kim, C. Boushey, D. Ebert, and E. 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.

[2] C. Ogden, M. Carrol, L. Curtin, M. Lamb, and K. Flegal, “Prevalence of high body mass index in us children and adolescents, 2007-2008,” *Journal of the American Medical Association*, vol. 303, no. 3, pp. 242–249, January 2010.

[3] A. Fagot-Campagna, J. Saadinem, K. Flegal, and G. Beckles, “Diabetes, impaired fasting glucose, and elevated hba1c in US adolescents: the third National Health and Nutrition Examination Survey,” *Diabetes Care*, vol. 24, pp. 834–837, 2001.

[4] M. Livingstone, P. Robson, and J. Wallace, “Issues in dietary intake assessment of children and adolescents,” *British Journal of Nutrition*, vol. 92, pp. S213–S222, 2004.

[5] H. Rockett, C. Berkey, and G. Colditz, “Evaluation of dietary assessment instruments in adolescents,” *Current Opinion in Clinical Nutrition & Metabolic Care*, vol. 6, pp. 557–562, 2003.

[6] R. McPherson, D. Hoelscher, M. Alexander, K. Scanlon, and M. Serdula, “Dietary assessment methods among school-aged children: validity and reliabality,” *Preventive Medicine*, vol. 31, pp. S11–S33, 2000.

[7] C. Larsson, K. Westerterp, and G. Johansson, “Validity of reported energy expenditure and energy and protein intakes in Swedish adolescent vegans and omnivores,” *American Journal of Clinical Nutrition*, vol. 75, pp. 268–274, 2002.

[8] L. Bandini, A. Must, H. Cyr, S. Anderson, J. Spadano, and W. Dietz, “Longitudinal changes in the accuracy of reported energy intake in girls 10-15 y of age,” *American Journal of Clinical Nutrition*, vol. 78, pp. 480–484, 2003.

[9] R. Klesges, L. Eck, and J. Ray, “Who underreports dietary intake in a dietary recall? Evidence from the Second National Health and Nutrition Examination Survey,” *Journal of Consulting and Clinical Psychology*, vol. 63, pp. 438–444, 1995.

[10] R. Johnson, R. Soultanakis, and D. Matthews, “Literacy and body fatness are associated with underreporting of energy intake in US low-income women using the multiple-pass 24-hour recall: a doubly labeled water study,” *Journal of the American Dietetic Association*, vol. 98, pp. 1136–1140, 1998.

[11] J. Tooze, A. Subar, F. Thompson, R. Troiano, A. Schatzkin, and V. Kipnis, "Psychosocial predictors of energy underreporting in a large doubly labeled water study," *American Journal of Clinical Nutrition*, vol. 79, pp. 795–804, 2004.

[12] G. Bathalon, K. Tucker, N. Hays, A. Vinken, A. Greenberg, M. McCrory, and S. Roberts, "Psychological measures of eating behavior and the accuracy of 3 common dietary assessment methods in healthy postmenopausal women," *American Journal of Clinical Nutrition*, vol. 71, pp. 739–745, 2000.

[13] A. Sawaya, K. Tucker, R. Tsay, W. Willett, E. Saltzman, G. Dallal, and S. Roberts, "Evaluation of four methods for determining energy intake in young and older women: comparison with doubly labeled water measurements of total energy expenditure," *American Journal of Clinical Nutrition*, vol. 63, pp. 491–499, 1996.

[14] L. Harnack, L. Steffen, D. Arnett, S. Gao, and R. Luepker, "Accuracy of estimation of large food portions," *Journal of the American Dietetic Association*, vol. 104, pp. 804–806, 2004.

[15] S. Nielsen and B. Popkin, "Patterns and trends in food portion sizes, 1977-1998," *Journal of the American Medical Association*, vol. 289, pp. 450–453, 2003.

[16] L. Young and M. Nestle, "The contribution of expanding portion sizes to the us obesity epidemic," *American Journal of Public Health*, vol. 92, pp. 246–249, 2002.

[17] S. Baxter, W. Thompson, M. Litaker, F. Frye, and C. Guinn, "Low accuracy and low consistency of fourth-graders' school breakfast and school lunch recalls," *Journal of the American Dietetic Association*, vol. 102, pp. 386–395, 2002.

[18] S. Rebro, R. Patterson, A. Kristal, and C. Cheney, "The effect of keeping food records on eating patterns," *Journal of the American Dietetic Association*, vol. 98, pp. 1163–1165, 1998.

[19] J. Trabulsi and D. Schoeller, "Evaluation of dietary assessment instruments against doubly labeled water, a biomarker of habitual energy intake," *American Journal of Physiology - Endocrinology And Metabolism*, vol. 281, pp. E891–E899, 2001.

[20] M. Livingstone and A. Black, "Validation of estimates of energy intake by weighed dietary record and diet history in children and adolescents," *Journal of Nutrition*, vol. 133, p. S895, 2003.

[21] J. Weber, L. Cunningham-Sabo, B. Skipper, L. Lytle, J. Stevens, J. Gittelsohn, J. Anliker, K. Heller, and J. Pablo, "Portion-size estimation training in second and third-grade American Indian children," *American Journal of Clinical Nutrition*, vol. 69, pp. 782S–787S, 1999.

[22] T. Schap, B. Six, E. Delp, D. Ebert, D. Kerr, and C. 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. 14, pp. 1–8, 2011.

[23] B. Six, T. Schap, F. Zhu, A. Mariappan, M. Bosch, E. Delp, D. Ebert, D. Kerr, and C. Boushey, "Evidence-based development of a mobile telephone food record," *Journal of American Dietetic Association*, pp. 74–79, January 2010.

[24] C. Boushey, D. Kerr, J. Wright, K. Lutes, D. Ebert, and E. Delp, "Use of technology in children's dietary assessment," *European Journal of Clinical Nutrition*, pp. S50–S57, 2009.

[25] F. Zhu, A. Mariappan, D. Kerr, C. Boushey, K. Lutes, D. Ebert, and E. Delp, "Technology-assisted dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging VI*, vol. 6814, San Jose, USA, January 2008.

[26] 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, USA, January 2009.

[27] 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*, San Jose, CA, January 2010.

[28] J. Chae, I. Woo, S. Kim, R. Maciejewski, F. Zhu, E. Delp, C. Boushey, and D. Ebert, "Volume estimation using food specific shape templates in mobile image-based dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, vol. 7873, San Francisco, USA, January 2011.

[29] "USDA food and nutrient database for dietary studies, 1.0." Beltsville, MD: Agricultural Research Service, Food Surveys Research Group, 2004.

[30] A. Jimenez, A. Jain, R. Ceres, and J. Pons, "Automatic fruit recognition: a survey and new results using range/attenuation images," *Pattern Recognition*, vol. 32, no. 10, pp. 1719–1736, 1999.

[31] D. Mery and F. Pedreschi, "Segmentation of colour food images using a robust algorithm," *Journal of Food Engineering*, vol. 66, no. 3, pp. 353–360, February 2005.

[32] D. Sun and C. Du, "Segmentation of complex food images by stick growing and merging algorithm," *Journal of Food Engineering*, vol. 61, no. 1, pp. 17–26, January 2004.

[33] S. Yang, M. Chen, D. Pomerleau, and R. Sukhankar, "Food recognition using statistics of pairwise local features," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 2010, pp. 2249–2256.

[34] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," *Proceedings of the International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 285–288.

[35] S. Arivazhagan, R. Newlin Shebiah, S. Selva Nidhyanandhan, and L. Ganesan, "Fruit recognition using color and texture features," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 1, no. 2, pp. 90–94, October 2010.

[36] K. Kitamura, T. Yamasaki, and K. Aizawa, "Foodlog: Capture, analysis and retrieval of personal food images via web," *Proceedings of the ACM Multimedia Workshop on Multimedia for Cooking and Eating Activities*, Beijing, China, November 2009, pp. 23–30.

[37] F. Zhu, M. Bosch, T. Schap, N. Khanna, D. Ebert, C. Boushey, , and E. Delp, "Segmentation assisted food classification for dietary assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, vol. 7873, San Francisco, USA, January 2011.

[38] F. Zhu, M. Bosch, and E. Delp, "An image analysis system for dietary assessment and evaluation," *Proceedings of the International Conference on Image Processing*, Hong Kong, China, September 2010, pp. 1853–1856.

[39] 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.

[40] M. Bosch, F. Zhu, N. Khanna, C. Boushey, and E. Delp, "Food texture descriptors based on fractal and local gradient information," *Proceedings of the 19th European Signal Processing Conference*, Barcelona, Spain, September 2011.

[41] M. Bosch, "Visual feature modeling and refinement with application in dietary assessment," Ph.D. dissertation, Purdue University, 2012.

[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, 2011.

[43] K. Ostmo, "Automatic portion estimation and visual refinement in automatic portion estimation and visual refinement in mobile dietary assessment," Master's thesis, Purdue University, 2009.

[44] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, Inc., 2008.

[45] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, Sep/Oct 2001.

[46] Y. Chang and J. Reid, "Rgb calibration for color image analysis in machine vision," *IEEE Transactions on Image Processing*, vol. 5, no. 10, pp. 1414–1422, 1996.

[47] H. Siddiqui and C. Bouman, "Hierarchical color correction for camera cell phone images," *IEEE Transactions on Image Processing*, vol. 17, no. 11, 2008.

[48] S.-H. Lee, T.-Y. Kim, and J.-S. Choi, "A color correction system using a color compensation chart," *Proceedings of the International Conference on Hybrid Information Technology*, Washington, DC, USA, November 2006, pp. 409–416.

[49] Y. Choi, Y. Lee, and W. Cho, "Color correction for object identification from images with different color illumination," *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC*, August 2009, pp. 1598–1603.

[50] S. Srivastava, T. Ha, J. Allebach, and E. Delp, "Color management using optimal three-dimensional look-up tables," *Journal of Imaging Science and Technology*, vol. 54, no. 3, May-June 2010.

[51] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," *Proceedings of the 23rd ACM national conference*, 1968, pp. 517–524.

[52] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[53] B. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[54] P. Kruizinga, N. Petkov, and S. E. Grigorescu, "Comparison of texture features based on gabor filters," *Proceedings of the 10th International Conference on Image Analysis and Processing*, Washington, DC, USA, September 1999, p. 142.

[55] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, April 2002.

[56] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* San Diego, Ca: Academic Press, 1990.

[57] R. Duta, P. Hart, and D. Stork, *Pattern Classification.* New York, NY: Wiley-Interscience, 2000.

[58] N. Cristianini and J. Taylor, *An Introduction to Support Vector Machines.* Cambridge: Cambridge University Press, 2000.

[59] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[60] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, March 2001.

[61] E. Chong and S. Zak, *An Introduction to Optimization*, 2nd ed. John Wiley and Sons, New York, 2001.

[62] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[63] R. Haralick and L. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[64] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[65] N. Pal and S. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[66] P. K. Sahoo, S. Soltani, A. K. Wong, and Y. C. Chen, “A survey of thresholding techniques,” *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 2, pp. 233–260, February 1988.

[67] E. Riseman and M. Arbib, “Computational techniques in the visual segmentation of static scenes,” *Computer Graphics and Image Processing*, vol. 6, no. 3, pp. 221–276, June 1977.

[68] W. Skarbek and A. Koschan, “Colour image segmentation - a survey,” Technical University of Berlin, Tech. Rep., 1994.

[69] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, “Adaptive perceptual color-texture image segmentation,” *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1524–1536, October 2005.

[70] D. Comaniciu and P. Meer, “Robust analysis of feature spaces: Color image segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, 1997, pp. 750–755.

[71] R. Gonzalez, R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB.* Prentice Hall Upper Saddle River, NJ, 2004.

[72] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1998.

[73] S. Osher and J. Sethian, “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[74] T. Chan and L. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[75] H. Zhao, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to multiphase motion,” *Journal of Computational Physics*, vol. 127, no. 1, pp. 179–195, 1996.

[76] L. Vese and T. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[77] J. Sethian *et al.*, *Level Set Methods And Fast Marching Methods.* Cambridge university press Cambridge, 1999.

[78] V. Caselles, F. Catté, T. Coll, and F. Dibos, “A geometric model for active contours in image processing,” *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993.

[79] R. Malladi, J. Sethian, B. Vemuri, *et al.*, “Shape modeling with front propagation: A level set approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, 1995.

[80] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[81] A. Yezzi Jr, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 199–209, 1997.

[82] G. Sapiro, "Vector-valued active contours," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1996, pp. 680–685.

[83] ——, "Color snakes," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 247–253, 1997.

[84] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[85] A. Tsai, A. Yezzi, and A. Willsky, "Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1169–1186, 2001.

[86] T. Chan, B. Sandberg, and L. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.

[87] C. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. 100, no. 20, pp. 68–86, 1971.

[88] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[89] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1101–1113, 1993.

[90] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in nd images," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, Vancouver, BC , Canada, July 2001, pp. 105–112.

[91] F. Estrada, A. Jepson, and C. Chennubhotla, "Spectral embedding and min-cut for image segmentation," *Proceedings of the British Machine Vision Conference*, 2004, pp. 7–9.

[92] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[93] A. Rabinovich, S. Belongie, T. Lange, and J. Buhmann, "Model order selection and cue combination for image segmentation," *Proceedgins of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2006, pp. 1130–1137.

[94] A. S. Incorp., *Adobe Photoshop CS5 User Guide*, 2011.

[95] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 1995, pp. 191–198.

[96] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, "A bayesian approach to digital matting," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2001, pp. 264–271.

[97] M. Ruzon and C. Tomasi, "Alpha estimation in natural images," *Proceedings of Computer Vision and Pattern Recognition*, vol. 1, June 2000, pp. 18–25.

[98] C. Corporation, "Knockout user guide," 2002.

[99] V. Vezhnevets and V. Konouchine, "Growcut: Interactive multi-label nd image segmentation by cellular automata," *Proceedings of Graphicon*, 2005, pp. 150–156.

[100] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[101] G. Friedland, K. Jantz, and R. Rojas, "Siox: Simple interactive object extraction in still images," *Proceedings of the Seventh IEEE International Symposium on Multimedia*, Dec. 2005, p. 7.

[102] P. Mattis, S. Kimball, and M. Singh, "Gnu image manipulation program," *URL: http://www. gimp. org.*

[103] T. I. Team, "www.inkscape.org."

[104] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal on Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[105] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3241–3248.

[106] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," *Proceedings of the Tenth IEEE International Conference on Computer Vision*, vol. 1, Oct. 2005, pp. 654–661.

[107] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," *Proceedings of the Tenth IEEE International Conference on Computer Vision*, vol. 1, Oct. 2005, pp. 370– 377.

[108] P. Kovesi, "http://www.csse.uwa.edu.au/ pk/research/matlabfns/."

[109] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[110] E. Sharon, A. Brandt, and R. Basri, "Fast multiscale image segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, vol. 1, June 2000, pp. 70–77.

[111] S. Yu, "Segmentation using multiscale cues," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 27 June - 2 July 2004, pp. 247–254.

[112] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2005, pp. 1124–1131.

[113] B. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface.* Wiley and Sons, USA, 2002.

[114] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications.* Wiley, England, 1990.

[115] J. Vehel, P. Mignot, and J. Merriot, "Multifractals, texture, and image analysis," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1992, pp. 661–664.

[116] N. Sarkar and B. Chaudhuri, "An efficient differential box-counting approach to compute fractal dimension of images," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 2, pp. 115 – 120, January 1994.

[117] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.

[118] W. Freeman and Y. Adelson, "The design and use of steerable filters," *IEEE Transactions on System, Man, and Cybernetics*, pp. 460 – 473, 1978.

[119] V. Vapnik, "The nature of statistical learning theory," *Springer-Verlag*, New York, NY, 1995.

[120] Y. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335–1346, August 1996.

[121] D. Martin, C. Fowlkers, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, July 2001, pp. 416–423.

[122] D. Martin and C. Fowlkers. The berkeley segmentation database and benchmark. `http://www.cs.berkeley.edu/projects/vision/grouping/segbench/`

[123] D. Martin, "An empirical approach to grouping and segmentation," Ph.D. dissertation, University of California, Berkeley, 2002.

[124] A. Goldberg. Csa: an efficient implementation of a scaling push-relabel algorithm for the assignment problem. `http://www.avglab.com/andrew/soft.html`

[125] F. Estrada and A. Jepson, "Benchmarking image segmentation algorithms," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 167–181, 2009.

[126] S. Borra and S. Sarkar, "A framework for performance characterization of intermediate-levelgrouping modules," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1306–1312, November 1997.

[127] A. Rabinovich, A. Vedaldi, and S. Belongie, “Does image segmentation improve object categorization,” Technical report, UCSD CSE Dept., 2007. 2, Tech. Rep.

[128] M. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454.

[129] M. Rousson and R. Deriche, “A variational framework for active and adaptative segmentation of vector valued images,” *Proceedings of the Workshop on Motion and Video Computing*, December 2002, pp. 56–61.

[130] C. Lee, W. Snyder, and C. Wang, “Supervised multispectral image segmentation using active contours,” *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 4242–4247.

VITA

VITA

Fengqing Zhu was born in Suzhou, Jiangsu Province, China. She received her Bachelor of Science degree in Electrical Engineering (with Distinction) and Master of Science in Electrical and Computer Engineering from Purdue University, West Lafayette, Indiana in 2004 and 2006.

Fengqing joined the Ph.D. program at Purdue University, West Lafayette, Indiana in 2007. Since 2005, she has served as Research Assistant at the Video and Image Processing Laboratory (VIPER). Her major advisor Professor Edward J. Delp is the Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering. While in the graduate program, Fengqing has worked on projects sponsored by grants from the National Institutes of Health and the Nokia Research Center. During the summer of 2007, she was a Student Intern at the Sharp Laboratories of America, Camas, Washington. Her research interests include video compression, image/video processing, image analysis, and computer vision.

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

Fengqing Zhu's publications from this research work include:

Journal Papers:

1. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, "Segmentation Based Video Compression Using Texture and Motion Models, *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, November 2011, pp. 1366-1377.
2. **Fengqing Zhu**, Marc Bosch, Insoo Woo, SungYe Kim, Carol J. Boushey, David S. Ebert, and Edward 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, August 2010, pp. 756-766.
3. Bethany L. Six, TusaRebecca E. Schap, **Fengqing Zhu**, Anand Mariappan, Marc Bosch, Edward J. Delp, David S. Ebert, Deborah A. Kerr, and Carol J. Boushey, "Evidence-Based Development of a Mobile Telephone Food Record," *Journal of American Dietetic Association*, January 2010, pp. 74-79.
4. TusaRebecca E Schap, **Fengqing Zhu**, Edward J. Delp, and Carol J. Boushey, "Merging Dietary Assessment with the Adolescent Lifestyle," *Journal of Human Nutrition and Dietetics*, submitted.

Conference Papers:

1. **Fengqing Zhu**, Marc Bosch, Ziad Ahmad, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Challenges in Using a Mobile Device Food Record Among Adults in Free-living Situations," *mHealth Summit*, Washington DC, December 2011.
2. **Fengqing Zhu**, Marc Bosch, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Multilevel Segmentation for Food Classification in Dietary Assessment," *Proceedings of* 7th *International Symposium on Image and Signal Processing and Analysis*, Dubrovnik, Croatia, September 4-6, 2011, pp. 337-342.
3. Meilin Yang, Ye He, **Fengqing Zhu**, Marc Bosch, Mary Comer, and Edward J. Delp, "Video Coding: Death Is Not Near," *Proceedings of the* 53rd *International Symposium ELMAR*, Zadar, Croatia, September 2011. (Invited Paper).

4. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Combining Global and Local Features for Food Identification and Dietary Assessment," *Proceedings of the International Conference on Image Processing*, Brussels, Belgium, September 2011.

5. Marc Bosch, **Fengqing Zhu**, Nitin Khanna, Carol J. Boushey and Edward J. Delp, "Food Texture Descriptors Based on Fractal and Local Gradient Information," *Proceedings of the* 19th *European Signal Processing Conference*, Barcelona, Spain, September 2011.

6. Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, **Fengqing Zhu**, Carol J. Boushey and Edward J. Delp, "Integrated Databases System for Mobile Dietary Assessment and Analysis," *Proceedings of the* 1st *IEEE International Workshop on Multimedia Services and Technologies for E-health in conjunction with the International Conference on Multimedia and Expo*, Barcelona, Spain, July 2011.

7. **Fengqing Zhu**, Marc Bosch, TusaRebecca E. Schap, Nitin Khanna, David S. Ebert, Carol J. Boushey and Edward J. Delp, "Segmentation Assisted Food Classification for Dietary Assessment," *Proceedings of the IS&T/SPIE Conference on Computational Imaging IX*, Vol. 7873, Burlingame, California, January 2011.

8. JungHoon Chae, Insoo Woo, SungYe Kim, Ross Maciejewski, **Fengqing Zhu**, Edward J. Delp, Carol J. Boushey, and David 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, Burlingame, California, January 2011.

9. **Fengqing Zhu**, Marc Bosch, Carol J. Boushey and Edward J. Delp, "An Image Analysis System for Dietary Assessment and Evaluation," *Proceedings of the International Conference on Image Processing*, Hong Kong, China, September, 2010.

10. Nitin Khanna, **Fengqing Zhu**, Marc Bosch, Meilin Yang, Mary Comer, and Edward J. Delp, “Information Theory Inspired Video Coding Methods: Truth Is Sometimes Better Than Fiction,” *Proceedings of the Third Workshop on Information Theoretic Methods in Science and Engineering*, Tampere, Finland, August 2010.

11. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, “Perceptual quality evaluation for texture and motion based video coding,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, November 2009.

12. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, “An Overview of Texture and Motion based Video Coding at Purdue University,” *Proceedings of the 27th Picture Coding Symposium (PCS)*, Chicago, USA, May 2009.

13. Anand Mariappan, Marc Bosch, **Fengqing Zhu**, Carol J. Boushey, David S. Ebert, Deborah A. Kerr, and Edward J. Delp, “Personal Dietary Assessment Using Mobile Devices,” *Proceedings of the IS&T/SPIE Conference on Computational Imaging VII*, Vol. 7246, San Jose, California, January 2009.

14. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, “Video Coding Using Motion Classification,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, San Diego, USA, October 2008.

15. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, “Models for texture based video coding,” *Proceedings of LNLA, IEEE International Workshop on Local and Non-Local Approximation in Image Processing*, Lausanne, Switzerland, August 2008.

16. **Fengqing Zhu**, Anand Mariappan, Carol J. Boushey, Deborah A. Kerr, Kyle Lutes, David S. Ebert, and Edward J. Delp, “Technology-Assisted Dietary Assessment,” *Proceedings of the IS&T/SPIE Conference on Computational Imaging VI*, Vol. 6814, San Jose, California, January 2008.

17. Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, "Spatial Texture Models for Video Compression," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, San Antonio, USA, September 2007.

18. Limin Liu, Marc Bosch, **Fengqing Zhu**, and Edward J. Delp, "Recent advances in video compression: what's next?," *Proceedings of the IEEE International Symposium on Signal Processing and its Applications (ISSPA 2007)*, Sharjah, United Arab Emirates, February 2007 (Plenary Paper).

19. **Fengqing Zhu**, Ka Ki Ng, Golnaz Abdollahian, and Edward J. Delp, "Spatial and Temporal Models for Texture-Based Video Coding," *Proceedings of the SPIE International Conference on Video Communications and Image Processing (VCIP 2007)*, San Jose, USA, January 2007.