



Michigan Technological University  
*Create the Future* Digital Commons @ Michigan Tech

---

Dissertations, Master's Theses and Master's  
Reports - Open

Dissertations, Master's Theses and Master's  
Reports

---

2012

## Automatic Food Intake Assessment Using Camera Phones

Fanyu Kong  
*Michigan Technological University*

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

Copyright 2012 Fanyu Kong

---

### Recommended Citation

Kong, Fanyu, "Automatic Food Intake Assessment Using Camera Phones", Dissertation, Michigan Technological University, 2012.  
<https://doi.org/10.37099/mtu.dc.etds/494>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

AUTOMATIC FOOD INTAKE ASSESSMENT USING CAMERA PHONES

By

Fanyu Kong

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

(Computer Engineering)

MICHIGAN TECHNOLOGICAL UNIVERSITY

2012

© 2012 Fanyu Kong



This dissertation, "Automatic Food Intake Assessment Using Camera Phones," is hereby approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING.

Department of Electrical and Computer Engineering

Signatures:

Dissertation Advisor \_\_\_\_\_  
Dr. Jindong Tan

Department Chair \_\_\_\_\_  
Dr. Daniel R. Fuhrmann

Date \_\_\_\_\_





## **Dedication**

To my wife and my parents!



# Contents

<b>List of Figures</b>	.xviii
<b>List of Tables</b>	xix
<b>Preface</b>	xxi
<b>Acknowledgments</b>	.xxiii
<b>Abstract</b>	xxv
<b>1 Introduction</b>	1
1.1 Obesity: A challenge to public health	1
1.2 Food intake assessment	3
1.3 Opportunity and challenges	4
1.4 Goals and specific aims	5
1.5 Organization of this thesis	6
Reference	7

<b>2</b>	<b>DietCam: Automatic Dietary Assessment with Mobile Camera Phones . . . .</b>	<b>11</b>
	Abstract . . . . .	11
2.1	Introduction . . . . .	11
2.2	System Architecture . . . . .	14
2.3	Food Classification . . . . .	17
2.3.1	Food Features . . . . .	17
2.3.2	Food Recognition . . . . .	18
2.3.3	Food Segmentation . . . . .	20
2.3.4	Occlusions and Redundancies . . . . .	20
2.4	Volume and Calorie Estimation . . . . .	21
2.4.1	Camera Calibration . . . . .	22
2.4.1.1	Intrinsic Parameters Calibration . . . . .	22
2.4.1.2	Extrinsic Parameters Calibration . . . . .	23
2.4.2	3D Model Reconstruction . . . . .	24
2.4.3	Volume Calculation . . . . .	25
2.4.4	Calorie Estimation . . . . .	26
2.5	Food Database . . . . .	26
2.6	Client-Server Architecture . . . . .	27
2.7	Evaluation . . . . .	28

2.7.1	Experimental Setup . . . . .	28
2.7.2	Implementation . . . . .	28
2.7.3	Recognition Accuracy . . . . .	30
2.7.4	Volume Calculation . . . . .	32
2.7.5	Unrecognized food and residues . . . . .	34
2.8	Discussion . . . . .	34
2.9	Related Work . . . . .	35
2.10	Conclusion and Future Work . . . . .	37
	Reference . . . . .	37
<b>3</b>	<b>DietCam: Multi-View Regular Shape Food Recognition with a Camera Phone</b>	<b>43</b>
	Abstract . . . . .	43
3.1	Introduction . . . . .	44
3.2	Related Work . . . . .	47
3.3	Multi-View Food Recognition . . . . .	49
3.3.1	Food features . . . . .	50
3.3.2	Camera calibration . . . . .	51
3.3.3	Perspective distance . . . . .	53
3.3.4	Multi-view food recognition . . . . .	54
3.3.5	Object recognition from 3D reconstruction . . . . .	58

3.4	Implementation . . . . .	58
3.5	Experiment . . . . .	59
3.5.1	Dataset . . . . .	59
3.5.2	Baseline methods . . . . .	61
3.5.3	Segmentation results . . . . .	62
3.5.4	Classification results . . . . .	62
3.6	Discussion . . . . .	64
3.7	Conclusion . . . . .	66
	Reference . . . . .	66
<b>4</b>	<b>DietCam: Multi-view, Multi-class Food Recognition Using a Multi-kernel Based SVM . . . . .</b>	<b>71</b>
	Abstract . . . . .	71
4.1	Introduction . . . . .	71
4.2	Related Work . . . . .	75
4.3	Ingredient detection . . . . .	76
4.4	Food classification . . . . .	80
4.4.1	Multiple viewpoints . . . . .	83
4.4.2	Multi-kernel . . . . .	85
4.5	Dataset . . . . .	86

4.6	Experiment . . . . .	87
4.6.1	Ingredient detection . . . . .	87
4.6.2	Food classification . . . . .	91
4.7	Discussion . . . . .	94
4.8	Conclusion . . . . .	96
	Reference . . . . .	96
<b>5</b>	<b>Indexing of Bags of Features: Efficient Image Retrieval from Large-Scale Database . . . . .</b>	<b>103</b>
	Abstract . . . . .	103
5.1	Introduction . . . . .	104
5.2	Related Work . . . . .	106
5.3	Short Representations of Images . . . . .	107
5.3.1	The average of SIFT features . . . . .	108
5.3.2	The weighted average of SIFT features . . . . .	108
5.3.3	PCA of SIFT features . . . . .	109
5.3.4	Image distinctiveness of SRs . . . . .	110
5.4	Binary index of short representations . . . . .	111
5.4.1	Locality similarity hashing . . . . .	112
5.4.2	Spatial hashing . . . . .	113



5.5	Food Database . . . . .	114
5.6	Experiment . . . . .	115
5.6.1	Short representation . . . . .	115
5.6.2	Image indexing . . . . .	117
5.7	Conclusion . . . . .	119
	Reference . . . . .	120
<b>6</b>	<b>DietVolume: Food Volume Estimation through Metric 3D Reconstruction on a Mobile Phone . . . . .</b>	<b>125</b>
	Abstract . . . . .	125
6.1	Introduction . . . . .	126
6.2	Related work . . . . .	129
6.2.1	Structure from motion . . . . .	129
6.2.2	Visual odometry . . . . .	130
6.2.3	Egomotion . . . . .	130
6.2.4	Image based food volume estimation . . . . .	131
6.3	Problem Formulation . . . . .	132
6.3.1	3D model reconstruction . . . . .	132
6.3.2	Scale estimation . . . . .	133
6.3.3	Volume calculation . . . . .	135

6.4	3D model reconstruction . . . . .	135
6.5	Scale factor estimation . . . . .	137
6.6	Volume Calculation . . . . .	141
6.7	Implementation and Visualization . . . . .	142
6.8	Experiment . . . . .	143
6.9	Conclusion . . . . .	145
	Reference . . . . .	146
<b>7</b>	<b>Conclusion . . . . .</b>	<b>151</b>
7.1	Conclusions . . . . .	151
7.2	Summary of contributions . . . . .	153
7.3	Future works . . . . .	153
	<b>References . . . . .</b>	<b>167</b>



# List of Figures

1.1	The goal of this thesis . . . . .	6
1.2	Outline of the thesis scientific contributions. . . . .	8
2.1	Expected system usage. . . . .	12
2.2	System architecture. . . . .	15
2.3	Training set examples. . . . .	18
2.4	Feature matching based food segmentation. . . . .	19
2.5	Camera intrinsic parameter calibration. . . . .	22
2.6	3D model reconstruction. . . . .	24
2.7	Tetrahedrons of an arbitrary shaped food item. . . . .	25
2.8	Calorie calculation function. . . . .	29
2.9	The segmentation accuracy. . . . .	30
2.10	Food item classification accuracy. . . . .	31
2.11	Food types used to test the volume estimation. . . . .	32
2.12	A sandwich in the experiment. . . . .	32

2.13	Average absolute deviations with increased number of food item . . . . .	32
2.14	Calorie monitoring with OCR . . . . .	34
2.15	Food residue recognition. . . . .	35
3.1	Food shapes. . . . .	44
3.2	Image perspectives. . . . .	46
3.3	Training sets from different perspectives. . . . .	51
3.4	Vocabulary tree structure. . . . .	52
3.5	Perspective distance. . . . .	53
3.6	Bayesian classifiers. . . . .	56
3.7	Multi-view image relations. . . . .	59
3.8	The client-server architecture of DietCam. . . . .	60
3.9	Classification accuracy of a single food item. . . . .	61
3.10	Classification accuracy of fruits. . . . .	63
3.11	Classification accuracy of fruits and fast food. . . . .	64
3.12	Classification accuracy of fruit, fast food, and home made food. . . . .	64
3.13	Classification accuracy of home made foods. . . . .	65
4.1	Food appearances. . . . .	72
4.2	Histogram of oriented gradients descriptor of chicken wings. . . . .	78

4.3	Extracted food ingredient textures. . . . .	79
4.4	The part location model of chicken wings. . . . .	80
4.5	Geometric similarity between two viewpoints. . . . .	84
4.6	Precision-Recall curve for models with difficulty 1 . . . . .	89
4.7	Precision-Recall curve for models with difficulty 2. . . . .	89
4.8	Precision-Recall curve for models with difficulty 3. . . . .	90
4.9	Precision-Recall curve for models with difficulty 4. . . . .	91
4.10	Precision-Recall curve for models with difficulty 5. . . . .	92
4.11	Precision-Recall curve for models with difficulty 6. . . . .	93
4.12	Recognition accuracy comparison between DieCam and SIFT. . . . .	93
4.13	Recognition accuracy comparison between DieCam and texture classifier. .	94
4.14	Recognition accuracy comparison between DieCam and single view classifier.	94
4.15	Unrecognized food images. . . . .	95
5.1	The hierarchical structure of the index. . . . .	104
5.2	Using PCA to reveal the internal property of a set of points. . . . .	109
5.3	One example of bit assignment. . . . .	112
5.4	Distances between SRs vs. similarity between images. . . . .	116
5.5	Image retrieval time evaluation. . . . .	118
5.6	Image retrieval precision evaluation. . . . .	119

6.1	Volume estimation of an apple. . . . .	127
6.2	3D reconstruction and scale recovery from videos. . . . .	131
6.3	Intrinsic Parameters Calibration. . . . .	133
6.4	Three coordinate systems. . . . .	138
6.5	Tetrahedrons of an arbitrary shaped food item. . . . .	141
6.6	DietVolume architecture. . . . .	141
6.7	3D object dimension measurement on iPhone. . . . .	143
6.8	Coins diameter and box dimension measurements. . . . .	144
6.9	Orange volume estimation. . . . .	145
6.10	Number of frames the scale factor needs to converge. . . . .	145

# List of Tables

1.1	Publications included in the thesis. . . . .	7
2.1	Calorie Density Chart. . . . .	27
2.2	Similar food classification accuracy . . . . .	33
2.3	Measured and estimated food volumes . . . . .	34
3.1	Food segmentation. . . . .	60
4.1	Food Difficulty Statistics. . . . .	87
4.2	Food Database Statistics. . . . .	97
5.1	Maximum and minimum numbers of images per index. . . . .	117





## **Preface**

This dissertation consists of my five journal publications, from Chapter 2 to Chapter 6. All of these publications are co-authored by my advisor Dr. Jindong Tan. In these publications, I provided methodology investigation, experiments, and writing. Dr. Jindong Tan provided invaluable revisions.



## Acknowledgments

I am sincerely thankful to my advisor Dr. Jindong Tan, together with whom we come of the research idea on food intake assessment. This work would not have been possible without the facilities in the Robotics and Wireless Sensor Networks lab at Michigan Technological University, which he leads.

I am particularly grateful to Dr. Chunxiao Chigan, Dr. Michael C. Roggemann, and Dr. Nilufer Onder, who accepted my research proposal and encouraged me in my work.

My gratitude goes to Dr. Timothy J. Schulz, Dr. Nilufer Onder and also to Dr. Michael C. Roggemann, for serving as my committees and co-examiners of the thesis.

I wish to thank all colleagues at the Robotics and Wireless Sensor Networks lab, all of them influenced my research: Sheng Hu, Lufeng Shi, Shuo Huang, Xi Chen, Ya Tian, Zhenzhou Shao, and Xiaolong Liu.

Moreover, my gratitude goes to my former colleague, Huaming Li, for his support when I arrived at Michigan and started my research.



## Abstract

Obesity is becoming an epidemic phenomenon in most developed countries. The fundamental cause of obesity and overweight is an energy imbalance between calories consumed and calories expended. It is essential to monitor everyday food intake for obesity prevention and management. Existing dietary assessment methods usually require manually recording and recall of food types and portions. Accuracy of the results largely relies on many uncertain factors such as user's memory, food knowledge, and portion estimations. As a result, the accuracy is often compromised. Accurate and convenient dietary assessment methods are still blank and needed in both population and research societies.

In this thesis, an automatic food intake assessment method using cameras, inertial measurement units (IMUs) on smart phones was developed to help people foster a healthy life style. With this method, users use their smart phones before and after a meal to capture images or videos around the meal. The smart phone will recognize food items and calculate the volume of the food consumed and provide the results to users. The technical objective is to explore the feasibility of image based food recognition and image based volume estimation.

This thesis comprises five publications that address four specific goals of this work: (1) to develop a prototype system with existing methods to review the literature methods, find their drawbacks and explore the feasibility to develop novel methods; (2) based on the prototype system, to investigate new food classification methods to improve the recognition accuracy to a field application level; (3) to design indexing methods for large-scale image database to facilitate the development of new food image recognition and retrieval algorithms; (4) to develop novel convenient and accurate food volume estimation methods using only smart phones with cameras and IMUs.

A prototype system was implemented to review existing methods. Image feature detector and descriptor were developed and a nearest neighbor classifier were implemented to classify food items. A reedit card marker method was introduced for metric scale 3D reconstruction and volume calculation.

To increase recognition accuracy, novel multi-view food recognition algorithms were developed to recognize regular shape food items. To further increase the accuracy and make the algorithm applicable to arbitrary food items, new food features, new classifiers were designed. The efficiency of the algorithm was increased by means of developing novel image indexing method in large-scale image database. Finally, the volume calculation was enhanced through reducing the marker and introducing IMUs. Sensor fusion technique to combine measurements from cameras and IMUs were explored to infer the metric scale of the 3D model as well as reduce noises from these sensors.



# Chapter 1

## Introduction

### 1.1 Obesity: A challenge to public health

Due to its drastic increase during the past decade, obesity is becoming an epidemic phenomenon in most developed countries. Overweight and obesity are defined as abnormal or excessive fat accumulation in body and it could impair health. Body mass index (BMI) is a measure of body fat based on height and weight that applies to adults. According to the fact sheet of March 2011 from World Health Organization (WHO), the worldwide obesity has more than doubled since 1980 [1]. In 2008, 1.5 billion adults were considered overweight ( $BMI \geq 25kg/m^2$ ). Of these over 200 million men and nearly 300 million women were obese ( $BMI \geq 30kg/m^2$ ). Further, in 2010, around 43 million children under five were overweight and the percentages are rising[1].

In the past three decades, obesity rates for both adults and children in the U.S. have increased significantly[2]. The fact that more than thirty-three percent of adults and sixteen percent of children are obese has proven to be one of the biggest public health challenges to the general population and social welfare. But obesity has still not been controlled effectively. In 2000, no state had an obesity prevalence of 30% or more. But in 2010, the number of states with an obesity prevalence of 30% or more has increased to 12 states [3]. No state has met the nation's Healthy People 2010 [3] goal to lower obesity prevalence to 15%. If this trend in obesity continues, the majority of the U.S. population could be overweight even obese in a few generations.

This brings us an alarming message because overweight and obesity are linked to leading causes of death. Overweight and obesity are the fifth leading risk for global death. Every



year, at least 2.8 million adults die as a result of being overweight and obese [1]. Besides, serious consequences of obesity also include severe health problems such as diabetes, stroke, and heart disease. According to the WHO statistics, about 44% of the diabetes burden, 23% of the ischaemic heart disease burden and between 7% and 41% of certain cancer burdens are attributable to overweight and obesity [1]. Compared with underweight, overweight and obesity cause more deaths worldwide.

Moreover, overweight and obesity also cause high-cost healthcare bills, which were estimated at \$147 billion in 2008 in the U.S. alone [4, 5]. In addition, the medical costs paid by third-party payers for people who are obese were \$1,429 higher than those of normal weight [1]. The continuing increase of overweight and obesity attributable death and spending have attracted increasing research interests to explore practical new technologies to control and prevent obesity.

In spite of the common sense that obesity is a complex condition caused by the interaction of many factors such as genetic makeup, secondary effects from medical treatment, neuroendocrine disorders, and emotions, the fundamental cause of obesity and overweight is an energy imbalance between calories consumed and calories expended. Most Americans have a high caloric intake of energy-dense foods that are high in fat, salt and sugars but low in vitamins, minerals and other micronutrients; and a low level in physical activity due to the increasingly sedentary nature of many forms of work, changing modes of transportation, and increasing urbanization. Increased caloric intake may have resulted from the easy access to high calorie fast food and soft beverage, pervasive advertisement of the fast food industry, the expanding baggage sizes of snacks and meals, and the leisure sitting and eating life styles. Nowadays people spend more time on television watching, video game playing and web surfing, while eating high caloric snacks.

It is generally believed that obesity prevention requires individuals to foster life-long healthy food choices and regular physical activities [6]. However, the usual case is that individuals with potential obesity problems are more likely to ignore their food intakes and regular exercise. Even people who care and pay attention to nutrition information may not be sufficiently knowledgeable about the calorie content of what they are eating. Monitoring eating behaviors is the prerequisite for individual obesity prevention and management as well as for research on disease intervention. However, few people are willing to use the current food intake assessment methods.

## 1.2 Food intake assessment

Food intake assessment and documentation play an important role in management of obesity and other health problems, such as heart diseases, hypertension and cancers. However, few people are aware of their food intakes and some of them are even not willing to assess food intakes. The reason is the burdensome assessment methods and a lack of real-time feedback with these existing methods.

Existing dietary assessment methods include food records and food diaries [7–9], which usually require manually recording and recall of food types and food portions. Using this method, people have to write down the food types and estimate the volumes of each type of food consumed. The advantage of this method is that it is applicable to most people since it does not require any professional knowledge. However, it is limited by the roughness of human knowledge and estimations of portion sizes[10]. Similar methods that suffer from the same drawbacks also include dietary histories [11], and food frequency questionnaires[12, 13]. In summary, accuracy of these methods largely relies on many uncertain factors such as user’s memory, food knowledge, and portion estimations. Therefore, the accuracy is often compromised.

Concerning about the inaccuracy caused by human estimations, researchers develop methods to monitor food intakes inside human organs. Typical methods include biological assessment and chemical analysis. Biological assessments, e.g. doubly-labelled water [14], plasma carotene [15], etc., monitor food intake through the introduction of biomarkers[16] and measure metabolic rate inside human body. The chemical analysis methods evaluate the dietary intake through tracking selected elements [17]. Both the biological and chemical methods report the validation and accuracy in food intake assessment [18]. However, these methods could only be used in the lab environment that is not available to everyone in the free living conditions and they are usually used for studies with short durations.

Efforts have been made to record calorie contents without user awareness or knowledge by processing chewing sounds of the user with on-body sensors[19]. Eating behaviors are recognized in sensory data and in each eating activity cycle, food intakes are inferred by means of recognizing food types as well as amounts from audio sensory data. However, the accuracy of food content recovery from audio signals is still questionable, and it presents the users with a lot inconvenience when wearing sensors over the neck all day long.

Accurate and convenient dietary assessment methods are still blank and needed in both population and research societies. Dietary assessment methods that do not rely solely on human’s knowledge and estimations, but provide with objective food descriptions could enhance the accuracy and efficiency of food intake assessment and contribute to improved

understanding of diet-disease relationship. With the advent of electric medical records, clinicians, researchers, and practitioners are increasingly interested in using objective food intake assessment as a tool for obesity prevention and health research.

### **1.3 Opportunity and challenges**

As the popularity of portable handheld computers such as smart phones and the cost of network decreases, opportunities for novel healthcare applications on networked handheld devices arise. Nowadays, smart phones are not only a mobile phone, but also a mobile smart terminal and gateway to gather and deliver objective information continuously over long time during free living conditions. Smart phones are often carried with people nearly everywhere and people usually keep their smart phones functioning and charged. In addition, smart phones are equipped with various sensors to gather health related information conveniently, such as cameras, motion sensors, global positioning system (GPS), and microphones. Data centers that on the other side of the network collect and store the information sent by smart phones. Computers could process the data in a centralized manner on both individual level and community level and report feedbacks to individuals, doctors, researchers, and other related databases.

One important advantage great opportunity that mobile smart phones provide is in creating valid, reliable, and objective descriptions of food intakes. Images or videos could tell much information about a meal without users' notations, or descriptions. What kind of food is in the meal, how the food is prepared, what kind of ingredient is used, and how much the meal is consumed, all these objective visual descriptions of a meal could be captured and stored by the camera on the smart phone. The inertial measurement unit on the smart phone such as gyroscopes, accelerometers may also record the motion and position of the camera, which will be helpful for size estimation. All these facilities foster the possibility of automatic food intake assessment with smart phones.

Another powerful extension of smart phone is to use it to deliver "just-in-time" interventions to users at the point of decision. The smart phone can not only capture and record information, but it could also process the information and provide feedbacks to the users. These feedbacks could include the calorie information the meal contains, the nutrition construction of the meal and advices about how the meal should be consumed. These feedbacks could be informative enough to have an useful impact on users' life styles and eating habits.

However, the challenges come together with these facilities of smart phone is the processing of gathered information, which is specifically the evaluation of food intakes from images.

The calorie intake depends on the calorie density of the food consumed, which is decided by the food types, and the portion size of the food. Therefore, the challenges of image based food intake assessment comprises two parts, food item recognition, and food intake volume estimation.

To recognize food from images is a specific problem of category recognition from computer vision. Category recognition is still under development and far from being solved because of the high degree of uncertainty and deformability of the appearances of the objects to recognize. Food recognition is even more difficult than general category recognition tasks such as animal recognition, flower recognition, and architecture recognition. The reason is that the appearance of a food item is affected by too many factors including recipes, cooking methods, chef's preferences, image perspectives, even lighting conditions. In additions, among all the factors, each of them may change food appearances significantly.

Food volume estimation from images is also a challenging problem even impossible. For any object captured through cameras, in the image the object's scale is lost due to the projections of lenses. Without the scale, even though shapes and outlines of the object could be reconstructed from images, it is still impossible to calculate its volume.

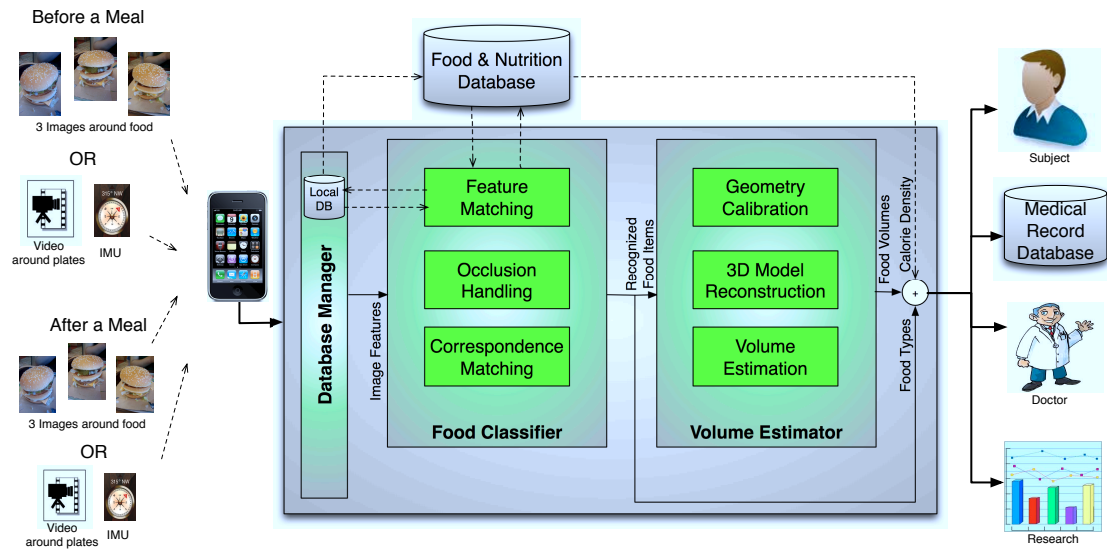
## **1.4 Goals and specific aims**

The goal of this thesis is to develop an automatic food intake assessment method using cameras, inertial measurement units (IMUs) on smart phones to help people foster a healthy life style.

Figure 1.1 illustrates a desired food intake assessment system. Images, videos and camera motion information is captured by the smart phone, before and after a meal. A food classifier recognizes each food item in the images and records them in a database. The database is used as both a knowledge base and a storage base of food intake records. The classified food items and camera motions are fed into a volume calculator, which estimates consumed food portions. The food types and portions together will be reported to subjects/patients, doctors, research institutes and electronic medical records.

The technical objective is to explore the feasibility of image based food recognition and image based volume estimation. Specifically, the following aims were investigated in this thesis:

1. Develop a prototype system with existing methods to review the literature methods,



**Figure 1.1:** The goal of this thesis: an automatic food intake assessment system using cameras, IMUs on smart phones to help people foster a healthy life style.

find their drawbacks and explore the feasibility to develop novel methods.

2. Based on the prototype system, investigate new food classification methods to improve the recognition accuracy to a field application level.
3. Design indexing methods for large-scale image database to facilitate the development of new food image recognition and retrieval algorithms.
4. Develop novel convenient and accurate food volume estimation methods using only smart phones with cameras and IMUs.

## 1.5 Organization of this thesis

This thesis is a collection of five scientific publications addressing the research goals and specific aims summarized in Section 1.4. From chapter 2 to 6, these five scientific publications are presented. In chapter 7, achievements and contributions of this thesis are summarized, and future works are discussed.

Table 1.1 lists the included publications and the chapter organizations. The publications are arranged in the order of the specific aims presented in Section 1.4. Originating from implementation of a prototype system in Chapter 2 using existing object recognition techniques,

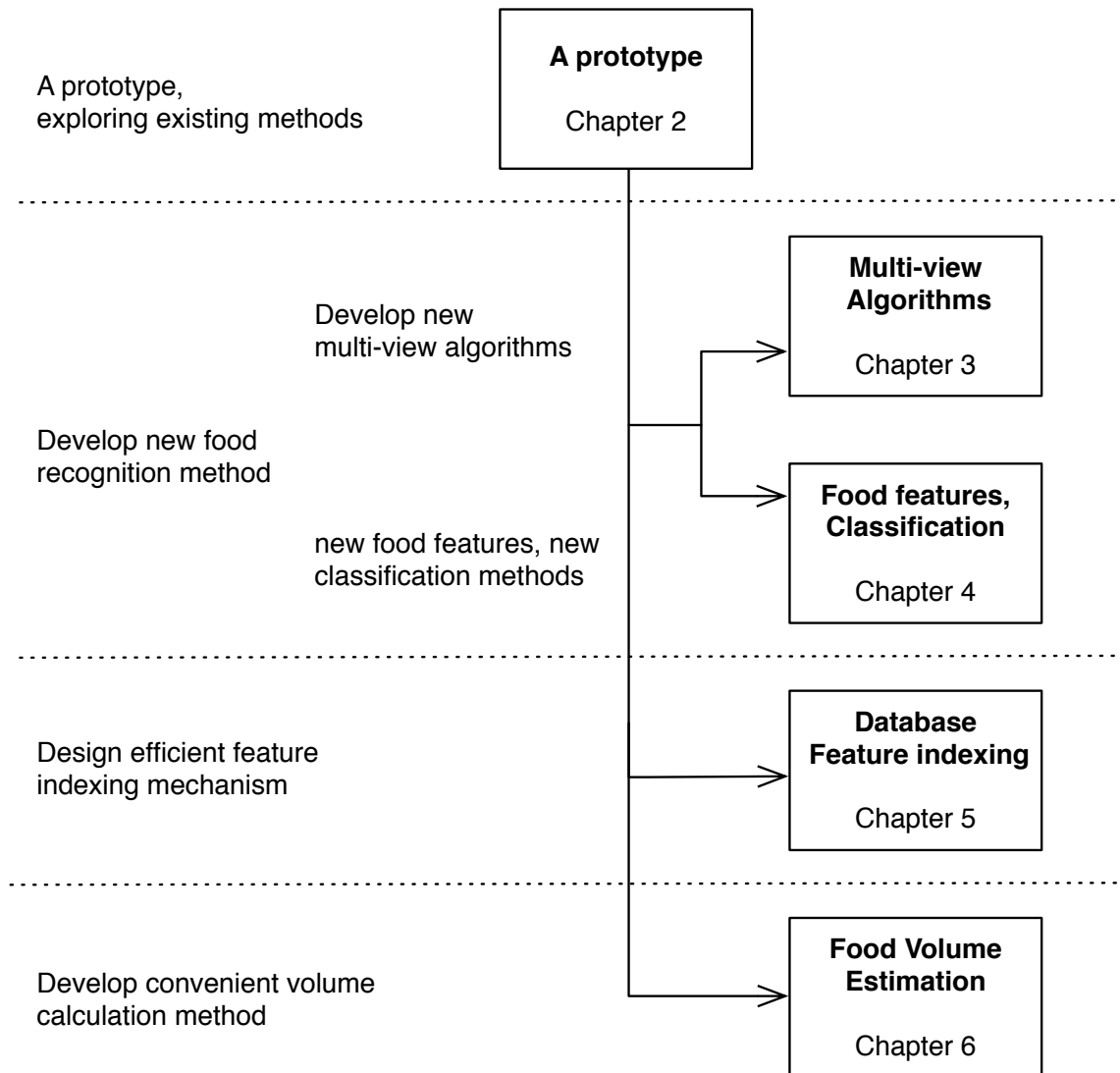
**Table 1.1**  
Publications included in the thesis.

Chapter	Publication
2	DietCam: Automatic Dietary Assessment with Mobile Camera Phones Fanyu Kong and Jindong Tan, Pervasive and Mobile Computing, Accepted, 2011
3	DietCam: Multi-View Regular Shape Food Recognition with a Camera Phone Fanyu Kong and Jindong Tan, Submitted to Systems, Man, and Cybernetics, IEEE Transactions on, 2011
4	DietCam: Multi-View, Multi-class Food Recognition Using a Multi-kernel Based SVM Fanyu Kong and Jindong Tan, Submitted to Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2011
5	Indexing of Bags of Features: Efficient Image Retrieval from Large-Scale Database Fanyu Kong and Jindong Tan, Accepted by Machine Learning and Data Mining, International conference on, 2011
6	DietVolume: Food Volume Estimation through Metric 3D reconstruction on a Mobile Phone Fanyu Kong and Jindong Tan, Submitted to Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2011

a nearest neighbor classifier for food classification, feature point based 3D reconstruction with credit card marker for food volume estimation. In Chapter 3, the recognition accuracy is improved through developing a multi-view food classification method for regular shape food. Chapter 4 increases the recognition accuracy further through investigating new food features, new food classification methods for arbitrary shape food items. In Chapter 5, a database structure and indexing mechanism are developed for high efficiency image recognition and retrieval from large-scale image database, which is essential for food image recognition. Then in Chapter 6, the volume calculation method is improved by mean of investigating sensor fusion techniques to infer the scale of 3D models so that the system does not rely on the existing of markers.

Fig. 1.2 illustrates the thesis contributions according to the specific aims presented in Section 1.4.

### Thesis Aims



**Figure 1.2:** Outline of the thesis scientific contributions according to the specific aims presented in Section 1.4.

# References

- [1] “World Health Organization [internet]. Obesity and overweight; 2011 [Updated March 2011, cited October 2011]. Available from: <http://www.who.int/mediacentre/factsheets/fs311/en/>.”
- [2] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [3] “Centers for Disease Control and Prevention [internet]. U.S. Obesity Trends; 2011 [Updated July 21, 2011, cited October 2011]. Available from: <http://www.cdc.gov/obesity/data/trends.html>.”
- [4] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [5] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [6] A. Ershow, J. Hill, and J. Baldwin, “Novel engineering approaches to obesity, overweight, and energy balance: public health needs and research opportunities,” *Engineering in Medicine and Biology Society, IEEE Annual International Conference of*, pp. 5212–5214, Jan 2004.
- [7] G. Godin, A. Bélanger-Gravel, A. Marie Paradis, M.-C. Vohl, and L. Pêrusse, “A simple method to assess fruit and vegetable intake among obese and non-obese individuals,” *Can J Public Health*, vol. 99, no. 6, pp. 494–8, Jan 2008.
- [8] M. A. Murtaugh, K. ni Ma, T. Greene, D. Redwood, S. Edwards, J. Johnson, L. Tom-Orme, A. P. Lanier, J. A. Henderson, and M. L. Slattery, “Validation of a dietary history questionnaire for American Indian and Alaska Native people,” *Ethn Dis*, vol. 20, no. 4, pp. 429–36, Feb 2011.
- [9] N. D. Wright, A. E. Groisman-Perelstein, J. Wylie-Rosett, N. Vernon, P. M. Diamantis, and C. R. Isasi, “A lifestyle assessment and intervention tool for pediatric weight



- management: the habits questionnaire,” *J Hum Nutr Diet*, vol. 24, no. 1, pp. 96–100, Feb 2011.
- [10] A. F. Smith, S. D. Baxter, J. W. Hardin, C. H. Guinn, and J. A. Royer, “Relation of children’s dietary reporting accuracy to cognitive ability,” *Am J Epidemiol*, vol. 173, no. 1, pp. 103–9, Jan 2011.
  - [11] L. A. Mainvil, C. C. Horwath, J. E. McKenzie, and R. Lawson, “Validation of brief instruments to measure adult fruit and vegetable consumption,” *Appetite*, vol. 56, no. 1, pp. 111–7, Feb 2011.
  - [12] M. A. Cardoso, L. Y. Tomita, and E. C. Laguna, “Assessing the validity of a food frequency questionnaire among low-income women in são paulo, southeastern brazil,” *Cad Saude Publica*, vol. 26, no. 11, pp. 2059–67, Nov 2010.
  - [13] F. H. Esfahani, G. Asghari, P. Mirmiran, and F. Azizi, “Reproducibility and relative validity of food group intake in a food frequency questionnaire developed for the tehran lipid and glucose study,” *J Epidemiol*, vol. 20, no. 2, pp. 150–8, Jan 2010.
  - [14] A. E. Dutman, A. Stafleu, A. Kruizinga, H. A. Brants, K. R. Westerterp, C. Kistemaker, W. J. Meuling, and R. A. Goldbohm, “Validation of an FFQ and options for data processing using the doubly labelled water method in children,” *Public Health Nutr*, pp. 1–8, Aug 2010.
  - [15] M. Aubertin-Leheudre, A. Koskela, A. Samaletdin, and H. Adlercreutz, “Plasma alkylresorcinol metabolites as potential biomarkers of whole-grain wheat and rye cereal fibre intakes in women,” *Br J Nutr*, vol. 103, no. 3, pp. 339–43, Feb 2010.
  - [16] G. L. Bowman, J. Shannon, E. Ho, M. G. Traber, B. Frei, B. S. Oken, J. A. Kaye, and J. F. Quinn, “Reliability and validity of food frequency questionnaire and nutrient biomarkers in elders with and without mild cognitive impairment,” *Alzheimer Dis Assoc Disord*, vol. 25, no. 1, pp. 49–57, Jan 2011.
  - [17] P. B. Ryan, K. A. Scanlon, and D. L. MacIntosh, “Analysis of dietary intake of selected metals in the nhexas-maryland investigation,” *Environ Health Perspect*, vol. 109, no. 2, pp. 121–8, Feb 2001.
  - [18] M. R. Ritchie, M. S. Morton, N. Deighton, A. Blake, and J. H. Cummings, “Plasma and urinary phyto-oestrogens as biomarkers of intake: validation by duplicate diet analysis,” *Br J Nutr*, vol. 91, no. 3, pp. 447–57, Mar 2004.
  - [19] O. Amft, “Automatic dietary monitoring using on-body sensors, detection of eating and drinking behaviour in healthy individuals,” *PhD dissertation, Swiss Federal Institute of Technology Zurich*, 2008, Jan 2008.

## Chapter 2

# DietCam: Automatic Dietary Assessment with Mobile Camera Phones

### Abstract

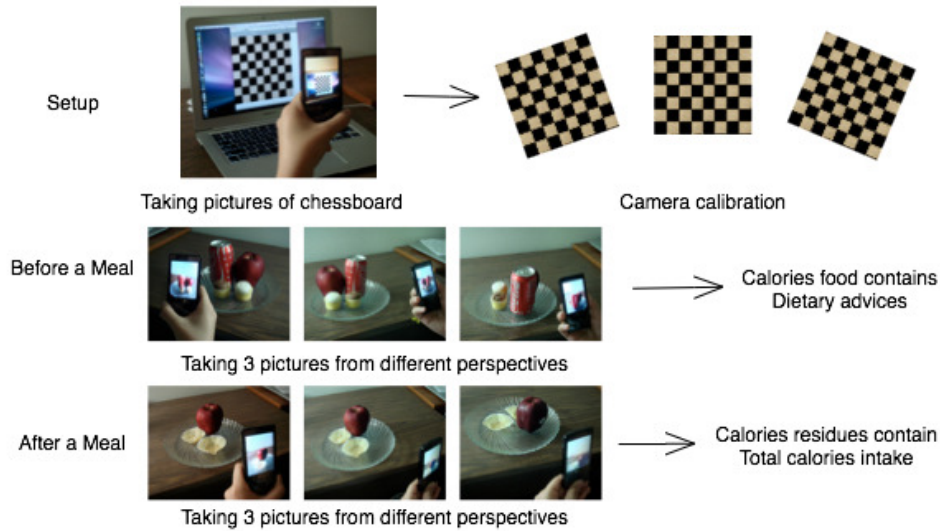
Obesity has become a severe health problem in developed countries, and a healthy food intake has been recognized as the key factor for obesity prevention. This paper presents a mobile phone based system, DietCam, to help assess food intakes with few human interventions. DietCam only requires users to take three images or a short video around the meal, then it will do the rest. The experiments of DietCam in real restaurants verify the possibility of food recognition with vision techniques.

### 2.1 Introduction

Mobile phones are becoming a popular and powerful platform, and many healthcare-related applications have been explored, such as remote health monitoring, SMS medical tips, fitness coaches, and diabetes guides[1]. Obesity, another possible cell phone aided healthcare problem, is becoming an epidemic phenomenon in most developed countries. In the past three decades, obesity rates for both adults and children in the U.S. have increased significantly[2]. The fact that more than thirty-three percent of adults and sixteen percent

---

The material contained in this chapter has been accepted for publication in the journal Pervasive and Mobile Computing.



**Figure 2.1:** Expected usage. The calorie information, which is a key to the obesity problem, will be extracted from three images or a short piece of video of the foods.

of children are obese has proven to be one of the biggest public health challenges to the general population and social welfare. The serious consequences of obesity include severe health problems such as diabetes, stroke, and heart disease, and high-cost healthcare bills, which were estimated at \$147 billion in 2008 in the U.S. alone [3, 4]. The continuing increase of overweight and obesity attributable spending has attracted increasing research interest to explore practical new technology to prevent obesity.

In spite of the common sense that obesity is a complex condition caused by the interaction of many factors such as genetic makeup, secondary effects from medical treatment, and calorie imbalance, it is generally believed that obesity prevention requires individuals to foster life-long healthy food choices and regular physical activities [5]. However, the usual case is that individuals with potential obesity problems are more likely to ignore their food intakes and regular exercise. Even people who care and pay attention to nutrition information may not be sufficiently knowledgeable about the calorie content of what they are eating. Efforts have been made to record calorie contents without user awareness or knowledge by processing chewing sounds of the user with on-body sensors[6]. However, the accuracy of food content recovery from audio signals is still questionable, and it presents the users with a lot inconvenience when wearing sensors over the neck all day long.

Opportunities for novel obesity management applications arise as mobile phones are becoming more powerful for people-centric computing. The fact that mobile phones nowadays are necessary and are carried by people nearly everywhere makes them perfect devices for information gathering and delivering during free living conditions. Cameras, which are

equipped on most smart phones, can provide rich and reliable information. Another powerful extension of mobile technology is the combination of accelerometers, which benefit in creating valid measures of physical activities. Even though obesity and diabetes related mobile phone applications have appeared, most of them only use the mobile phone as a food diary[7–10] or fitness diary[11–13] that requires large amounts of user input. Cameras help record dietary information automatically[14], but users still have to manually review the processed image results. We have developed a health-aware smart phone system which employs an obese prevention application utilizing the embedded camera and accelerometer. Besides extracting physical activity data through built-in accelerometer readings, it monitors food intake automatically with few user interventions.

In this paper, the automatic food calorie estimation system DietCam in a health-aware system is proposed, as shown in Fig. 2.1. It is able to recognize foods and calculate the calorie content of a meal automatically from images or videos with few human interventions. Before it is in use, the camera on the cell phone needs to be calibrated in a user-friendly way. When utilizing DietCam, users only have to put a credit card beside the plate and take three pictures around the dish approximately every 120 degrees or shoot a piece of video. After that, DietCam will do the rest for the users to obtain the calorie information. Vision techniques are utilized to extract visual cues of the calorie information from images or the piece of video (if equipped with a digital compass) around the plate. Based on these visual cues, food recognition algorithms are designed to classify the food items. At the same time, three-dimensional (3D) models of visible food items will be reconstructed in order to estimate the volume of the food. The metric scale of the 3D model is inferred from the credit card. Types, volumes, and calorie densities of the food items together identify the calorie content of the meal.

The accurate measurement of food contents through vision techniques is challenging. At present, there is no technology that allows users to estimate the calorie contents of a meal automatically and comfortably. The following challenges exist in this project.

1. Many different kinds of food have the same or very similar appearance that is hard to distinguish from a camera's point of view.
2. Even though some kinds of food have specific appearances, the diversity of the same kinds of food makes it impossible to recognize all these foods.
3. A meal usually has more than one food items. It is hard to segment those foods with irregular shapes, especially when occlusions exist in the image. The varying lighting conditions of restaurants make this problem even harder.
4. Even if the types of food have been recognized correctly, the amount of food is another factor affecting the calorie intake directly. Sometimes people will not eat the whole meal. It is necessary to estimate the portion consumed.

5. Even though all the above challenges are solvable by carefully designed algorithms, is it practical to implement these algorithms on a mobile phone?

Our technique addresses these challenges by utilizing a multiple-view method. The approaches are lightweight and feasible on a commercial smart phone. A prototype has been implemented on an iPhone, and the results are promising. Our main contributions are as follows,

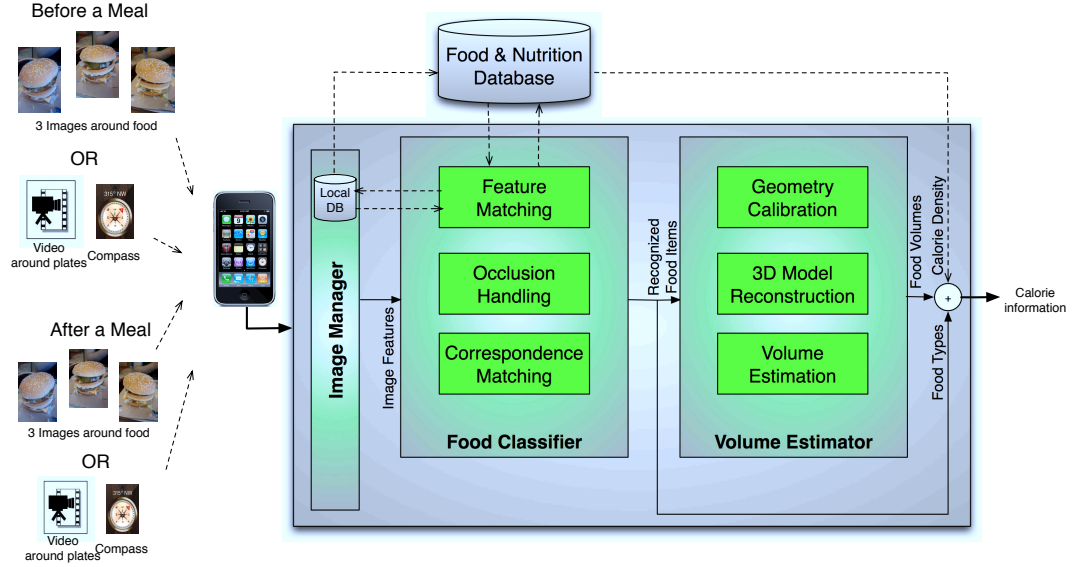
1. **Identifying the possibility of obtaining calorie information of a meal through a camera phone.** A prototype has been implemented on an iPhone. The algorithms are under study on Windows Mobile, Android, RIM and Symbian platforms.
2. **Developing multiple-view image understanding algorithms for contents recovery.** We perform simple feature extraction on multiple images. Novel segmentation, classification, occlusion, and correspondence handling algorithms are developed for food classification. A model based volume estimation mechanism is developed.
3. **Evaluation of the scheme at home and in real restaurant locations.** We collect test samples at home, different local restaurants, and supermarkets with different combinations of food items and at different times of a day. As many as 21 business restaurants are covered. An average recognition accuracy of 92% is achieved.

The rest of this paper solves each of the challenges and elaborates on these contributions. Section 2.7 evaluates DietCam with field experiments. We discuss the related work in section 2.9, and conclude the paper in section 2.10.

## 2.2 System Architecture

DietCam mainly consists of three parts: image manager, food classifier and volume estimator. The overall architecture and data flow are shown in Fig. 2.2. High-level data flows are described first, followed by the internal details.

The system begins with sensing food images, and ends with rendering food calories. The images are recorded, processed, and transmitted to a server, and the results are shown to the user. The input to the system could be three images around a meal or a piece of video if the cell phone is equipped with a digital compass. If a short video is recorded, the image manager will extract three different perspective frames from the video according to the digital compass readings. The outputs of the image manager are feature descriptions in the



**Figure 2.2:** System architecture. The types of food in a meal are classified by the food classifier. The volume of every food item is generated by the volume estimator.

three images. Those features are abstractions of the image used to describe special points in the images. The food classifier uses these features to separate and classify each food item. It matches the features of every food item against the references in the database, which is a large container of many kinds of food. It is possible that the classifier will find no matches in the database for some food items, which means that these foods could not be recognized from their appearances. Non-appearance based recognition methods will be adopted. The recognized food items are forwarded to the volume estimator, which estimates the volumes of each food item. As a result, the food type recognized by the food classifier, the calorie density information in the database, and the volume of the food together determine its calories.

The food classifier processes the features in the three food images to segment and classify every food item. The features in every image will be matched to a local food database first. If they are not matched in the local database, they will be queried in the larger global database. As a result, every kind of food differentiable through appearances will be recognized. However, occlusions in a single image could cause some food items to be covered by others. So, in order to recover as many food items as possible, three images are used. With the information of food types existing in the images, occlusion and correspondence handling algorithms are developed to segment and extract every food item in the meal. Consider the fact and challenge that some types of food may have the same appearance, which means that those foods cannot be recognized through appearances only. Therefore there might be some foods whose features cannot be matched in the database. During

the feature matching process, these types of food can be separated from appearance differentiable foods. They will be identified later by means of optical character recognition (OCR) techniques[15] or user inputs. Another challenge is that the same types of food item may have different shapes or colors, and there are no two food items that are exactly the same. Obviously, object recognition algorithms that are good at matching an object from one image to another are not suitable for recognizing food classes. A Bayes decision theory based probabilistic food classification algorithm is developed to classify food items. The approach is built upon feature matching based object recognition and the statistical nature of the features, considering the noise in the measuring camera sensors. In addition, a vocabulary tree data structure[16] is used to make image matching scalable.

The volume estimator calculates the volume and portion of each recognized food item. In order to get the geometry properties of the scenes, camera calibration is required. The calibration process is not trivial, since cameras on different brands or different series of mobile phones could be different, and every time the camera shoots an image, its position and direction could be varied. One possible method is to make the users take a marker with them and place the marker in the camera's field of view when taking images[17]. Obviously, it is not convenient for the users to take a useless marker when the application is not used. DietCam uses an automatic correspondence based [18, 19] calibration method to estimate parameters of the camera. When the application is installed, the intrinsic parameters of the camera are calibrated. In other words, the constant intrinsic parameters are calibrated only once. The extrinsic parameters that are changing when the application is running are calibrated on the go. With the camera information, the volumes are estimated by rebuilding 3D models of the food items. The scale of the scene is inferred from a known size object. A credit card is put into the scene when the images or video is taken. In order to calculate the volume more accurately, the 3D models are divided into two groups: models of regular shaped food items and models of irregular shaped food items. On the one hand, regular shaped food items can be modeled as spheres or cylinders. The volumes of these types of food items can be estimated by calculating the parameters of their shapes. On the other hand, 3D models of irregular shaped food items will be reconstructed by the means proposed in [20]. During the reconstruction process, features extracted for food classification are used again to find correspondent points in the multiple images. The correspondent points will be the vertices of the 3D model. After that, the volume of the 3D model can be calculated with the coordinates of the vertices.

The food databases collect food images and nutrition information such as calorie densities of most kinds of food. The local database collected by the image manager stores food types the users have eaten. It provides a chance to increase the searching efficiency when looking up food types in the database. The large global database resides outside the system. It collects food images from all the users and other resources. Searching time in the large database will be much longer than that in the local database.

Extra work is needed to obtain the calorie information of unrecognized foods, which are unrecognizable through appearances. The labels and tags on the bags and bottles give us a straightforward method to know the calorie facts. When having food with a label, the users can shoot the label with the camera. OCR techniques[15] can be used to recognize the label and provide calorie information. OCR is the mechanical or electronic translation of images of handwritten, typewritten or printed text into machine-editable text. The accurate recognition of typewritten text is now considered largely a solved problem on applications where clear imaging is available. With the knowledge of food types and food dimensions, calorie density is used to roughly estimate the calories a food item contains. Food calorie densities are from the USDA Food and Nutrient Database[21], and they are stored in the food database.

## **2.3 Food Classification**

Recognizing the type of food in a meal is the first step of dietary management. The food classifier segments each food item from the scenes and recognizes each of them. What the classifier needs are the image features of three images. It works with visual features rather than the unprocessed images. In a food image, separating a clutch of food into food items is a challenge. A probabilistic food classification algorithm is developed to identify food types. The algorithm is required to recognize the same kind of food with different appearances. Occlusion and correspondence handling algorithms help to integrate the food items in all three images together to obtain the actual food items and types on the plate.

### **2.3.1 Food Features**

The image features used by the food classifier are extracted by the image manager, whose role is defined in section 2.5. The visual features describe the images by detecting special points and abstracting the characteristics of the points. Every type of food is associated with visual features that describe its characteristics in images. Many kinds of visual features have been developed in the literature of computer vision[22, 23]. DietCam requires a feature detector and descriptor that is invariant to lighting changes, rotation, and scale, since it will be used to recognize food items from different perspectives at different places.

The scale invariant feature transform (SIFT)[24, 25] is an ideal feature detector and descriptor meeting the requirements of DietCam. It is identified as the most popular feature detector and descriptor for object recognition because of its invariance to scale, orientation, affine distortion, and partial invariance to illumination changes. To recognize food items





**Figure 2.3:** Training set of cheeseburgers and apples from different perspectives.

in an image, DietCam matches SIFT features to those reference features known as certain kinds of foods in the database. However, the fact that the SIFT feature is a continuous 128-dimensional vector, and an image has several hundred SIFT features, makes it expensive to determine the similarity between images by matching SIFT features. This problem is addressed by clustering SIFT features into visual words with an efficient hierarchical k-means clustering algorithm[16].

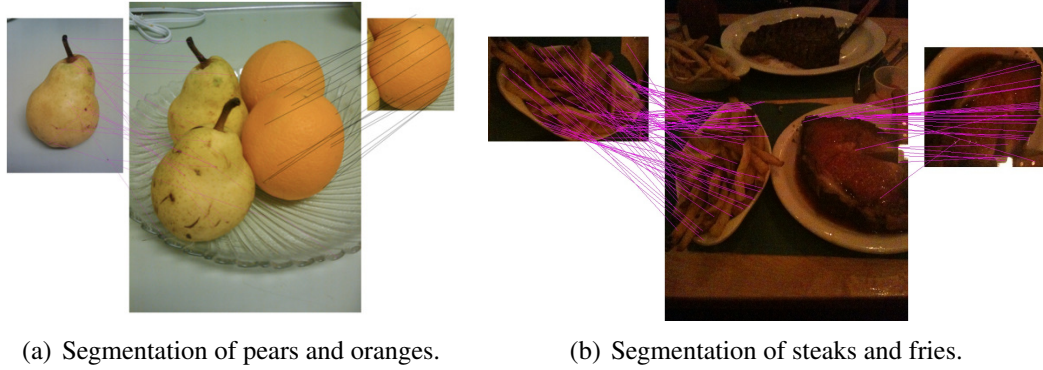
### 2.3.2 Food Recognition

When matching food features to the database, it is possible that a food item is not matched to the correct type in the database. This is caused by the diversity of food. Even the same food item may have different appearances from different perspectives. Take apples as an example: if there is only one green apple in the database, a red apple might not be recognized as an apple. The uncertainty of food appearances makes it impossible for the database to cover all the possible visual appearances. Mismatches happen especially when too few samples of the same food type exist in the database.

This problem is solved with two approaches. On the one hand, more samples of the same type of food are collected in the database. On the other hand, a probabilistic food classification method is developed based on the database.

In the database, a food category has multiple visual descriptions to cover more possibilities. Multiple instances of the same food type are picked and their images are taken in different perspectives and lighting settings. Therefore, in the database a type of food will have a large number of training images. Each of these training images contains only one food item. In this way, features of each image will be a clean description of the food item contained, rather than being messed up by other food types. Fig. 2.3 shows an example of a training set of cheeseburgers and apples.

Considering the uncertainties for a food item belonging to a food group mentioned above, the recognition process has to classify the unknown food item to the most probable food type by matching features against the references in the database. The number of matched SIFT features determines the similarity between two food images. Therefore, a classifier



**Figure 2.4:** Feature matching based food segmentation. 2.4(a) shows how it works to segment a plate of fruit. The red lines indicate the matches between the pears in the image and in the database. The black lines present the matches between the grape fruits. 2.4(b) shows the segmentation of a steak meal in dark lighting conditions.

that classifies food items based on the number of matched vectors is required.

Many classifiers have been proposed in the pattern recognition field. Most of the methods can be grouped into linear classifiers and non-linear classifiers. Obviously, the food classification problem is not a linear classification problem, since the evaluation method is to find the numbers of matched vectors rather than a linear function. The similarity between two images is defined as the number of matched features. The more features matched, the more similar these images are. The most probable type a food item belongs to is the group with the largest number of matched features. This can be solved by a nearest-neighbor classifier, a type of probabilistic classifier that is simple enough to run on the cell phone.

If there are  $M$  food types in the database,  $\omega_1, \omega_2, \dots, \omega_M$ , and an unknown food item represented by a visual word vector  $x$ , the possibilities  $x$  belongs to  $\omega_1, \omega_2, \dots, \omega_M$  are  $P(\omega_i|x), i = 1, 2, \dots, M$ . The most probable food type of food item  $x$  is type  $\omega_i$  which has the largest  $P$  value.  $P$  can be calculated through the Bayes rule. The class conditional probability density functions  $p(x|\omega_i), i = 1, 2, \dots, M$  describe the distribution of the feature in each of the classes. They are defined in the feature matching process. When  $x$  is matched against the multiple instances of the same food category  $\omega_i$ , the number of total features and the matched features in every instance are recorded. Then,  $p(x|\omega_i)$  is defined as the maximum proportion of the matched visual words to the total number of features in  $x$ , which is the nearest neighbor.

### **2.3.3 Food Segmentation**

Segmenting food items is important for both the recognition process and the volume calculation process. There have been some image segmentation methods designed specially for food segmentation. In IBM's Veggie Vision[26], histograms are used to segment the food products from the backgrounds. However, in this commercial system, only one type of food is involved in the image and the lighting condition is constant. When there are different types of food, food items cannot be segmented correctly based only on the histogram.

The food features in the database can serve as food templates to extract food items of that type from the food clutch. When a food item is matched in the database, the template fits in the image, as shown in Fig. 2.4. In this way, a subtraction based mechanism is developed to divide the whole scene into individual food items, after the visual features of the image have been extracted. The generated visual features will be classified by matching against the database. The food item classified with the largest number of visual features will be recorded in a list and its visual features in the image will be subtracted. Then the classification process will operate again, until there are no visual features left or the remaining visual features cannot be matched to any kind of food.

### **2.3.4 Occlusions and Redundancies**

A multi-view food classification method is developed, since it is not always possible to recognize all the food items in a meal from only one image. Occlusions cause some food items to be covered by others. An intuitive idea is to look at those covered food items from another perspective. In other words, on the hand-held camera phones, a multi-view food classification method is desired. Another benefit a multi-view scheme brings about is a transition from a single 2D image to a 3D environment, where food volume calculation becomes possible.

From multiple views, the problem of missing food items caused by occlusions from a static point of view can be solved. In this way, however, a single food item might be taken into account more than once. Therefore, food item redundancies exist, and the food items in these images need to merge together to reflect what exactly is on the plate. Reducing the redundancies caused by reproductions of the same food item from multiple views is a challenge. Since this problem is caused by the ignorance of the correspondences between multiple views, the idea is to look up the visual similarities between pairs of views and find correspondences between images then get rid of the redundancies.

---

**Algorithm 1** Redundancy Reduction

---

**Require:** Images  $\{I\}$ , SIFT features  $\{S\}$ , Redundant Food Items  $\{F\}$

**Ensure:** Essential Food Items,  $\{F\}$

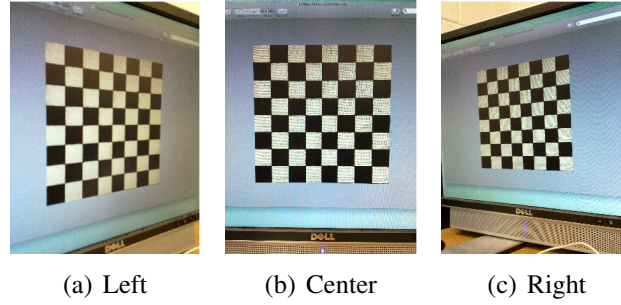
```
1:  $n \leftarrow 3$ , number of image pairs
2: for  $i = 1$  to  $n$  do
3:   Find  $i$ th image pair  $I_1, I_2$  from  $I$ 
4:   Find 8 corresponding boundary SIFT feature point in  $I_1, I_2$ 
5:   Calculate the fundamental matrix of the  $I_1, I_2$ 
6:   Find all the food items  $\{F_1\}$  in  $I_1$  with  $\{S\}$ 
7:   for Every food item  $f$  in  $\{F_1\}$  do
8:     Find correspondent searching window in  $I_2$ 
9:     if The same type of food exists in the window then
10:      Keep  $f$ 
11:     else
12:        $F = F - f$ 
13:     end if
14:   end for
15: end for
```

---

Algorithm 1 shows the whole procedure. It operates a pair of images one at a time. Every pair of images will be processed. For an image pair, the algorithm starts with initializing the geometry relation between them. The SIFT features are used again as the image descriptor, by matching which, the similarities between these views are found. The geometry relationships between the two views are calculated with the matched feature pairs and modeled by the fundamental matrix in epipolar geometry[19]. (Epipolar geometry is the intrinsic projective geometry between two views, which is encapsulated by the fundamental matrix. With the help of the fundamental matrix, a point in an image corresponds to a line in the other image. In other words, it reduces finding correspondent points in other images to searching points along a line). Starting from one image, for a food item recognized in the image, a searching window will be found in the other view through the fundamental matrix. The searching window of a food item is the correspondent position containing the same food item in the other image. Therefore, in the searching window, if the same kind of food item exists, this food item in the second view is redundant and it will be deleted from the food list.

## 2.4 Volume and Calorie Estimation

The food volume estimator calculates the volume of each food item recognized by the food classifier. It takes food categories and feature points as input and gives out the volume of



**Figure 2.5:** Camera intrinsic parameter calibration. Three different perspective chessboard images will give enough information to calibrate the camera, which is easy and convenient for the users.

each food item. In this process, the scale information of the food is no longer available when looking through the camera. Therefore, the camera on the cell phone needs to be calibrated. However users should be freed from the calibration process. With the scale information, the volume is estimated by calculating the volume of the food 3D models. It is a challenge to accurately build food 3D models from feature points. Recognized food categories and known shape patterns help to define the 3D model of each item. For those types of food known with irregular shapes, 3D models are reconstructed based only on the feature points. After that, the volume of these models will be figured out with geometry calculations.

## 2.4.1 Camera Calibration

First of all, the camera on the cell phone needs to be calibrated. In order to reduce the user intervention, the intrinsic parameters that require user interactions to calibrate are calibrated separately from the extrinsic parameters. The calibration of the static intrinsic parameters proceeds offline and only acts once. In contrast, the calibration of extrinsic parameters is carried out every time volume calculations are required.

### 2.4.1.1 Intrinsic Parameters Calibration

Since DietCam will be installed on different types of mobile phones with different kinds of cameras, a user-friendly and general method to calibrate a camera's intrinsic parameters is needed. Many camera calibration methods have been proposed in the computer vision literature[27]. Among these methods, the flexible camera calibration method[18] is well suited for the requirements. This method does not require any professional knowledge

other than the user shooting a planar pattern from two or more perspectives. We provide a chessboard pattern online, which is not only convenient for the users to access, but also a known standard pattern to calibrate different types of cameras.

The parameters of a camera can be represented as

$$P = A[R \ T] \quad (2.1)$$

where

$$A = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

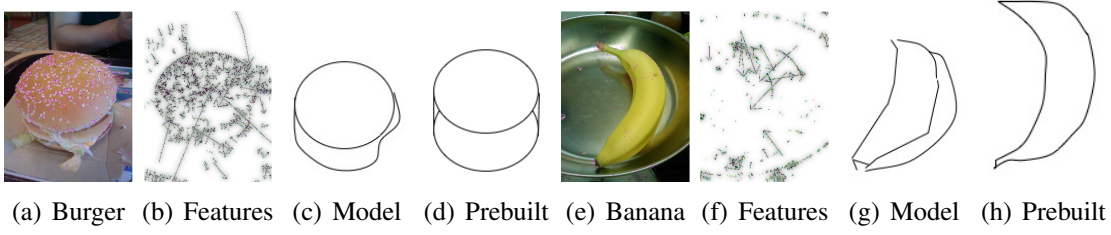
is the intrinsic parameter matrix:  $R$  and  $T$  are extrinsic rotation and translation parameters. In the intrinsic matrix  $A$ ,  $(u_0, v_0)$  is the coordinate of the principal point,  $\alpha$  and  $\beta$  are the scale factors in the  $u$  and  $v$  axes, and  $c$  is the parameter representing the skewness of the axes  $u$  and  $v$ . The intrinsic matrix is calibrated by finding correspondences between multiple views and solving the constraint equations established by the correspondences[18]. Once calibrated, it will be stored on the hard drive and no longer needs to be calculated again.

When DietCam is installed, the camera's intrinsic matrix is calibrated. The users take three pictures of the chessboard under different orientations by moving the mobile phone as shown in Fig. 2.5. In the images, the inner corners of the chessboard will be detected. After this, the intrinsic parameters will be estimated with the closed-form solution of the constraint equations.

#### 2.4.1.2 Extrinsic Parameters Calibration

Unlike the static intrinsic parameters  $A$ , the extrinsic parameters  $R$  and  $T$  are always changing when taking pictures. Consequently, they need to be estimated every time right after the food pictures have been taken. We assume these three images are taken by three different cameras at the same time. Therefore, three cameras have to be calibrated through three images. In order to calculate the extrinsic parameters of these three cameras, the images are grouped into two pairs, where the image in both pairs defines the world coordinate. This camera is defined as the reference camera. The other two cameras are calibrated pair by pair.

In order to make the users unconcerned with the calibration process, epipolar geometry is used because it only requires correspondences in the image to calibrate the extrinsic parameters. The correspondences between a pair of images are already extracted in the feature matching process. With a pair of images, the camera matrix of the reference image



**Figure 2.6:** 2.6(a) is the SIFT extraction of a burger. 2.6(b) shows these points in the 3D space. 2.6(c) shows the 3D model reconstructed directly from those points. 2.6(d) presents the supposed prebuilt model, where it is clear to see the differences. Similarly, 2.6(e) to 2.6(h) present an example of a banana, where the reconstructed model has an obviously larger volume than the prebuilt model.

can be chosen as

$$P = A[I \ 0] \quad (2.3)$$

where  $I$  is a  $3 \times 3$  unit matrix. By doing this, the world coordinate system is decided. After the mobile phone is moved to take another picture, the new camera matrix related to the world coordinate system is determined as

$$P' = A[R \ T]. \quad (2.4)$$

The extrinsic parameters  $R$  and  $T$  can be estimated with the intrinsic matrix and correspondences between these two views. In epipolar geometry, the essential matrix  $E$  encapsulates the projection relationship between two intrinsically calibrated cameras. On the one hand, it has the property

$$pEp' = 0 \quad (2.5)$$

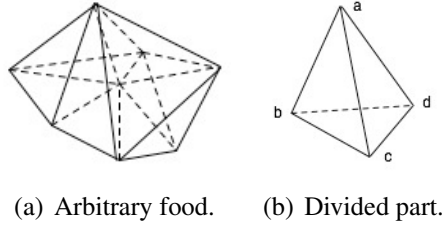
where  $p$  and  $p'$  are correspondent points in two views. Hence,  $E$  can be estimated with the correspondent points. On the other hand, according to its definition

$$E = [T]_{\times} R \quad (2.6)$$

where  $[T]_{\times}$  is the skew-symmetric matrix of  $T$ . As a result, the extrinsic parameters  $R$  and  $T$  can be estimated by singular value decomposition (SVD)[28].

## 2.4.2 3D Model Reconstruction

The volume of a food item is defined as the volume of the food item's 3D model. With the calibrated camera, the 3D positions of the feature points matched in any pair of images are computable through back-projection. The 3D models are reconstructed by the points



**Figure 2.7:** Tetrahedrons of an arbitrary shaped food item. 2.7(a) shows all the tetrahedrons: the point in the center is the estimated mass point. 2.7(b) shows a single tetrahedron with point coordinates  $a, b, c$ , and  $d$ . The volume of this tetrahedron is calculated with equation (2.7).

belonging to the food items. An intuitive method is to reconstruct 3D models of each food item directly from the points, then calculate the volume of each 3D model. However, the resolution of the 3D models reconstructed with sparse feature points is low. In the experiment, it is observed that low-resolution 3D model reconstructions cause inaccuracies. In some cases, the feature points are not enough to cover the entire food item and in other cases the feature points cover more than the actual food area. Fig. 2.6 shows these situations.

In order to increase the accuracy, predefined shape models are used for regular shaped food items, and the 3D models reconstructed directly from the points are only used for irregular shaped food items. For those types of food with regular shapes, for example apples, bananas, hamburgers, etc., a volume estimation model is prebuilt associated with the food type. The apple is modeled as a sphere, where the parameter is the diameter; a hamburger as a cylinder, where the diameter and height are the factors deciding the volume. For those food items with irregular shapes, Kong's method[20] is utilized to reconstruct the 3D model from the points.

### 2.4.3 Volume Calculation

After 3D models are reconstructed, their volumes can be calculated if their geometric properties are measured. Taking into account the predefined food item models, the parameters are the key to calculate the volumes. Therefore, the task is to estimate the values of these parameters. For example, the diameter of a sphere-type model is measured as the longest distance between all the 3D points and searched among all the points. The boundary of the searching is defined by the outline of the food item in the images. To decide the height of a cylinder-type food item, a directional longest distance along the  $y$  axis will be determined.

For an arbitrary shaped model, whose volume is not computable directly from the co-



ordinates of the points, its volume is calculated by dividing the whole model into small elements, based on the idea of finite element analysis[29]. In finite element analysis, a 3D object can be divided into a finite number of arbitrary shaped parts. A meal is divided into several food items based on the classification information, and a food item is divided up further. For every food item, the coordinates of all the points of this item are calculated. Then, the mass point of the item is estimated by averaging the coordinates of all the points. After that, the mass point is connected to each 3D point, forming a group of tetrahedrons, as Fig. 2.7 shows. The volume of the food item is the sum of the volume of every single tetrahedron. With the coordinates of the four points of the tetrahedron, the volume can be calculated with a dot product and a cross product as

$$V = \frac{(a - d) \cdot ((b - d) \times (c - d))}{6} \quad (2.7)$$

where  $a, b, c, d$  are the coordinate vectors of the points.

#### 2.4.4 Calorie Estimation

With the knowledge of food types and food scales, the calorie density is used to roughly estimate the energy a food item contains. Table 2.1 shows part of the calorie density chart DietCam makes use of, which is from online resources[21]. With the volume  $v$ , mass density  $\rho$ , calorie density  $c$ , the number of calories is defined as

$$Cal = v \times \rho \times c \quad (2.8)$$

### 2.5 Food Database

DietCam has two databases, a global database and a small personal database. The global database stores a large number of food types. Noticing the large size and the slow searching time in the global food database, a small personal database is developed as a cache in the image manager. The image manager has the image recording function besides extracting SIFT features. The images recorded will form a small personal food database. The fact that people are more likely to have a certain dietary style gives this feasibility. Considering the high possibility of food recurrence, it will be valuable to keep a record of what kind of food the users have eaten. When looking for the food types, this small database will have a higher hit rate compared with the large global database. In this way, before looking up in the large database, the personal database will be checked first. The main contents of the food database are food types, visual descriptions of each food type, and their nutrition

**Table 2.1**  
Calorie Density Chart.

Food Type	Total kal in 100 grams
Apple, raw, w/ skin	83
Wheat brand bread	248
Sliced sour dough bread	255
Cheese burger	286
Steak	176
Sandwich, ham & cheese	241

information. The database is built from the most popular food types including fast food, steak meals, fruits, and other high-calorie foods. The images are collected manually from the developers' input and from a food image website[30]. Every type of food is associated with SIFT features that describe its characteristics in images. The features are clustered into visual words with an efficient hierarchical k-means clustering algorithm. The visual words are stored in the database. The calorie density information of a type of food is another key content in the database. The USDA Food and Nutrient Database provides an accurate energy measure.

In the database, a food type will have multiple visual descriptions to cover more lighting and perspective possibilities. Food images taken in different settings are chosen as training images. Each of these training images contains only one food item. In this way, the features of this image will be a clean description of the food item contained, rather than being messed up by other food types.

## 2.6 Client-Server Architecture

DietCam uses a client-server configuration for the connectivity between mobile phones and the database. With the concerns of security, the clients will not connect to the database server directly. The clients connect to a web service and post data to that page first, then the web service will gather the data and send it to the database. The process starts from the food classifier, which sends the image features to the web server through HTTP protocol. The web service sends a query to the database server and retrieves the results. The results will be sent back to the mobile phones in an XML file that will be parsed on the phone. Since the information in the network is closely related to privacy, we are considering utilizing SSH as the communication protocol to enhance security in the future.

## 2.7 Evaluation

DietCam has been implemented on the iPhone platform and evaluated with experiments in real settings. In this section, the implementation and experimental setup are presented first, and then the performance is evaluated.

### 2.7.1 Experimental Setup

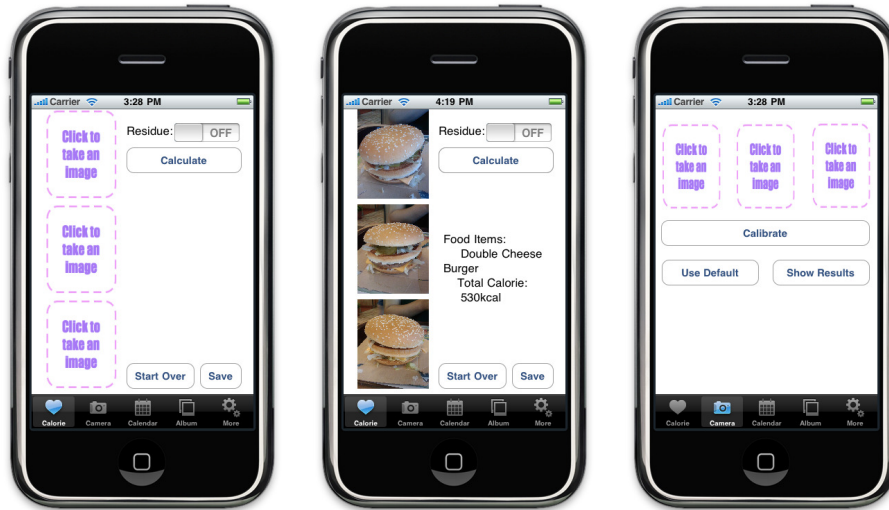
A prototype of DietCam has been implemented on the iPhoneOS platform. The evaluation is based on an iPhone 3Gs mobile phone. The iPhone 3Gs has a three megapixel camera and a powerful ARM Cortex A8 processor, which gives DietCam sufficient image resolution and computing resources. At present, DietCam is also under development on Windows Mobile, Android, RIM and Symbian platforms.

The food image database is built upon a large number of images of fast food, steak, home-made foods, and fruits. Images of different kinds of hamburgers, French fries, chicken strips, subs, and drinks were collected in McDonald's, KFC, Subway and Arby's. Images of the steaks and homemade meals were gathered in local restaurants and users' homes. For diversity and comparison, fruit images were also collected in supermarkets: these images include apples, bananas, pears, peaches, and oranges. In order to test the classification algorithm and volume estimation algorithm, test samples were collected at different restaurants with different combinations of food items and at different times of day. Moreover, food image collection was not limited to the restaurants existing in the database. Images of foods in home were also collected as test cases.

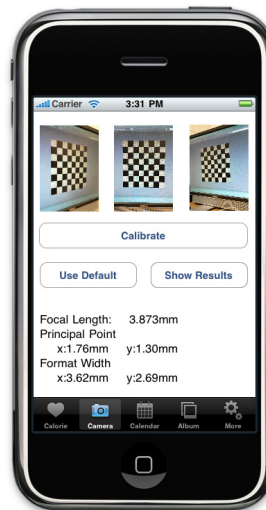
In the experiment, both still images and dynamic videos are taken into consideration. When a piece of video is taken, three frames will be extracted from the video at the start, middle and end. After that, the frames have the same experiment procedure as the still images. The results are combined. Therefore, compared with the image based method, the video based method has an additional frame extraction time.

### 2.7.2 Implementation

The iPhone prototype of DietCam has five main functions, which are organized by a tab bar view controller. The "Calorie" tab shows the main function to calculate calories of a meal ( as shown in Fig. 2.8(a)). When the user taps inside the "Click to Take an Image"



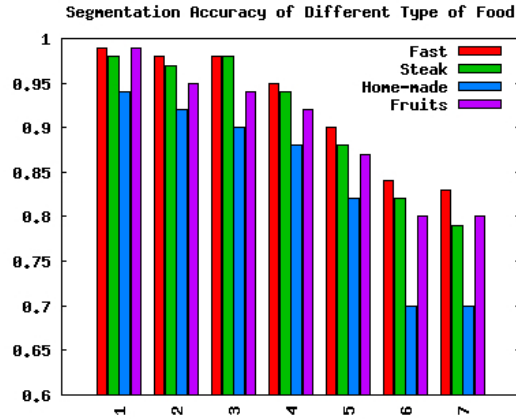
(a) Before calorie calculation (b) After calorie calculation (c) Before camera calibration



(d) After camera calibration

**Figure 2.8:** Calorie calculation function.

button, the camera will be activated and the image taken will be drawn on the button. If the user chooses to take a piece of video instead of three images in the application settings, the video will be activated and three images will be picked automatically from the video. After three images have been taken, the user could tap the “Calculate” button. Then, the food items recognized and calorie information will be displayed (as shown in Fig. 2.8(b)). After this meal, if there are residues in the plate, the user could switch the “Residue” to “On” position and take images again. The information will be stored when “Save” is tapped.



**Figure 2.9:** The segmentation accuracy drops when the number of food items in the plate increases.

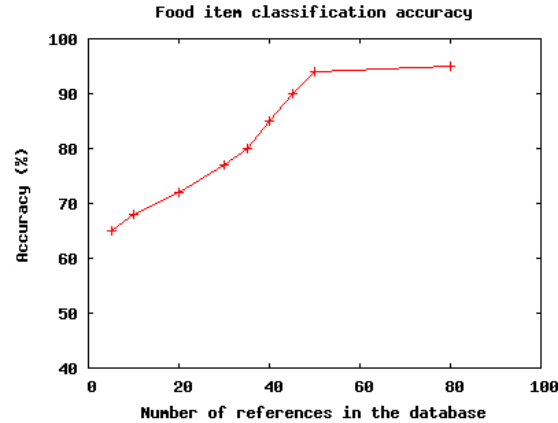
The “Camera” tab leads the user to the camera calibration menu (as shown in Fig. 2.8(c)). The upper three buttons control the camera to take images or shoot videos. After images are taken, the user could tap the “Calibrate” button to calibrate the camera. The results and basic camera information including focal length, principal points, and frame size will be shown in the screen (as shown in Fig. 2.8(d)). Since the application is currently developed for an iPhone, default camera information is provided. In the case of users being unable to access the online calibration board or users wanting to have a quick experience of the application, the default camera information could be read from a property list file.

The “Calendar” and “Album” tabs give the users two diet history viewing options. The “Calendar” menu leads the user to view the history by date when meal records are applicable. The “Album” menu organizes all the food items as a frequency list, and shows them in a table. In the history view, both the meal and the residues will be drawn if applicable. The concrete meal information will be shown below the images.

The “More” tab leads the user to the application information and application settings, where the user can choose to shoot videos or images. There will also be a dietary suggestion function which is still under construction.

### 2.7.3 Recognition Accuracy

The food classifier’s accuracy is affected by many factors, such as fault segmentations, failing classifications to a different shaped food type, misinterpretations between similar shaped food types, and food missing in the database. Therefore, the algorithms were tested



**Figure 2.10:** Food item classification accuracy.

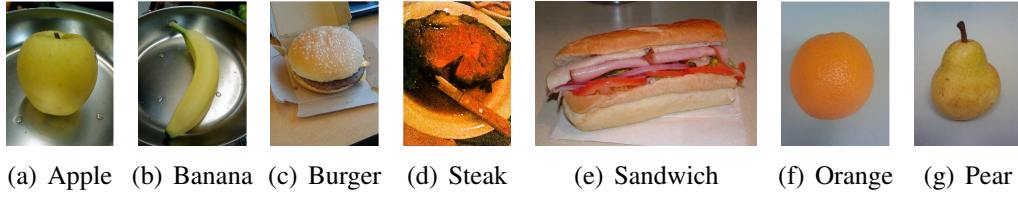
one by one with different types of test cases to cover as many situations as possible. Then, the overall accuracy was evaluated.

The segmentation algorithm was evaluated by testing fast foods, steak meals, homemade foods, and plates of fruits. Fig. 2.9 shows the segmentation accuracy of each kind of meal. It is clear that, when the number of food items increases in the meal, the segmentation accuracy will drop. But it is still acceptable when the number of food items is less than six. Another fact is that the algorithm performs well on fast food, steak meals, and fruits. However, homemade foods are hard to segment accurately. This is because the homemade foods usually do not have a standard pattern.

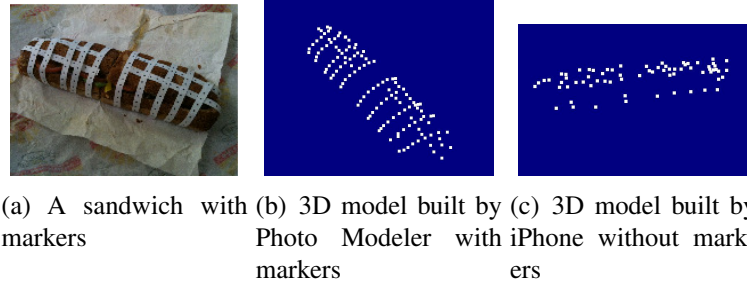
The classification algorithm was evaluated in two steps. In order to examine its ability to classify food items with particular features, the classifier was tested with a certain food item and a given number of references in the database. The results are shown in Fig. 2.10. The accuracy of DietCam increases as the number of reference grows, since the more references there are in the database, the more patterns the database will cover.

Food items with similar appearances were tested, such as cheeseburgers, double cheeseburgers, and hamburgers without cheese. Another test case is veggie subways and big Philly Cheesesteak subways. Table 2.2 shows the results. The sample size is 20.

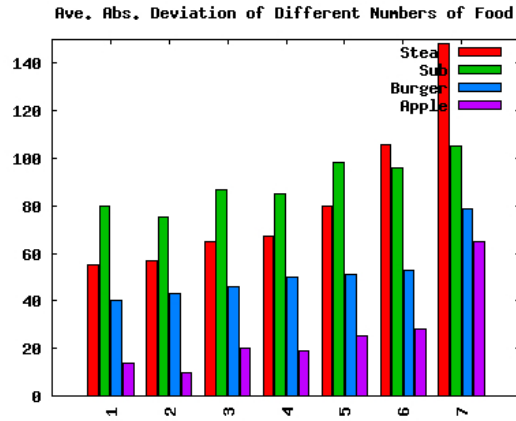
Considering all the above factors, the overall accuracy is still 92%, when the reference number of a food category is larger than 50 and the number of food items to recognize is less than six. This is acceptable since a typical meal usually consists of less than five items. The main resource of the inaccuracies is from the database. If the database covers more references, the accuracy will increase further.



**Figure 2.11:** Food types used to test the volume estimation.



**Figure 2.12:** A sandwich in the experiment.



**Figure 2.13:** Average absolute deviations with increased number of food item:  $cm^3$ .

## 2.7.4 Volume Calculation

The volume calculation algorithms were evaluated with a group of fruits and common food items. More than one food item was put on the plate to simulate a real meal. The algorithms were evaluated to estimate the volume of each item. Fig 2.11 shows the food items used.

The real volumes of the food items were evaluated by two methods. The volumes were

first measured by water displacement. Considering the errors and low accuracy in the measurement process, we measured the volume of food items with the commercial software PhotoModeler[31]. It can build arbitrary 3D models through referencing markers in multiple images. Fig. 2.12 shows a sandwich with markers on it. In the experiment, eight images of this sandwich were taken from different perspectives. After assigning and referencing all the markers in the images, the 3D model of this sandwich could be built. The volume of the 3D model was calculated by PhotoModeler. The real volume value is the average of the value from water displacement and that from PhotoModeler. In order to analyze the accuracy of the 3D model built with the iPhone, the model generated through PhotoModeler was used to compare with that of the iPhone.

The estimated volume values were calculated using methods with predefined shape models and without the shape models respectively. Therefore, it is clear to see the contribution of the predefined models to the volume estimation. In order to evaluate the conclusion confidence of the volume estimator, a large number of food items were tested and the average absolute deviation was calculated on the estimated and measured volumes. The average absolute deviation was calculated according to Eq. 2.9.

$$D = \frac{\sum_{i=1}^n |V_i - M_i|}{n} \quad (2.9)$$

where  $n$  is the number of food items,  $V_i$  is the estimated volume, and  $M_i$  is the measured volume.

Table 2.3 shows the mean values of each kind of food item in the experiment and the estimated values. The sample size of each test was 10. The algorithm was tested by placing more than one item on the plate. The measured actual volumes are presented at the first line followed by the values estimated with two algorithms.

Fig. 2.13 shows the average absolute deviations when the number of food items in the plate increases. From the figure it is clear that the standard deviation increases when the number of food items on the plate grows. It is easy to understand this phenomenon, since occlusions affect the performance of the algorithm. However, another obvious fact is that the algorithm with predefined shape models suffers little from the occlusions caused by the increased number of food items. It gave out confident estimates in the experiments.

**Table 2.2**

Similar food classification accuracy (sample size: 20).

ham	cheese	d. cheese	veggie sub	Philly sub
0.95	0.85	0.85	0.90	0.85



**Table 2.3**  
Measured and estimated food volumes:  $cm^3$  (sample size: 10).

		apple	orange	pear	banana	burger	sub	steak
Measured	mean	310.5	207	221	215.8	678.2	1280.1	288.5
Est. w/o model	mean	275.8	185.4	192	193.3	571.9	948.2	242.5
	AAD	34.7	21.6	29	22.5	106.3	331.9	46
Est. w/ model	mean	286.7	198.4	194.2	204.1	623	1211.7	na
	AAD	23.8	8.6	26.8	11.7	55.2	68.4	na



(a) A bag of fries with calorie information (b) A bottle of soft drink (c) Fries calorie information (d) Coke calorie information

**Figure 2.14:** Calorie monitoring with OCR for labeled food items and drinks.

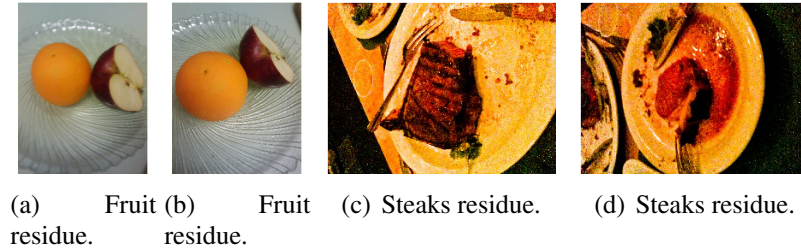
## 2.7.5 Unrecognized food and residues

Since not all the food items could be recognized through appearances, OCR is deployed to help recognize bagged food items. If some food items in the meal are bagged and nutrition information is provided, it will be more accurate to recognize the calorie values on the label. Users only need to take pictures of the labels. Fig.2.14 shows the results of two samples.

The residue of a meal is difficult to recognize since the leftovers might have arbitrary shapes and appearance. Using predefined shape models is not feasible here. Residues recognizable by the cameras were tested; the results are shown in Fig. 2.15. The volumes were estimated by the arbitrary model method. Arbitrary residue classification is one of our future works.

## 2.8 Discussion

Obesity is a complex condition caused by the interaction of many factors such as genetic makeup, secondary effects of medicines, bad emotional states, irregular physical activities



**Figure 2.15:** Food residue recognition.

and unhealthy food choices. DietCam aims to facilitate food intake assessments so as to foster a healthy food choice. The accuracy of DietCam is limited by a few factors including ingredients and recipes. On the one hand, the calorie density of each instance of food ingredient could be different. On the other hand, the recipes for food from different restaurants could be different too. Therefore, in the USDA database[21], the calorie density of each kind of food represents an average calorie density of that kind of food.

From the experiment, DietCam shows practicability when the number of food items in the scene is less than six. The recognition accuracy of 92% shows that it produces a satisfactory result in most situations. The standard deviation evaluation shows at most a  $\pm 20\%$  error of the volume estimation, which means that an estimation of the calories could be made based on the volume estimation and the average calorie density.

## 2.9 Related Work

This section reviews similar research projects and commercial obesity care systems first, followed by related visual recognition and volume estimation algorithms.

Many research and commercial mobile phone applications exist to help address obesity-related challenges. Zhu et al[17, 32] proposed the Technology-Assisted Dietary Assessment project to process food images with a mobile device. However, in that project it is assumed that the plate has to be white, food items in the plate are separated, and users have to take a chessboard-like marker to calibrate the images, all of which makes it seldom serve the purpose in real settings. The advantage of this paper is setting users free from any extra operations besides shooting pictures through automated food recognition and 3D volume reconstruction. The food arrangement is not restricted. Mobile phones and Internet services have also been used in diet monitoring such as creating appropriate meal plans and physical activity schedules[7], recording food choices[14], and tracking their real-time calorie balances[8]. Smart phones and wearable sensors are also used to stimulate activities, such

as Chick Clique[12] and TripleBeat[13]. Fujiki et al.[33] encourage users to increase non-exercise activity with a mobile phone equipped with an accelerometer. Patrick et al.[1] discuss health-related applications like reminders, patient monitoring, and web-based services with mobile phones. Some commercial applications have appeared in recent years, such as MyFoodPhone by Sprint[9], Diet Fitness Diary by Verizon[11], and Sensei[10]. These existing academic and commercial systems rely heavily on manual data analysis and labor intensive user interaction. Automatic dietary monitoring has been developed by analyzing chewing sounds detected by on-body sensors[6]. However, it is not possible for people to wear sensors all the time and it is not accurate enough to estimate the food intake only with chewing sounds.

Object recognition has been a well-studied problem in computer vision. Studies in this area have been mainly focused on two directions. The first is to identify objects from a single view. Most work in this direction has been based on analysis on image patches that are invariant to image scaling, affine transformation, and visual occlusion. The image patches are typically extracted by an interest point detector[22, 23] and described by a patch descriptor. The most popular detector and descriptor is SIFT [24, 25].

The second direction is to recognize objects from multiple views. Camera networks are set up to acquire images of a common object from multiple viewpoints; the ability to jointly recognize object classes from multiple views is promising. When multiple images share a set of features on the same objects, correspondences can be established across camera views, which motivated the SIFT framework[24]. Cheng et al.[34] proposed obtaining a vision graph by matching SIFT features. In wireless camera networks, multiple-view SIFT feature selection was studied by Christoudias et al.[35]. Compressive sensing theory is used to encode SIFT-type object histograms in a distributed manner[36]. Object classification algorithm in this paper based on single view object recognition combined with Bayes decision theory to classify the food classes, which differentiates our work from all other object classification algorithms.

There are two methods to reconstruct 3D object models. The first is reconstructing 3D models from multiple views based on triangulation and projection. Techniques mostly used are stereo vision[37] and structure from motion (SfM) [38]. Seitz et al.[39] provided a good classification, comparison, and evaluation of multi-view stereo reconstruction algorithms. The typical steps involved in SfM solution are extracting features from pictures, finding an initial solution of the structure, and the motion of the camera, extending the solution with optimization, calibrating the cameras, finding a dense representation of the scene, inferring the geometric, textural and reflective properties of the scene. Kien[40] reviewed the basic routine for 3D reconstruction from video sequences. The challenges faced by geometry reconstruction are the inaccuracies caused by the conversion from 2D measurements to 3D models.

Another method is reconstructing 3D models from a single still image with inference techniques. With this method, triangulation and geometry computation is no longer used. But the visual cues in the image are more helpful. Saxena et al.[41] used a Markov random field (MRF) to infer a set of “plane parameters” that capture both the 3D location and 3D orientation of the patch. A 3D reconstruction method from a small number of sparse monocular visions was also presented in[42]. Supervised learning techniques were utilized to infer the relation between image features and location/orientation of the planes. By utilizing prior knowledge of a class of scenes, a probabilistic framework for reconstructing scene geometry was used in[43]. This paper uses a triangulation based model and method proposed in [20].

## **2.10 Conclusion and Future Work**

This paper has presented DietCam, a camera phone based automatic food intake monitoring system aiming to help prevent obesity. The advantage is to automate calorie estimations of a meal with few user interventions. A feature based food classification approach and a multiple-view method to obtain the calorie values of food items through 3D model reconstruction (to calculate the volume) and occlusion reductions have been developed. Food databases consisting of personal and global databases have been constructed. A prototype of DietCam has been implemented on the iPhone platform. The evaluation results show that DietCam performs well at classifying foods even with similar appearances.

Future work will focus on the database construction, image features, and portability to other popular mobile platforms. At present, the global food database is still collected manually. An automatic image collection method is under development on the Internet to accelerate the process. Other work is to increase the classification accuracy by investigating new visual features for food images. An image feature descriptor combining food shapes, colors and textures is under development. We are also investigating extending the portability of DietCam to Windows Mobile, Android, RIM and Symbian platforms.

# References

- [1] K. Patrick, W. Griswold, F. Raab, and S. Intille, “Health and the mobile phone,” *American Journal of Preventive Medicine*, vol. 35, no. 2, pp. 177–181, Aug 2008.
- [2] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [3] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [4] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [5] A. Ershow, J. Hill, and J. Baldwin, “Novel engineering approaches to obesity, overweight, and energy balance: public health needs and research opportunities,” *Engineering in Medicine and Biology Society, IEEE Annual International Conference of*, pp. 5212–5214, Jan 2004.
- [6] O. Amft, “Automatic dietary monitoring using on-body sensors, detection of eating and drinking behaviour in healthy individuals,” *PhD dissertation, Swiss Federal Institute of Technology Zurich, 2008*, Jan 2008.
- [7] “USDA’s center for nutrition policy and promotion, mypyramid,” *[Online]*. Available: <http://www.mypyramid.gov/>.
- [8] C. Tsai, G. Lee, F. Raab, G. Norman, T. Sohn, W. Griswold, and K. Patrick, “Usability and feasibility of pmeb: A mobile phone application for monitoring real time caloric balance,” *Mobile Networks and Applications*, vol. 12, no. 2-3, pp. 173–184, Jun 2007.
- [9] “My food phone,” *[Online]*. Available: <http://www.mycanutrition.com/>.
- [10] “Sensei diet program,” <http://www.sensei.com/sensei/>.
- [11] “My food diary,” *[Online]*. Available: <http://www.myfooddiary.com/>.

- [12] T. Toscos, A. Faber, S. An, and M. Gandhi, “Chick clique: persuasive technology to motivate teenage girls to exercise,” *Human factors in computing systems, CHI extended abstracts on*, pp. 1873–1878, 2006.
- [13] R. Oliveira and N. Oliver, “Triplebeat: enhancing exercise performance with persuasion,” *Human computer interaction with mobile devices and services, International conference on*, pp. 255–264, 2008.
- [14] S. Reddy, A. Parker, J. Hyman, and J. Burke, “Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype,” *Embedded networked sensors, workshop on*, pp. 13–17, Jan 2007.
- [15] Ø. Trier, A. Jain, and T. Taxt, “Feature extraction methods for character recognition-a survey,” *Pattern recognition*, vol. 29, no. 4, pp. 641–662, Jan 1996.
- [16] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2161–2168, 2006.
- [17] I. Woo, K. Otsmo, S. Kim, D. Ebert, E. Delp, and C. Boushey, “Automatic portion estimation and visual refinement in mobile dietary assessment,” *Computational Image VIII, Proceedings of the SPIE*, vol. 7533, pp. 1–10, Dec 2010.
- [18] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” *Computer Vision, IEEE International Conference on*, pp. 666 – 673, 1999.
- [19] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision, 2nd ed,” *Book, Cambridge University Press*, 2004, Jan 2004.
- [20] F. Kong and J. Tan, “A 3d object model for wireless camera networks with network constraints,” *Distributed Smart Cameras, Third ACM/IEEE International Conference on*, pp. 1–8, Aug 2009.
- [21] “U.s. department of agriculture, agricultural research service. 2009.” *USDA National Nutrient Database for Standard Reference, Release 22. Nutrient Data Laboratory Home Page*, <http://www.ars.usda.gov/ba/bhnrc/ndl>, 2009.
- [22] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, Jan 2004.
- [23] K. Mikolajczyk, B. Leibe, B. Schiele, M. Syst, and G. Darmstadt, “Local features for object class recognition,” *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1792 – 1799, 2005.
- [24] D. Lowe, “Object recognition from local scale-invariant features,” *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1150 – 1157, 1999.

- [25] S. Helmer and D. Lowe, "Object class recognition with many local features," *Computer Vision and Pattern Recognition Workshop, Conference on*, pp. 187–195, 2004.
- [26] R. Bolle, J. Connell, N. Haas, R. Mohan, and G. Taubin, "Veggie vision: A produce recognition system," *Automatic Identification Advanced Technologies, IEEE Workshop on*, pp. 35–38, Feb 1997.
- [27] J. Salvi, X. Armangue, and J. Battle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern recognition*, vol. 35, pp. 1617–1635, 2002.
- [28] G. Strang, "Introduction to linear algebra, 3rd ed," *Wellesley-Cambridge Press*, 1998, 1998.
- [29] R. Taylor and O. Zienkiewicz, "The finite element method for solid and structural mechanics," *Butterworth-Heinemann*, 2005.
- [30] "Stockfood - the food image agency. food pictures for professionals," [online] <http://www.stockfood.com>.
- [31] "Photomodeler: Accurate and affordable 3d modeling-measuring-scanning," <http://www.photomodeler.com/index.htm>.
- [32] F. Zhu, A. Mariappan, C. Boushey, D. Kerr, K. Lutes, D. Ebert, and E. Delp, "Technology-assisted dietary assessment," *Computational Imaging, Proceedings of the IS&T/SPIE Conference on*, pp. 1–10, Jan 2008.
- [33] Y. Fujiki, K. Kazakos, C. Puri, and P. Buddharaju, "Neat-o-games: blending physical activity and fun in the daily routine," *Computers in Entertainment*, vol. 6, no. 2, pp. 1–22, 2008.
- [34] Z. Cheng, D. Devarajan, and R. Radke, "Determining vision graphs for distributed camera networks using feature digests," *Advances in Signal Processing, EURASIP Journal on*, vol. 2007, no. 1, pp. 220–231, Jan 2007.
- [35] C. Christoudias, R. Urtasun, and T. Darrell, "Unsupervised distributed feature selection for multi-view object recognition," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2008.
- [36] A. Yang, S. Maji, C. Christoudias, and T. Darrell, "Multiple-view object recognition in band-limited distributed camera networks," *Distributed Smart Cameras, Third ACM/IEEE International Conference on*, pp. 1–8, Aug 2009.
- [37] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Computer Vision, International Journal of*, vol. 47, no. 1/2/3, pp. 7–42, Jan 2002.

- [38] T. Jebara, A. Azarbayejani, and A. Pentland, “3d structure from 2d motion,” *IEEE Signal Processing Magazine*, vol. 16, no. 3, pp. 66–84, Jan 1999.
- [39] S. Seitz, B. Curless, J. Diebel, and D. Scharstein, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, pp. 519 – 528, 2006.
- [40] D. Kien, “A review of 3d reconstruction from video sequences,” *Intelligent Sensory Information Systems technical report, University of Amsterdam, 2005*, 2005.
- [41] A. Saxena, M. Sun, and A. Ng, “Learning 3-d scene structure from a single still image,” *Computer Vision, IEEE International Conference on*, pp. 1–8, 2007.
- [42] A. Saxena, M. Sun, and A. Ng, “3-d reconstruction from sparse views using monocular vision,” *Computer Vision, IEEE International Conference on*, pp. 1–8, 2007.
- [43] W. Zhang and T. Chen, “A probabilistic framework for geometry reconstruction using prior information,” *Image Processing, IEEE International Conference on*, vol. 2, pp. 529–532, Jan 2007.



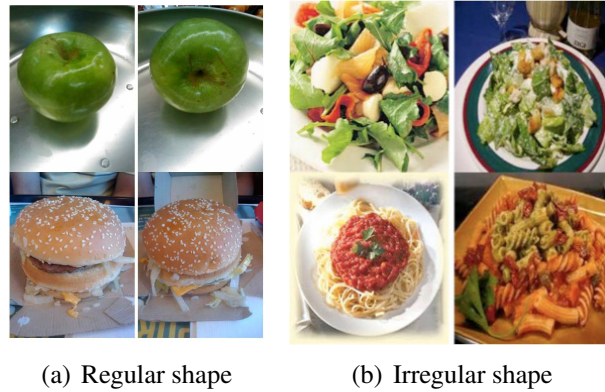


## **Chapter 3**

# **DietCam: Multi-View Regular Shape Food Recognition with a Camera Phone**

### **Abstract**

The purpose of this paper is to develop an automatic camera phone based multi-view food classifier as part of a food intake assessment system. Food intake assessment is important for obesity management, which has shown significant impacts in public healthcare. Conventional dietary record based food intake assessment methods exhibit insufficient popularity due to their low accuracy and high dependence on human interactions. Image based food recognition appears recently. But it is still under development and far away from field applications. This paper presents DietCam, a camera phone based application to recognize food intakes automatically from multiple perspectives. Food recognition from images is afflicted currently with a low recognition accuracy caused by the uncertainties of food appearances. The deformable nature of food items together with the complex background environment makes the problem even harder. DietCam utilizes a multi-view recognition method which separates every food item through evaluating the best perspective and recognize each of them from multiple images with a probabilistic method. The recognition accuracy is increased significantly compared with single-view recognition methods. A prototype of DietCam has been implemented on iPhone. In the field experiments, it shows an accuracy of 91% for regular shape food items.



**Figure 3.1:** Food shapes.

### 3.1 Introduction

Obesity has been a severe public health challenge to the general population and social welfare in many developed countries[1, 2]. In the past three decades, the obesity rate in U.S. has increased significantly[3], resulting in serious consequences such as diabetes, stroke, heart disease even cancers. Food intake assessment is significant for obesity management. However, few people are aware of their food intakes and they are not willing to assess food intakes. The reason is the burdensome assessment methods and a lack of real-time feedback with these methods. Traditional food record or food diary methods require manual records of the food type and the portion of the food taken, and the accuracy is limited by human estimations of the food portion. Computer aided and automatic food intake assessment methods do not suffer from manual records or inaccurate human estimations. Cameras and computational resources could recognize foods and assess food intakes. However, image-based food recognition is still under development.

Visual object recognition has been a popular research topic in computer vision for many years. Topics could be classified into instance recognition, category recognition and a special case between them, face recognition. Instance recognition and face recognition usually are broken down to object detection and object recognition and they have been the most successful recognition applications. The most challenge version of visual recognition is the general category recognition, which is still at the level of a two-year old child [4]. The difficulties come from the occlusions in the cluster of objects and the variability intrinsic within a category. Due to the complex non-rigid and extreme variations in shapes and appearances, it is unlikely to perform matching against a database of examples.

Food recognition is a special case of category recognition, with larger degrees of variations. The appearance of any particular meal is always affected by many factors such as

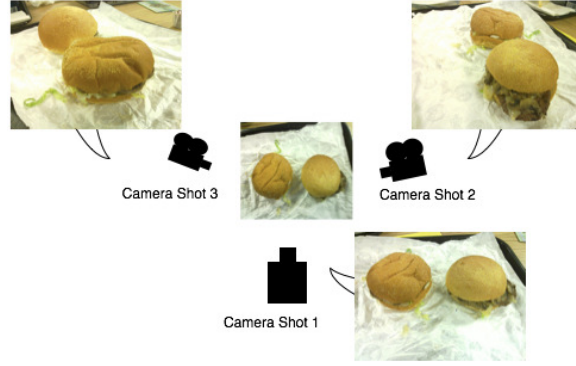
ingredients, cooking methods, cutting patterns, ingredient positions, occlusions, lighting conditions, etc. These factors are so complex that even meals of the same category always have different appearances. Contradictively, different types of food could have similar appearances that are difficult to distinguish through human eyes. These intra-class differences and inter-class similarities make food recognition an extremely hard category recognition problem.

Current category recognition methods represent an image as a collection of feature descriptors. These descriptors describe the image features at special locations, such as corners, edges or certain blob areas. Objects in the image are classified through counting the descriptors statistically. More sophisticated approaches use not only the statistical properties of the descriptors, but also the geometric relationships between them. Certain objects have parts arranged in a dedicated way. Besides these methods, recent methods also recognize categories through understanding the context and environment, as context plays an important role in human object recognition. All the above methods have gained certain successes in recognizing categories such as buildings, furnitures, landmarks, cars and animals. However they are still inferior and not enough to recognize food categories.

Food recognition is a challenging problem. The challenges to evaluate food intakes include occlusions, food segmentation, classification, and volume. Image based food recognition has been proposed recently[5–8]. However, due to the immaturity of current approaches to recognize deformable food appearances, intra-class uncertain food appearances, and limitations of perspectives and occlusions, food recognition still exhibits low accuracies.

Our method solve these research challenges through investigating a multi-view graphical model based food recognition method. Meanwhile, the multi-view method is capable for food volume estimation, which is detailed in another technical paper. We divide the food categories into regular shape and arbitrary shape, which are shown in Fig. 3.1, and develop specific recognition algorithms for each of them respectively. The regular shape food such as fruits, hamburgers and pizzas, usually has a certain shape pattern. Irregular shape food always has deformable shapes, such as noodles, pastas, salads and other kinds of meals. The multi-view method this paper will present and evaluate is a new food recognition algorithm for regular shape food items. The method to recognize irregular shape food items is different, it will be presented in another paper.

The recognition method is motivated by the observation in a camera network that a cluster of objects will be easier to segment from a certain viewpoint and compared with single viewpoint object recognition, recognition accuracy could be increased with images from multiple viewpoints. Consider a scene consisting of two food items as shown in Fig. 3.2, it will have more chances to separate these two items from the camera where the distance between the object projections in the image is farther than those from other cameras. Sim-



**Figure 3.2:** Right perspective could help food segmentations. In the scene, the two cheeseburgers will be easier to segment from images of camera 1. But from the perspectives of camera 2 and 3, they are clustered and partial occluded. The goal is to segment food items with multiple viewpoints and model the occlusions.

ilarly, the recognition results from one camera could be verified by other cameras so that the recognition accuracy will be enhanced. Another potential benefit multiple views provide is the possibility to recover the 3D information and estimate food volumes, which are valuable for food intake assessment.

Despite the observation in the multiple-viewpoint environment, it is still challenging to apply this intuitive method to food recognition for several reasons. First, given a number of images, the proposed method requires the knowledge that which one has the best viewpoint for segmentation, especially when the background is complicated and the appearances of the food items could be uncertain. Second, it is necessary for the proposed method to tell occlusions from fault recognitions. If an object is detected in image *A*, but disappeared in image *B*, the classifier has to be able to tell that the inconsistency in image *A* and *B* is caused by occlusions in image *B* or fault recognitions of image *A*.

Concerning the above challenges, the proposed method models food geometric locations and classifies each food item with a graphical model. It can be summarized as follows. The camera phone will be used to take a short video surrounding the food items or three images. If a short video is taken, three frames that are from viewpoints will be extracted from the video. Interest points in these images will be detected, classified and used to calibrate the camera positions through detecting the corresponding points. Our approach segments each food item through categorizing interest points and inferring the distance between interest points of the same category. After the segmentation, the probability of the existence of a specific food item is modeled as a joint probability distribution of this item in the three images, with the concerning in the three images of occlusions, viewpoints and food item appearances. Compared with single viewpoint food recognition, our approach enhances the recognition results significantly.

This paper has two main contributions. First, it complements current category recognition literature with a new method. It provides a new means to integrate the features and their geometric relations. Second, it contributes an integrated multi-view object recognition framework, where the appearances of the object are not assumed and occlusions are modeled. In this framework, the contribution of every viewpoint is adaptive to the contents of the images. With such a framework, not only foods but other deformable objects could be recognized more accurately.

## 3.2 Related Work

Food intake assessment has been a popular research topic in biomedical and health related areas for years. People recognize its importance and try to find methods to evaluate dietary intakes accurately. A simple and widely used method is food diaries and records [9–11]. People have to write down the food types and estimate the volumes of each type of food. It is applicable to most people since it does not require any professional knowledge. However, it is limited by the roughness of human estimations[12]. Similar methods that suffer from the same drawbacks also include dietary histories [13], and food frequency questionnaires[14, 15].

Concerning about the inaccuracy caused by human estimations, researchers develop methods to monitor food intakes inside human organs. Typical methods include biological assessment and chemical analysis. Biological assessments, e.g. doubly-labelled water [16], plasma carotene [17], etc., monitor food intake through the introduction of biomarkers[18]. The chemical analysis methods evaluate the dietary intake through tracking selected elements [19]. Both the biological and chemical methods report the validation and accuracy in food intake assessment [20]. However, these methods could only be used in the lab environment and are not available to everyone.

Automatic and accurate image-based food intake assessment has not appeared. But single-view object recognition has been a research topic in computer vision for decades and it is still a challenging problem. Techniques developed to recognize objects from images could be categorized into three groups according to target applications, instance recognition, identity recognition and category recognition. The first two are the most successful applications of all the three. But category recognition is still an unsolved problem, even at a level of two-year old child [4].

Instance recognition involves re-recognizing a known 2D or 3D rigid shape object, but potentially from a different view point, and/or under different lighting conditions, and/or

with different backgrounds, and/or with partial occlusions, and/or from different distances. Rigid shape object recognition usually uses template-based methods[21–23], which exhibit good performance for single object recognition, e.g. cars[22, 23]. An important limitation of these methods is their inflexibility to capture the variances of object appearances. Templates are restricted to rigid shapes that lack information on object transformations. Hough transform has been utilized for transformation based instance recognition [24]. It tries to estimate the parameters of a transformation that defines a mapping of the model point set to the point set derived from the scene image. Besides template-based method, another method classifies an object through extracting 2D sparse features and matching them to a feature database. Several feature detector and descriptor have been proposed in the last decade, the most popular is SIFT-based features[25].

Identity recognition is similar to instance recognition with respect to that it also re-recognizes known objects such as faces, irises and finger prints. But it is more difficult since it has to detect the object first and the object it needs to recognize is deformable.

Category recognition involves recognizing objects belonging to extremely varied categories, such as animals, furnitures, flowers, etc. Food classification also belongs to this type. Typical methods include bag-of-words, part based models, and context understanding. Bag-of-words method treats image features as visual words and compares the distribution of the words with those found in training images. Usually it involves key-patch detection, feature extraction, histogram computation and histogram classification. There are several popular patch detectors. Harris-Laplace region [26] detectors are used to find corner-like structures in the image. DoG regions [27] are proposed to detect blob like structures. In addition, other detectors appear such as Hessian-Laplace regions [28], Salient regions [29] and MSER [30]. Popular patch descriptors include SIFT [25], PCA-SIFT [31], Moment based and cross correlation. There are many classifiers proposed to classify the features, such as k-means, hierarchical k-means[32], randomized k-d tree, geometric hashing, nearest neighbor, Bayesian, boosting and support vector machine. Bag-of-words method is limited by the fact that it only considers the existence of certain features, rather than the geometric location of these features. But cases exist that objects in the image cannot be recognized correctly without considering their geometric locations. Part based object recognition algorithms extract edges in the image and match them to part templates. The object is recognized through considering the category and the spatial distribution of the parts. Besides the location of the parts, sometimes the context around the object also gives a good clue what category the object belongs to [33].

Multi-view object recognition appears recently as an approach to increase the recognition accuracy of single-view object recognition. The basic idea is to recognize objects jointly from more than one perspectives. The extra perspectives provide the possibility that more distinct features of the object could be extracted so that the recognition accuracy could be improved. An implicit assumption behind this idea is the certain appearance of the object in

all the images. Multi-view recognition has been applied in vision-aided robot arm control and visual recognition. In vision-aided robot arm control, it is used to recognize objects in a complex environment with occlusions. The multi-view method enables the robot arm to recognize and grab the target accurately [34]. In multi-view object recognition, researches are mainly about how to fuse the recognition results from different views. Compressive sensing has been utilized to compress the features from different images and then classify the features in a central server [35]. Bayesian classifiers have also appeared to fuse the recognition results from single views [36].

Food recognition is a specific case of category recognition. Currently, it still has not drawn significant attentions and has no specific solutions. Some researchers try to recognize foods from images with existing methods from instance recognition and general category recognition. But the accuracy is still not high enough for applications. For example, Martin et al. use a general color histogram of the image to recognize food items[5]. Wu et al. present interest points (SIFT) based method to recognize fast food[6]. But the accuracy is under 70%. Zhu et al. detect food textures from the image first and utilize the histogram of textures to classify food items[7]. Yang et al. extend the food texture histogram to the pair-wise texture distributions for fast food recognition[8] and the accuracy is still only 78%.

Compared with these existing food recognition methods and general object recognition algorithms, the method presented in this paper is differentiated with a multi-view recognition framework without the object appearance assumption. All the existing food recognition methods are single view algorithms derived from histogram based methods and show inferior accuracies. The texture-based methods exhibit better recognition accuracy than interest points based methods. However, texture detection will be affected by lighting conditions and background clusters. Another fact is that texture based methods are only applicable to single food item recognition, and it does not have the segmentation ability. The multi-view method has not been explored in food recognition, and the assumption of general multi-view recognition algorithms that the object must appear in the image makes it impossible to utilize them directly for food recognition. We develop a new multi-view recognition algorithm for food recognition, where more than one food object could exist in the scene at the same time and they could be occluded by each other from certain perspectives.

### **3.3 Multi-View Food Recognition**

In this section, we present DietCam. It starts from detecting representative feature points in the images and classifying them with single-view object recognition algorithm. With the feature points in each image, the camera will be calibrated to find geometric relations be-



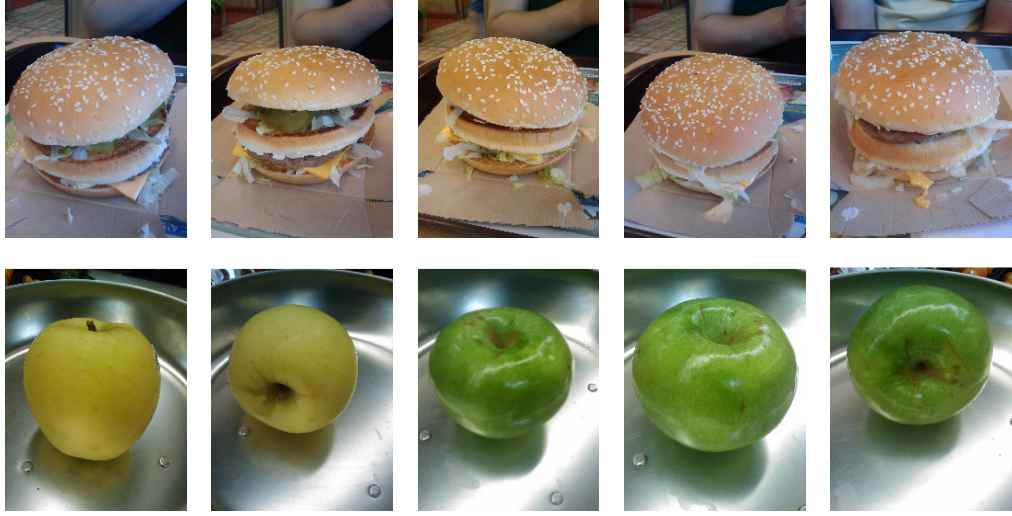
tween perspectives. The food items will be segmented based on the feature point categories and their locations. The same type of food is separated based on the object distance between interest points. In the three images, the object distance of two interest points is modeled as their maximum in-image distance, which is modeled as the number of object boundaries between their locations in the image. If the object distance between interest points is small enough, they belong to the same food item. After the segmentation, the recognition results from all the images will be integrated together to determine the existence of a food item. The existence of a food item is modeled as a joint probability distribution of this item in the three images. Concerning the geometric relations between images, if a food item appears in one image, it has a probability to appear at a certain location in the other image. Based on this probability, occlusion could be modeled and classification results could be verified.

### 3.3.1 Food features

Before classifying food items from the images, we detect and extract local feature points in every image and classify these features based on an existing feature database. We choose a difference of Gaussian region detector and Scale Invariant Feature Transform (SIFT)[25] descriptor as the feature since the results are invariant to lighting, scaling, affine transmission and partial visual occlusions. The locations of the feature points are detected in the image scale space. The scale space of an image is defined as a function  $L(x, y, \sigma)$ , which is produced from the convolution of a variable-scale Gaussian,  $G(x, y, \sigma)$ , with an input image  $I(x, y)$ :  $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$ , where  $*$  is the convolution operation in  $(x, y)$ . The feature points are detected as the local maxima in  $L(x, y, \sigma)$ . Therefore, the difference of Gaussians are computed,  $D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$ . The features are described by a  $4 \times 4$  image gradient histogram from  $16 \times 16$  sample arrays. Every image could have hundreds of SIFT features.

The feature database consists of SIFT features extracted from a large number of labeled training images. Concerning the fact that the same kind of food could have different appearances and even one food item may have different looks from different perspectives, a number of training images of the same type of food are collected and features from different perspectives of these images are extracted and stored in the database. Each of the training images contains only one food item, so that features of the image will be a clean description of the food item contained, rather than messed up by other food types. We use lab still images of a recently published food image database PFID[37] as part of the training images and we also collect a large set of samples. Fig. 3.3 shows an example of part training set of cheeseburgers and apples.

A food feature is classified in the database to the categories of the features that have the



**Figure 3.3:** Training set of cheeseburgers and apples from different perspectives.

minimum Euclidean distances with a probability, shown in Fig. 3.4. We choose the top five nearest neighbors as the candidates. The probability the feature  $f$  is the same category with neighbor  $i$  is defined as

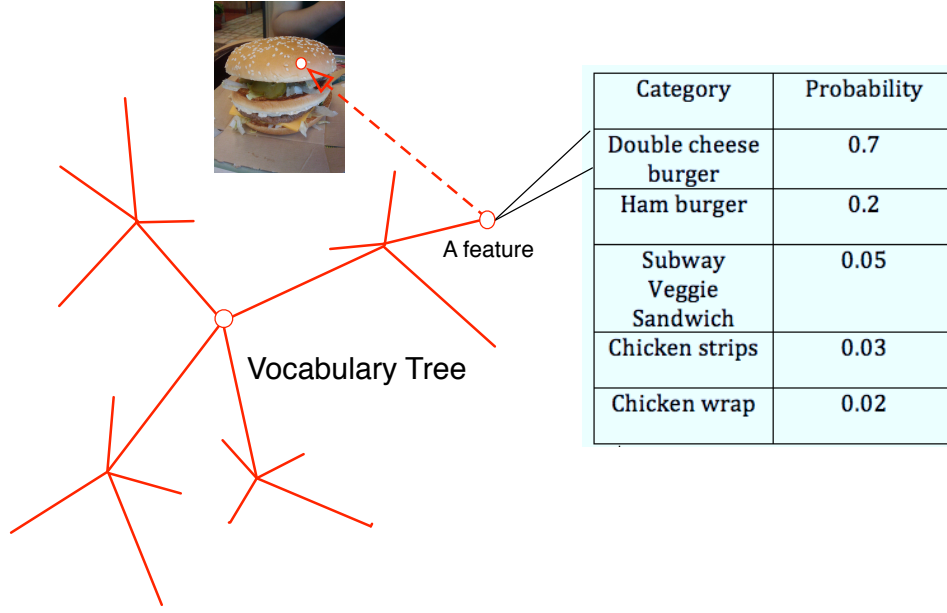
$$P(C(i)|f) = 1 - \frac{distance(f,i)}{\sum_{j=1}^5 distance(f,j)} \quad (3.1)$$

where  $C(i)$  is the category of the  $i$ th nearest neighbor,  $distance(f,i)$  is the Euclidean distance from  $f$  to the  $i$ th nearest neighbor. Looking for the nearest neighbors in a large database is time consuming. Therefore, a vocabulary tree [38] is built in the database to organize the features to reduce the search time. The vocabulary tree is built with hierarchical k-means. Then the complexity of looking for the nearest neighbor is reduced from the number of the features to the depth of the tree.

### 3.3.2 Camera calibration

Food items could be classified from an image with the nearest neighbor method above. However, the accuracy is limited by the perspectives and occlusions. DietCam increases the recognition accuracy through result verifications from multiple viewpoints. It considers the images are taken by three cameras at a synchronized time. The locations of the cameras and the geometric relation between them could be calibrated from the images.

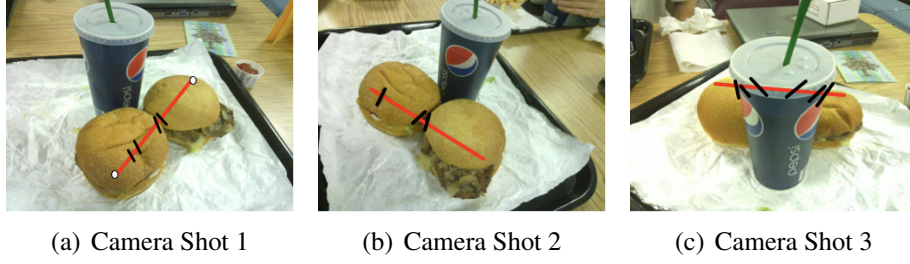
The parameters of a camera can be represented as  $P = A[R \ T]$ , where  $A$  is the intrinsic



**Figure 3.4:** A tree structure organizes all the features in the feature space. A new feature from an image of double cheese burger is matched to a leaf of the index tree. The table on the right shows the possible categories the new feature belongs to with probabilities.

parameter matrix,  $R$  and  $T$  are extrinsic rotation and translation parameters. The intrinsic matrix could be calibrated offline and a proper matrix could be selected according to the cell phone's brand and model. In order to make the users unconcerned with the calibration process, epipolar geometry is used because it only requires correspondences in the image to calibrate the extrinsic parameters. The correspondences between a pair of images could be found through matching SIFT features. In a pair of images, one image is chosen as the reference image and its camera matrix can be chosen as  $P = A[I \ 0]$  where  $I$  is a  $3 \times 3$  unit matrix. By doing this, the world coordinate system is decided. After the mobile phone is moved to take another picture, the new camera matrix related to the world coordinate system is decided as  $P' = A[R \ T]$ . The parameters  $R$  and  $T$  can be estimated correspondences between these two views. In epipolar geometry, the essential matrix  $E$  encapsulates the projection relationship between two intrinsically calibrated cameras. It has the property  $pEp' = 0$  where  $p$  and  $p'$  are correspondent points in two views. According to its definition  $E = [T]_{\times}R$  where  $[T]_{\times}$  is the skew-symmetric matrix of  $T$ , the parameters  $R$  and  $T$  can be estimated by singular value decomposition[39].

If the food volume is concerned, for example food volume estimation, the scale information will be critical. We propose a credit card method to calibrate the scale of the scene. When taking images or videos around the food items, a credit card is put by the side of the food items. Since the dimension of credit cards is known and consistent, through detecting the



**Figure 3.5:** Perspective distances of two cheese burger features from three perspectives. The black lines indicate the edges that are between two features.

corners of the credit card, the scale of the food items could be estimated.

### 3.3.3 Perspective distance

We attempt to segment features of each food item from the other's. Features belonging to different types of foods could be separated from the classification process. But it is hard to separate those belonging to the same type of food, because of the unpredictable boundaries between food items and the missing geometric information of feature points in a single image. We design a new technique and name it perspective distance, which models the likeliness that a pair of features belongs to the same food item.

Perspective distance reflects the geometric relation between two features concerning their appearances in all the possible perspectives. Given a perspective and the image from that perspective, the perspective distance of two features is defined to be greater when the two features have more object boundaries detected between them.

$$D_{image}(f_1, f_2, I) = boundary(f_1, f_2, I) \quad (3.2)$$

Fig. 3.5 shows an explanation of the perspective distances of a scene consisting of two cheeseburgers and a drink from three perspectives. The boundaries of food items in the image are detected with a standard Canny edge detector[40]. Since edges inside a food item boundary would also be detected, the number of boundaries will be no more than the number of edges detected, i.e.  $boundary(f_1, f_2) \leq canny(f_1, f_2)$ . Therefore, without losing the generality, we use the number of edges to represent the perspective distance approximately.

A pair of features appearing in one image could also appear in other images from differ-

ent perspectives. The correspondent features will be found through feature matching and camera parameters calibrated in section 3.3.2. The scene perspective distance between two features is the summation of the perspective distances from all the perspectives. For example, if we have three images around a cluster of food items, and a pair of features of the same type of food appears in all of the three images, the perspective distance is defined

$$D_{scene}(f_1, f_2) = \sum_{I=1}^3 D_{image}(f_1, f_2, I). \quad (3.3)$$

A threshold  $T$  is learned from a set of food images. If  $D_{scene}(f_1, f_2)$  is larger than  $T$ , then  $f_1$  and  $f_2$  belong to different food items. In the experiment,  $T$  is learned and set to 8.

### 3.3.4 Multi-view food recognition

Classifying a food image from a single viewpoint with the feature points would be inaccurate, since occlusions could block key food ingredients. We convert the single viewpoint recognition to multiple viewpoints recognition, which considers food appearances from more perspectives.

In section 3.3.1, the features have already classified to possible categories  $C$  with a probability  $P(C_i|f)$ . In this section, we concern a set of features  $F$  that belongs to the same food item, which is segmented in section 3.3.3. The probability  $P(C_i|F)$  that  $F$  belongs to category  $C_i$  is a joint probability concerning its appearances in all the three images.

The typical method used in the literature of multi-view object recognition is Bayesian classification, where it is assumed that the object appears in all the images. Let's see what would happen if we use a Bayesian classifier to classify multiple food items from multiple views.

For a Bayesian classifier, classification is achieved by finding a category  $C_i$  so that given a set of features  $F$  the probability  $P(C_i|F)$  is the largest compare with other categories. In this multi-view food recognition problem, the set of features is observed from three images. The projections of  $F$  in the images are  $I_1(F)$ ,  $I_2(F)$  and  $I_3(F)$ . The probability of category  $C_i$  is

$$P(C_i|F) = P(C_i|I_1(F), I_2(F), I_3(F)). \quad (3.4)$$

According to the Bayes rule,

$$\begin{aligned}
P(C_i|F) &= P(C_i|I_1(F), I_2(F), I_3(F)) \\
&= \frac{P(I_1(F), I_2(F), I_3(F)|C_i) \times P(C_i)}{P(I_1(F), I_2(F), I_3(F))}
\end{aligned} \tag{3.5}$$

In a naive Bayesian classifier, it is assumed the three images are independent, then the probability could be written as

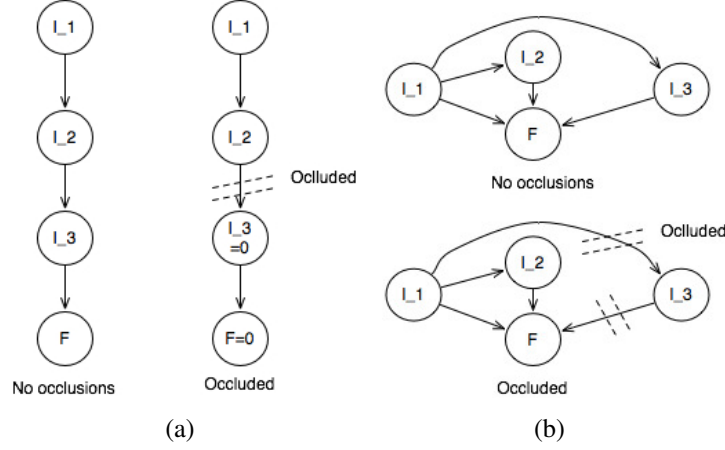
$$\begin{aligned}
P(C_i|F) &= P(C_i|I_1(F), I_2(F), I_3(F)) \\
&= \frac{P(I_1(F), I_2(F), I_3(F)|C_i) \times P(C_i)}{P(I_1(F), I_2(F), I_3(F))} \\
&= \frac{\prod_{j=1}^3 P(I_j(F)|C_i) \times P(C_i)}{\prod_{j=1}^3 P(I_j(F))}
\end{aligned} \tag{3.6}$$

where  $P(I(F)|C_i)$  is the prior probability that learnt from the vocabulary tree.  $P(C_i)$  is the probability that a feature is food type  $C_i$  out of all the possibilities. If no prior knowledge is available, equal prior can be specified for each type. Then the features  $F$  could be classified.

However, in this food classification case, it would not work since the three views are not independent and they are related to each other by the geometry occlusions in the scene. Therefore, we cannot assume its naiveness. The denominator  $P(I_1(F), I_2(F), I_3(F))$  could be written as

$$\begin{aligned}
&P(I_1(F), I_2(F), I_3(F)) \\
&= P(I_2(F)|I_1(F))P(I_3(F)|I_1(F))P(I_1(F)).
\end{aligned} \tag{3.7}$$

The value of this equation could be zero when features  $F$  are occluded by other objects from a certain perspective. It means if we choose an arbitrary perspective as the starting point, for example image  $I_1$ , and if the features shown in  $I_1$  are occluded and not visible



**Figure 3.6:** Bayesian classifiers. The chain structure classifier in (a) cannot handle the occlusions. When some features in Image 3 ( $I_3$ ) are occluded, the final classification result would be zero, which means the features do not belong to any category. With the new classifier in (b), the final result is not decided by the occlusions in certain images, but it will take all the images into consideration.

from either  $I_2$  or  $I_3$ , the denominator of the Bayesian classifier could be zero and it would be failed. This case is shown in Fig. 3.6(a). The success of classifying  $F$  depends on the joints of a series of observations from  $I_1$ ,  $I_2$  and  $I_3$ . From this chain like structure, if any part of the chain breaks, the classification would be failed. The broken chain is caused by the occlusions in the scene, which block the appearances of features in other images.

Since a chain like Bayesian network structure suffers from occlusion, we develop a Bayesian network with a structure shown in Fig. 3.6(b). This structure shows both intuitive and reasoned respectable behavior to handle the occlusions. Intuitively, occlusions in one image should not affect the possibility that the features appear in the other images. And the probability of that  $F$  belongs to  $C_i$  should be contributed equally by all the observations from each perspective.

In a reasoning process, let's start from a single view. If there is only one image  $I_1$ , the probability is

$$P(C_i|F) = P(C_i, I_1(F)|F) \quad (3.8)$$

The term  $P(C_i, I_j(F)|F)$  defines the probability  $F$  showing up in image  $I_j$  belongs to category  $C_i$ .  $P(C_i|F, I_1(F))$  is the same as it is classified in section 3.3.1.

When another image  $I_2$  is used from a different perspective to recognize the object, the probability is

$$P(C_i|F) = P(I_1)P(C_i, I_1(F)|F) + P(I_2)P(C_i, I_2(F)|F) \quad (3.9)$$

When other images are added, the probability  $F$  belongs to  $C_i$  is

$$P(C_i|F) = \sum_{j=1}^n P(I_j)P(C_i, I_j(F)|F) \quad (3.10)$$

Different from the first image,  $P(C_i, I_j(F)|F)$  ( $j \neq 1$ ) is not the distribution learned in vocabulary tree. It is related to the results in the first image and is used to model the occlusions and fault recognitions. If  $F$  is occluded in image  $j$ , it shows a small probability that food  $F$  exists in the scene. But in the other two images, where  $F$  is not or partial occluded, the probability  $F$  exists is still high. On the other hand, if  $F$  does not exist, but it is detected in the first image, which is a fault recognition, in the other two images  $F$  still has a small probability to be detected. In this way, the recognition results will be enhanced.

In the image where  $F$  is first detected,  $P(C_i|F, I_1(F))$  is the same as it is classified in section 3.3.1.  $P(C_i|f, I_2(F))$  and  $P(C_i|f, I_3(F))$  are derived from the images with equation

$$P_j(I_j(F)) = \frac{Area(F, I_j)}{Window(F, I_j)}. \quad (3.11)$$

$Window(F, I_j)$  denotes the area of the projection of food item  $F$  in image  $I_j$  if  $F$  is not occluded. It is calculated with feature points  $F$  detected in the first image. A set of correspondent points of  $F$  could be found in the other two images with the geometric relations calculated in section 3.3.2. These points compose a correspondent window in the images.  $Window(F, I_j)$  represents the area of the window.  $Area(F, I_j)$  represents the overlap of the window and the actual area of item  $F$  appeared in image  $I_j$ . It is calculated from the actual feature points belonging to  $F$  detected in the image. Fig. 3.7 illustrates the definition.



### 3.3.5 Object recognition from 3D reconstruction

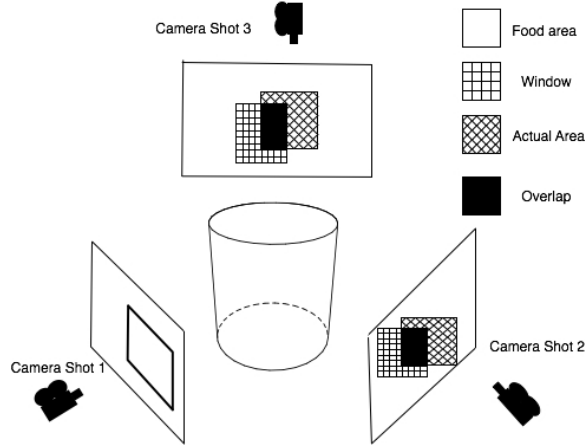
Another benefit that multi-view brings to us is the possibility to reconstruct 3D locations of the image features in the 3D space. It is possible that the recognition accuracy could be improved if we have this rich geometry location information.

Occlusion could be decided through estimate the visibility of the feature from a certain perspective. The visibility of a feature from a perspective could be calculated through finding the 3D location of the feature and other objects between the feature and the camera. The 3D locations of the feature, camera and object could be calculated by back-projections. This visibility could also be modeled through equation 3.11, and it is more convenient to calculate and deal with uncertainties.

The 3D geometry locations of the features could also help merge features from different images to the same object in the 3D space. Then the object could be classified by the merged features. However the number of the features would be limited to those with correspondences in at least two images. Therefore food recognition through 3D reconstruction will show comparable results with our multi-view classification method. But 3D reconstruction will spend extra time in computing 3D locations of the features.

## 3.4 Implementation

We implement DietCam on an iPhone 4. iPhone 4 has a five-megapixel camera and 1GHz processor. However, we faced a problem when we tried to implement and classification algorithm totally on the iPhone. The problem is that the iPhone 4 is still not powerful enough for complex image feature extraction and feature classification algorithms. We tested it would take about two seconds for iPhone to extract SIFT features from a  $800 \times 600$  image. If we have three images to process, it would take about seven seconds. In addition, it will take longer to classify those features. Therefore, we adopt a client-server structure for DietCam. DietCam is part of a dietary assessment and reporting system, which besides food recognition also includes food volume estimation. In Fig. 3.8, we show the architecture of the whole system of DietCam. The food classifier and volume estimator are implemented on the server side. The smart phone catches the images or videos and send them to the server. The image manager module receives data from the clients and store them in a database. Feature extraction, segmentation and classification tasks are performed by the food recognition module on the server. Then the results are recorded in a summary database and reported back to the clients. It is also possible for the result reporting system to send the results to patient's medical record database, doctors, health surveillance system, and



**Figure 3.7:** The food area is detected in one image shown as a square. The correspondent windows in the other two images are shown as the grids. The actual areas of that food item detected in the two images are represented as the diagonal grids. The probability the food item exists from a camera’s point of view is the ratio of the overlap area to the window area in the image taken by the camera.

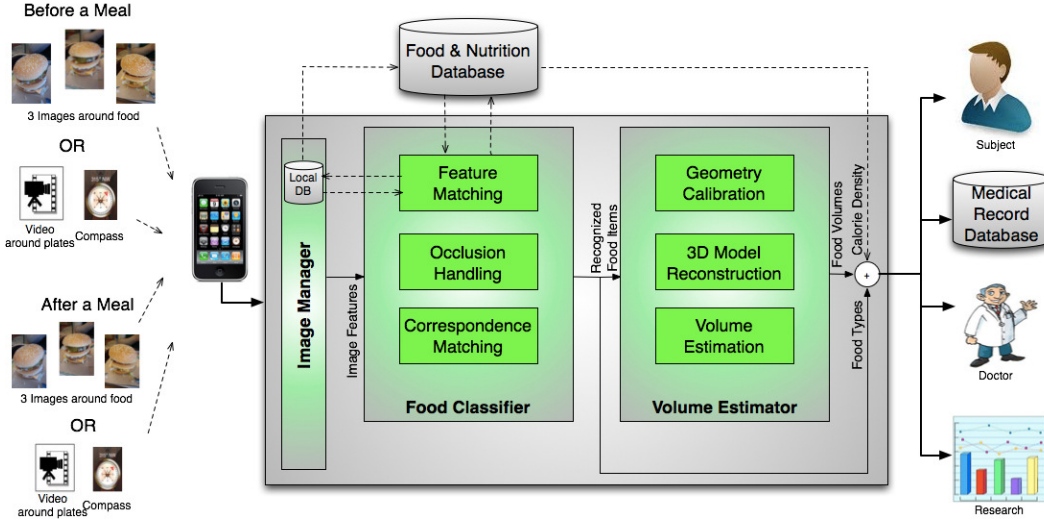
healthcare researchers.

## 3.5 Experiment

DietCam was evaluated with images taken from real restaurant environment and a published food image database. The results were compared with other literature methods, including color histogram, bags of SIFT features, and texture classification. The experiment results were from a prototype of DietCam that had been implemented on the iPhone platform.

### 3.5.1 Dataset

DietCam was evaluated on two data sources, one is the recently released Pittsburgh Food Image Dataset (PFID)[37] and another one is the food images collected from our local restaurants. The PFID dataset is a collection of food images from 13 chain restaurants acquired under lab and realistic conditions. There are 61 categories of specific food items and each of them contains three different instances of the food and six images from six viewpoints of each food instance. The data sources we collected include hundreds of food



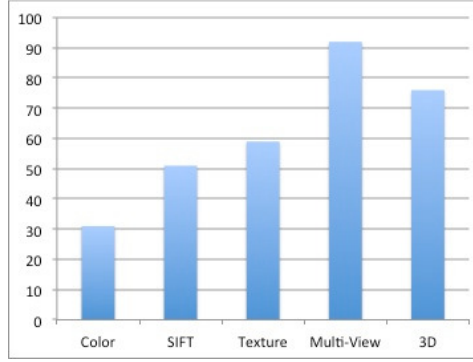
**Figure 3.8:** The client-server architecture of DietCam.

**Table 3.1**  
Food segmentation.

Food number	Fault	Correct	Total	Success rate(%)
1	0	50	50	100
2	2	48	50	96
3	5	45	50	90
4	6	44	50	88
5	12	38	50	76

images of different combinations of pizzas, hamburgers, French fries, chicken strips, sandwiches and drinks. For the diversity and comparison, fruit images were also collected in supermarkets, including apple, apricot, pear, cherry, peach, plum, grape, blueberry, orange, and banana. We also have types of home made foods, including steak, baked potato, and sausage. In total, there are 81 specific food categories and 5110 images. In one image, the number of food items ranges from 1 to 5. In PFID, there is only one food item in an image. In our manually collected images, there are more than one food item in an image. Test samples were collected in real conditions at different restaurants with different combinations of food items and at different times of a day. In the experiment, the food items were placed on the dining table in real dining environment, together with knives, folks, plates etc. Wrapped food items, such as hamburgers and sandwiches, were unwrapped so that the food contents were visible.

The training images consist of PFID images that were taken under the lab environment and 50% of our new images. The realistic images in PFID together with the other new food



**Figure 3.9:** Classification accuracy of a single food item. 135 test cases were selected and 405 images were used. In these images, only one food item appears and there are 81 food categories, including fast food, fruits, and home made foods.

images were test samples. The ground truth of food items in the images was encoded in the file name.

### 3.5.2 Baseline methods

We used three baseline methods that were popular in object recognition literature and recently used for food recognition, including color histogram with a support vector machine (SVM) classifier, texture histogram with a SVM classifier and bags of SIFT with a nearest neighbor classifier. When implementing them for multi-view object recognition, we used a naive bayesian classifier to fuse the classification results from each single view.

We employed a standard RGB 3-dimensional color histogram with four quantization levels per color band. Each pixel in the image was mapped to its closet cell in the histogram to produce a 64 dimensional histogram of the image. Then a multi-class SVM was used for classification. The poor performance of color histogram of multiple food recognition was predictable since it is inadequate for food segmentations. SIFT had been widely used in general object recognition due to its invariance to transformations and illumination conditions. In the experiment, we represented every food image as a histogram of occurrence frequencies of each food type's features. The features were extracted and classified with a nearest neighbor classifier. Since there would be more than one food item in the images, we counted the features of each type of food. But it was still foreseeable that if more than food item of the same category exists in the image, SIFT will not be able to segment them.

Another method used as the baseline was the texture recognition. The texture of an image was found through convolving the image with a texton bank. The image will be represented

as the filter responses. Based on the textures, the image could be segmented and each segmented piece will be classified with a multi-class SVM.

All the baseline methods were single view methods. In order to compare our method with these methods, we implemented a naive bayesian classifier to fuse the classification results from each single view.

### **3.5.3 Segmentation results**

We tested the performance of food segmentation with perspective distance. The test cases were divided into five groups according to the number of food items in the test scene. The number ranged from one to five, and each group had 50 test cases. We took three images for each test scene. The images were taken in the supermarket, restaurants, and home environment. We compared the segmentation accuracy for each group of test cases.

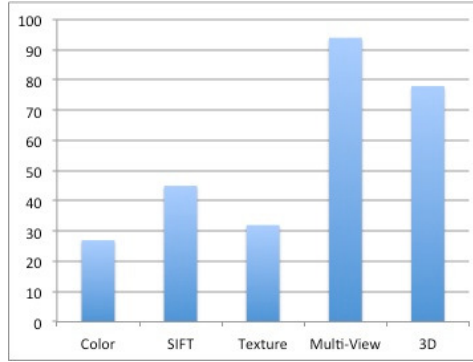
Table 3.1 shows the comparison results. The success rate of food segmentation dropped when the number of food items in the test cases increased. When there was only one food item in the test case, the segmentation algorithm did not negatively segment and food item. When two food items appeared in the scene, the success rate was also good at 96%. When the number increased to three and four, the success rates were acceptable at about 90%. But if there were five items in the scene, the algorithm had only 76% of all the test cases successfully segmented.

### **3.5.4 Classification results**

We implemented DietCam and also the classifier with feature's 3D model reconstruction. The accuracy of these classifiers are compared with those baseline methods.

The accuracy of the food classifier is defined as the fraction of the number of successfully classified images in the total number of test cases. It is affected by many factors, such as fault segmentations, fault classifications to a different shaped food type, misinterpretations between similar shaped food types and food missing in the database.

We first tested the classification results of a single food item. 135 test cases were selected and 405 images were used. In these images, only one food item appears and there are 81 food categories, including fast food, fruits, and home made foods.



**Figure 3.10:** Classification accuracy of fruits, which include apple, apricot, pear, cherry, peach, plum, grape, blueberry, orange, and banana.

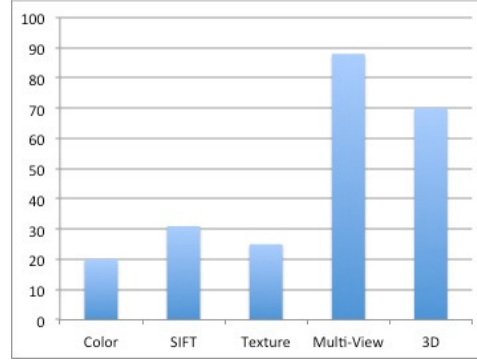
The results are shown in Fig. 3.9. Since the food item does not have to be segmented, the baseline methods exhibit their ability for object recognition. But the accuracies were still too low for field applications. The multi-view method classified 91% food items accurately. 3D reconstruction did not help increase the recognition accuracy. Instead, it reduced the number of features in the reconstruction process, because the reconstructed features came from the common features in all the perspectives.

In order to test DietCam with multiple food items, we chose the test cases with various difficulties. Fruits were tested first, and then fast food was added into the test cases and at last home made foods were added. The number of food items in these test cases is varied between two to four.

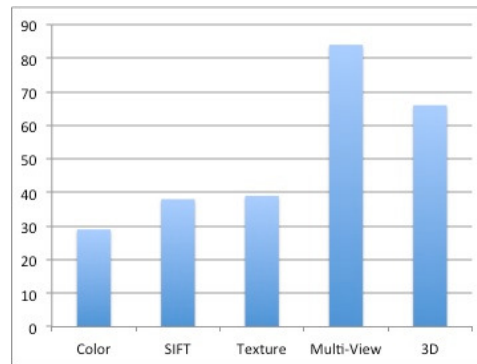
Fig. 3.10 shows the results of fruit classification. Fruit classification is expected to have the highest accuracy compared with fast food and home made food. Fruit has more stable shape patterns, which will give us more consistent features. From the results, it showed feature based classification methods exhibited higher classification accuracies. The multi-view classification method had an accuracy of 94%.

Fig. 3.11 shows the results of fruit and fast food classification. When fast food was added into the test cases, it became more complex. The accuracy of the baseline methods dropped to below 30%. The multi-view method had an accuracy of 88%.

Fig. 3.12 shows the results of fruit, fast food, and home made food classification. It is clear that the three baseline methods have very low accuracies, lower than 40%. This fact is caused by their incapability of segmentation, even we use a naive Bayesian classifier to fuse the results from every view. Our multi-view method has an accuracy of 84%. We examine the wrong results and find that among these test cases, 68% are caused by fault



**Figure 3.11:** Classification accuracy of fruits and fast food.



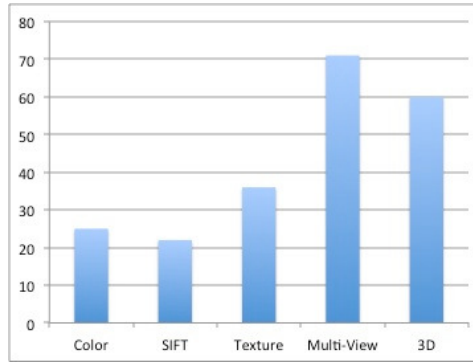
**Figure 3.12:** Classification accuracy of fruit, fast food, and home made food.

segmentations and 32% are caused by fault recognitions.

Fig. 3.13 shows the classification accuracy result of home made foods. The foods include steaks, sausages, and baked potatoes. The accuracy is not as high as those of fast food and fruits. The reason is that the food features we use to describe food items are not distinctive enough to distinguish home made food. In the results, SIFT related methods all exhibited lower accuracies for home made foods. But the color histogram and texture method are not affected by the food categories that much.

## 3.6 Discussion

This paper explores a new research area in biomedicine and computer vision literature, multi-view multi-number food recognition with occlusions. The focus is not to develop



**Figure 3.13:** Classification accuracy of home made foods, which include steaks, sausages, and baked potatoes.

complex image descriptors to describe food features. Instead, the contribution of this paper is a new multi-view object recognition framework that could be applied in food recognition and dietary assessment.

From the experiment, the recognition accuracy is still not high enough even though it is improved significantly compared with other object recognition algorithms. The reason is that the features we use cannot represent each food category completely. The SIFT features rely on statistical properties of food shapes, rather than on other characteristic properties of the food such as colors and textures.

The recognition accuracy could be improved through developing a new food feature that is also capable of describing arbitrary shape food items. The new feature should encode more food properties such as shapes, colors, and textures. In addition, the new feature will analyze food elements such as rice, noodle, and classify the construction of the small elements. This will be our future work.

The experiment is limited to regular shape food classification. However, the multi-view object recognition method presented in this paper is not limited to only regular shape food recognition. It is still applicable for arbitrary food recognition, even for arbitrary object recognition. Image features designed specifically for arbitrary food classification would be presented in the other food classification paper. In the future arbitrary food classification paper, we will test and improve this multi-view classification method with more sophisticated features and more food images.

Currently, we use a client server architecture, and images are processed on the server side. This causes one problem that the client side of the application is still not power efficient. The most power consuming operation is image transmissions. The smart phones are limited



by their computational abilities, so the feature extraction and classification could not be handled on the smart phone. If the smart phone is powerful enough to fast extract features from images, data transmission between the client and server would be reduced to the image features, rather than the whole images.

### **3.7 Conclusion**

This paper presents an automatic camera phone based multi-view food classifier as a novel and convenient approach for food intake assessment, which is critical for obesity management and healthcare. Standard object recognition algorithms and recently proposed food recognition algorithms indicate ill-suitabilities and inferior accuracies. The advantage of this paper is to increase the recognition accuracy to the field application level through a novel multi-view method. A multiple-viewpoint food segmentation and recognition algorithm has been explored and evaluated under a published food database. Compared with standard object recognition algorithms, the results exhibit its accuracy of 84% and 91% when recognizing arbitrary number of or single food item respectively.

# References

- [1] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [2] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [3] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [4] R. Szeliski, “Computer vision: Algorithms and applications,” *Springer*, 2010.
- [5] C. Martin, S. Kaya, and B. Gunturk, “Quantification of food intake using food image analysis,” *Engineering in Medicine and Biology Society. Annual International Conference of the IEEE*, pp. 6869 – 6872, 2009.
- [6] W. Wu and J. Yang, “Fast food recognition from videos of eating for calorie estimation,” *Multimedia and Expo, IEEE international Conference on*, pp. 1210 – 1213, Jan 2009.
- [7] 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,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 4, pp. 756 – 766, 2010.
- [8] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, “Food recognition using statistics of pairwise local features,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2249 – 2256, 2010.
- [9] G. Godin, A. Bélanger-Gravel, A. marie Paradis, M.-C. Vohl, and L. Pêrusse, “A simple method to assess fruit and vegetable intake among obese and non-obese individuals,” *Can J Public Health*, vol. 99, no. 6, pp. 494–8, Jan 2008.
- [10] M. A. Murtaugh, K. ni Ma, T. Greene, D. Redwood, S. Edwards, J. Johnson, L. Tom-Orme, A. P. Lanier, J. A. Henderson, and M. L. Slattery, “Validation of a dietary

- history questionnaire for american indian and alaska native people,” *Ethn Dis*, vol. 20, no. 4, pp. 429–36, Feb 2011.
- [11] N. D. Wright, A. E. Groisman-Perelstein, J. Wylie-Rosett, N. Vernon, P. M. Diamantis, and C. R. Isasi, “A lifestyle assessment and intervention tool for pediatric weight management: the habits questionnaire,” *J Hum Nutr Diet*, vol. 24, no. 1, pp. 96–100, Feb 2011.
  - [12] A. F. Smith, S. D. Baxter, J. W. Hardin, C. H. Guinn, and J. A. Royer, “Relation of children’s dietary reporting accuracy to cognitive ability,” *Am J Epidemiol*, vol. 173, no. 1, pp. 103–9, Jan 2011.
  - [13] L. A. Mainvil, C. C. Horwath, J. E. McKenzie, and R. Lawson, “Validation of brief instruments to measure adult fruit and vegetable consumption,” *Appetite*, vol. 56, no. 1, pp. 111–7, Feb 2011.
  - [14] M. A. Cardoso, L. Y. Tomita, and E. C. Laguna, “Assessing the validity of a food frequency questionnaire among low-income women in são paulo, southeastern brazil,” *Cad Saude Publica*, vol. 26, no. 11, pp. 2059–67, Nov 2010.
  - [15] F. H. Esfahani, G. Asghari, P. Mirmiran, and F. Azizi, “Reproducibility and relative validity of food group intake in a food frequency questionnaire developed for the tehran lipid and glucose study,” *J Epidemiol*, vol. 20, no. 2, pp. 150–8, Jan 2010.
  - [16] A. E. Dutman, A. Stafleu, A. Kruizinga, H. A. Brants, K. R. Westerterp, C. Kistemaker, W. J. Meuling, and R. A. Goldbohm, “Validation of an FFQ and options for data processing using the doubly labelled water method in children,” *Public Health Nutr*, pp. 1–8, Aug 2010.
  - [17] M. Aubertin-Leheudre, A. Koskela, A. Samaletdin, and H. Adlercreutz, “Plasma alkylresorcinol metabolites as potential biomarkers of whole-grain wheat and rye cereal fibre intakes in women,” *Br J Nutr*, vol. 103, no. 3, pp. 339–43, Feb 2010.
  - [18] G. L. Bowman, J. Shannon, E. Ho, M. G. Traber, B. Frei, B. S. Oken, J. A. Kaye, and J. F. Quinn, “Reliability and validity of food frequency questionnaire and nutrient biomarkers in elders with and without mild cognitive impairment,” *Alzheimer Dis Assoc Disord*, vol. 25, no. 1, pp. 49–57, Jan 2011.
  - [19] P. B. Ryan, K. A. Scanlon, and D. L. MacIntosh, “Analysis of dietary intake of selected metals in the nhexas-maryland investigation,” *Environ Health Perspect*, vol. 109, no. 2, pp. 121–8, Feb 2001.
  - [20] M. R. Ritchie, M. S. Morton, N. Deighton, A. Blake, and J. H. Cummings, “Plasma and urinary phyto-oestrogens as biomarkers of intake: validation by duplicate diet analysis,” *Br J Nutr*, vol. 91, no. 3, pp. 447–57, Mar 2004.

- [21] P. Viola and M. Jones, “Robust real-time face detection,” *Computer Vision, IEEE International Conference on*, vol. 2, pp. 747 – 747, Jul 2001.
- [22] M. Weber, M. Welling, and P. Perona, “Unsupervised learning of models for recognition,” *Computer Vision, European Conference on*, pp. 18 – 32, Jan 2000.
- [23] H. Schneiderman and T. Kanade, “A statistical method for 3d object detection applied to faces and cars,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 746 – 751, Jun 2000.
- [24] D. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern recognition*, vol. 13, no. 2, pp. 111–122, Jan 1981.
- [25] D. Lowe, “Object recognition from local scale-invariant features,” *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1150 – 1157, 1999.
- [26] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Apr 2005.
- [27] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Jan 2004.
- [28] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, Jan 2004.
- [29] T. Kadir and M. Brady, “Saliency, scale and image description,” *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, Jan 2001.
- [30] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, pp. 761–767, Jan 2004.
- [31] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 506–513, 2004.
- [32] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua, “Optimizing kd-trees for scalable visual descriptor indexing,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3392 – 3399, 2010.
- [33] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, “Context-based vision system for place and object recognition,” *Computer Vision. IEEE International Conference on*, vol. 1, pp. 273–280, 2003.

- [34] A. Collet and S. Srinivasa, “Efficient multi-view object recognition and full pose estimation,” *Robotics and Automation, IEEE International Conference on*, pp. 2050–2055, 2010.
- [35] N. Naikal, A. Yang, and S. Sastry, “Towards an efficient distributed object recognition system in wireless smart camera networks,” *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1 – 8, 2010.
- [36] D. Williams, “Bayesian data fusion of multiview synthetic aperture sonar imagery for seabed classification,” *Image Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1239 – 1254, 2009.
- [37] M. Chen, K. Dhingra, W. Wu, and L. Yang, “PFID: Pittsburgh fast-food image dataset,” *Image Processing, IEEE International Conference on*, pp. 289 – 292, Jan 2009.
- [38] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2161–2168, 2006.
- [39] G. Strang, “Introduction to linear algebra, 3rd ed,” *Wellesley-Cambridge Press*, 1998, 1998.
- [40] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 8, no. 6, pp. 679 –698, 1986.

## **Chapter 4**

# **DietCam: Multi-view, Multi-class Food Recognition Using a Multi-kernel Based SVM**

### **Abstract**

Food recognition is an extremely difficult problem but it is helpful for human to evaluate everyday food intake. In this paper, we present DietCam, an automatic food classification method, which overcome challenges caused by the uncertainties of food appearances. DietCam divides the food classification problem into two steps, ingredient detection and food classification. Food ingredients are detected through a combination of deformable part model and texture verification model. With detected ingredients, DietCam classifies the food item from multiple viewpoints, multiple scales, and with multiple classification methods. In the experiment with 55 food types and 15262 food images, DietCam achieves promising results compared with existing image based food recognition methods. The accuracy is increased by over 60% for complex ingredient composition food items.

### **4.1 Introduction**

Food recognition from images is such a difficult problem in computer vision and pattern recognition. It is a complex problem that is still far from being solved. The best results achieved so far is 84% for fast food [1] and 70% for general food [2]. In addition, it



**Figure 4.1:** Food appearances. It is a difficult problem to classify food from its appearances. There are a large variety of food types and even the same food type could have different appearances. In the figure, there are steaks, salads, fried rices, pastas, and sandwiches. But even the same type of food looks different.

still does not draw sufficient attentions in the literature. But food recognition is worth attentions and efforts. One reason is that automatic food recognition will be beneficial to healthcare related applications, such as obesity management. Another reason is that food recognition exposes new challenges to the current pattern recognition literature and stimulates the stemming of novel techniques for generalized object recognition.

Visual food recognition could help people assess their food intake. Currently, obesity has been a severe public health challenge to the general population and social welfare in many developed countries[3, 4]. In the past three decades, the obesity rate in U.S. has increased significantly[5], resulting in serious consequences such as diabetes, stroke, heart disease even cancers. Food intake assessment is important for obesity management. However, few people are aware of their food intakes and they are not willing to assess food intakes. The reason is the burdensome assessment methods and a lack of real-time feedback with these methods. Traditional food record or food diary methods require manual records of the food type and the portion of the food taken, and the accuracy is limited by human estimations of the food portion. Commercialized iPhone application Meal Snap recently appeared, and helped people record and recognize food images. But it needs a large number of human labors to recognize the images manually. Computer aided and vision based automatic food intake assessment methods do not suffer from manual records or inaccurate human estimations.

Even though vision based object recognition has been explored for decades, and good results have been achieved in recognizing some objects such as face [6], and cars[7], food recognition imports new challenges into the current pattern recognition researches. Food recognition could be categorized into the category recognition sub-class of object recognition. Category recognition involves recognizing objects belonging to extremely varied categories, such as animals, furnitures, flowers, etc. It is still an unsolved problem, even

at a level of two-year old child [8]. Food recognition is one of the most difficult problems in category recognition. The appearance of food has a higher degree of uncertainties than categories such as animals and flowers. Fig. 4.1 shows the uncertain appearances of five kinds of food. In each column, there are steaks, salads, fried rices, pastas, and sandwiches. However, even the appearances of the same kind of food look different. If food recognition problem could be solved, other category recognition problem could also benefit.

Generalized object recognition methods have been appeared in research papers for food recognition. These techniques include color histogram [9], texture [10] and bag-of-feature classifications [11]. However, none of them achieved acceptable results in food recognition. Color histogram classification only considers the statistical properties of colors. But colors would be affected by lighting conditions, and at most of the time, food could not be differentiated only through color. The same reason exists for texture classification. Not every food category has the unique texture pattern. Bag-of-feature method considers the statistics of key image patches. The patch detector tries to find specific locations that look differently from other locations. But the problem is that the specific locations are not necessarily the key characters of food items.

Novel techniques are required for food recognition. Most of the new pattern recognition methods are stemmed from human cognition. Given a complex food composition, human recognizes its ingredients first from the ingredient shape, texture and color. Then, from the combination of the ingredients, human could know some possible candidates for what they haven seen. Appearance based food recognition could be difficult even for human beings. People sometimes do not have sufficient knowledge to distinguish foods only through their appearances. But computer has advantages over human beings in remembering the knowledge.

Even though humans could recognize food from ingredients, there are challenges for computers to finish the same task. On one hand, different food types could have similar ingredient appearances. On the other hand, the same food type could have ingredients of different appearances. In addition, the appearances of food ingredients are affected by recipes, cooking methods, and chef's personal preferences. Another challenge is that even though ingredients could be detected correctly, food types could also have unstructured ingredient constructions. For some kinds of food, the ingredients are distributed randomly. Food recognition from different scaled images is another challenge. Some types of food could not be differentiated through their own sizes in images. For example brown rice looks similar to a baked potato if their relative scale is not considered. But we also have to consider the scale for the same type of food. Another important challenge is the occlusion in images. Food is always placed in certain containers so that some key elements could be covered or occluded by other ingredients.



This paper presents a new ingredient based food recognition method, which solves all these challenges. Our first innovation involves enriching the current part based object recognition model using relative scales, textures verifications, and flexible location models to detect food ingredients. The state of the art part detectors are not suitable for food ingredient detection concerning the above challenges. We modify the detector in three ways so that it is able to detect food ingredients. The ingredient detector tries to find food ingredients on a single scale in order to retain the ingredient relative scales. After that, the scale invariance is achieved in a multi-scale support vector machine (SVM) used in the food classifier. The existing part model uses the shape of the objects as the key property. We enrich the model with the ability to verify detection results with texture models, which means both the shape model and texture model are used to detect food ingredients. Color is not considered since lighting conditions in different restaurants varies significantly. In the existing part based model, the geometry locations of the part are modeled strictly. In our model, we employ a more flexible location model for food ingredients, so that the degree of the location flexibility could be controlled.

Our second contribution includes developing a new multi-view, multi-kernel formulation for SVM to classify various food ingredient combinations and handle occlusions. We design a SVM with multiple kernel learning method. The kernels include a hierarchy of element kernels. The top level is the viewpoint level. Concerning the occlusion challenge, we adopt a multi-view scheme to get rid of occlusions. On the viewpoint level, each view corresponds to a kernel function, and all the kernel functions from multi-view are combined together according to the images similarities between viewpoints. Under each viewpoint kernel, we design spatial pyramid kernels to achieve scale invariance. Then under each scale, there are a linear combination of linear, quasi-linear, and non-linear element kernels to classify food ingredient features. By employing such a hierarchy of kernel functions, we accomplish a classifier which could combine multiple viewpoints, detect and classify objects of different scales. In the experiments, we show that this classifier achieves guaranteed results.

The rest of this paper is organized as follows: section 4.2 enumerates related researches about food intake assessment and object category recognition techniques, and then we present the details of ingredient detection in section 4.3, food classifier and multi-view technique in section 4.4. After that, the dataset we used in training and experiment is introduced in section 4.5. Section 4.6 evaluates the technique with field experiments and the results are discussed in section 4.7. Finally we conclude the paper in section 4.8.

## 4.2 Related Work

This paper presents an ingredient based food recognition method for automatic food intake assessment. It is an interdisciplinary study that is related to biomedical food intake assessment research and visual object category recognitions.

Food intake assessment has been a popular research topic in biomedical and health related areas for years. People recognize its importance and try to find methods to evaluate dietary intakes accurately. A simple and widely used method is food diaries and records [12–14]. People have to write down the food types and estimate the volumes of each type of food. It is applicable to most people since it does not require any professional knowledge. However, it is limited by the roughness of human estimations[15]. Similar methods that suffer from the same drawbacks also include dietary histories [16], and food frequency questionnaires[17, 18].

Concerning about the inaccuracy caused by human estimations, researchers develop methods to monitor food intakes inside human organs. Typical methods include biological assessment and chemical analysis. Biological assessments, e.g. doubly-labelled water [19], plasma carotene [20], etc., monitor food intake through the introduction of biomarkers[21]. The chemical analysis methods evaluate the dietary intake through tracking selected elements [22]. Both the biological and chemical methods report the validation and accuracy in food intake assessment [23]. However, these methods could only be used in the lab environment and are not available to everyone.

In computer vision, food recognition is a specific case of category recognition. Currently, it still has not drawn significant attentions and has no specific solutions. Some researchers tries to recognize foods from images with existing methods of instance recognition and general category recognition. But the accuracy is still not high enough for applications. For example, Martin et al. use a general color histogram of the image to recognize food items[9]. Wu et al. present interest points (SIFT) based method to recognize fast food[11]. But the accuracy is under 70%. Zhu et al. detect food textures from the image first and utilize the histogram of textures to classify food items[10].

Ingredient based food classification methods have appeared recently [2]. The ingredients are detected through classifying food textures. Then the food is classified through calculating the pairwise statistics between food ingredients. But, the accuracy is still only 78%. In this paper, we use a combined model of texture models and state of the art part based model to extract food ingredients. Then a multi-view, multi-model SVM is used to classify the food ingredients.

Methods for texture representation can be mainly categorized into two classes, texture in spatial domain and texture in frequency domain. Texture in spatial domain usually relies on local or global texture descriptors that are invariant to geometric or illumination changes [24, 25]. The popular methodology is to extract local patches first, then quantize these patches into a texon dictionary. Texture on frequency domain method usually is done in wavelet domain [26–28]. Overall, the performance in frequency domain is not as good as that in spatial domain [28]. Texture information has been an important part of visual vocabulary for flower classification [29]. In these texture classification methods, only Semantic Texon Forests [30] has been used for food ingredient classification [2].

Part based recognition is an extension of template based recognition method. Template based recognition involves re-recognizing a known 2D or 3D rigid shape object, but potentially from a different view point, and/or under different lighting conditions, and/or with different backgrounds, and/or with partial occlusions, and/or from different distances. Rigid shape object recognition usually uses template-based methods[7, 31, 32], which exhibit good performance for single object recognition, e.g. cars[7, 32]. An important limitation of these methods is their inflexibility to capture the variances of object appearances. Part based recognition introduces a geometric distribution model of different parts to represent the variance of object appearances [33]. Based on part based recognition, recently visual phrase recognition appears to recognize complex visual composites, such as “a person riding a horse” [34], which motivates our food ingredient composition. However, food ingredient composition is more flexible and more complicated.

SVM is one of the most successful techniques in classification problems. One example is face classification [35]. The kernel function of SVM plays a critical role in discriminant analysis and dimensionality overcoming [36]. A common method to choose kernel function is to design a function according to the classification problem. Histogram intersection kernel has been studied in [37].  $\chi^2$  kernel has been used in [38]. The scale invariance is achieved through spatial pyramid kernels [39–41]. In order to select kernels optimally and automatically, multiple kernels learning has been studied in [42–44]. In this paper, we develop a multi-kernel method based on multiple kernels learning method.

### 4.3 Ingredient detection

Attributes that differentiate foods are ingredients. Food could be prepared with different cooking methods, and different condiments. However, the dominating ingredients are the same for the same kind of food. Therefore, if we could find the key ingredients, the food could be classified according to the combination of these ingredients. The food classification process is divided into two functions, ingredient detection, and ingredient combination

classification.

It is not easy to find food elements only through their colors, shapes, or textures. The color attributes of an object are always affected by lighting conditions. The shape of a food element is decided by many factors, such as its natural shape, cutting patterns and perspectives. Different types of food could also appear to have the same type of texture. Therefore, typical color histogram classifier, shape template based method and texture classification methods are not suitable for food ingredient detection.

A food ingredient detector combining part models and texture models is developed in this paper. Part based models are popular for rigid shape object detection and classifications. It concerns not only deformable part detections but also the geometric relations between parts. However, it is not suitable to detect food ingredients. The reason is that different ingredients could have the same shape appearance. We integrate texture filters into part based detection, where after the parts are detected, the texture of the parts will be verified that it has the possible food texture.

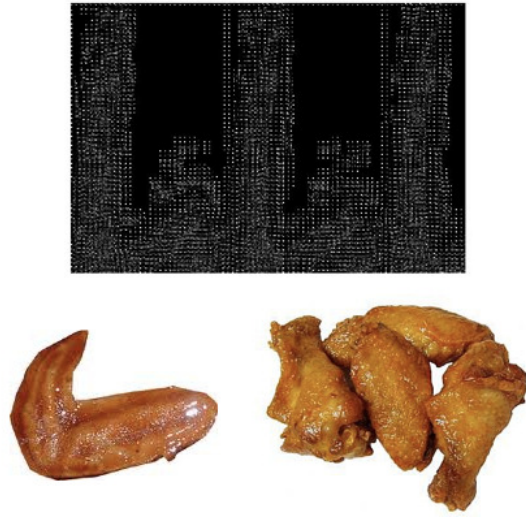
The part model is based on the state of the art part detectors in [33] using deformable models. It learns models at different resolutions, and builds geometrical location models of the parts. When detecting objects, it builds feature maps of the image at different resolutions and different scales. The feature maps will be filtered with the learnt root model, part model and location model. Then, the filter responses will be combined together to find the root locations.

The existing part detector could be used to detect food items with certain structures, such as chicken wings and sandwiches. Fig. 4.2 shows a trained histogram of oriented gradients descriptor for chicken wing model. In this figure, every whole chicken wing and unconnected segment of the chicken wing is a food ingredient. Therefore, the part model is a synthesis of five chicken wing ingredients together.

However, part model is not suitable directly for foods with the following attributes: 1) food with similar shape but different scales, such as rice, meat ball, and baked potato, etc. 2) food with similar shape but only differentiable through textures, such as beef steak, fish, and pork steak, etc. 3) food with more flexible geometry distributions of ingredients, such as pizza toppings, rices, noodles, etc.

In this paper, we made three modifications to the part detector for food elements detection.

**Modification 1: detecting in the same scale.** The existing detector tries to find objects at different scales in order to achieve the scale invariance. However, it loses the relative scale between different objects. For example, a part model of a baked potato could also detect

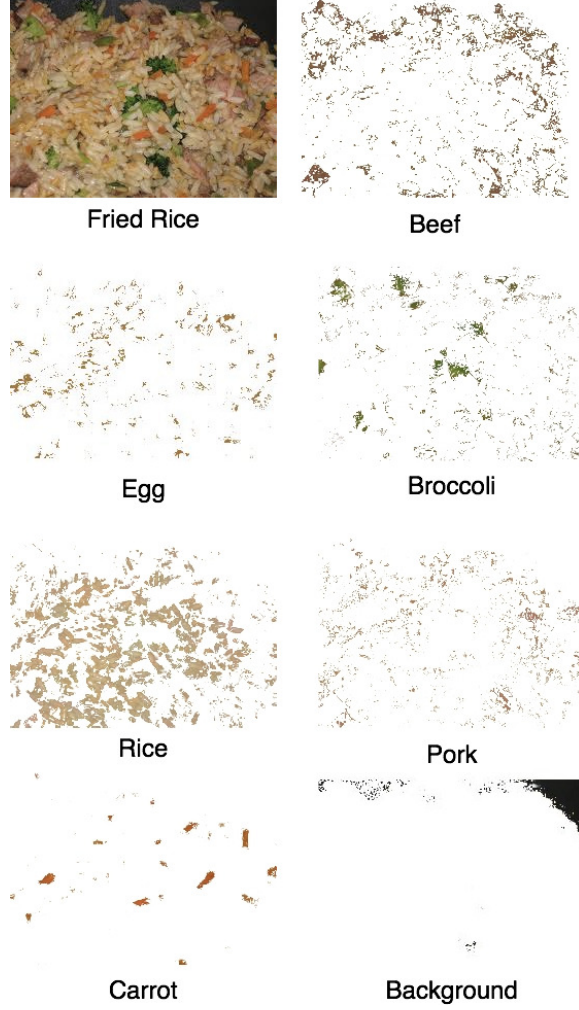


**Figure 4.2:** Histogram of oriented gradients descriptor of chicken wings. Every unconnected segment of the chicken wing is a part of the ingredient model. Therefore, in this figure, there are five chicken wing ingredients and the model synthesizes all of them.

a piece of rice. The technique is modified so that at the ingredient detection phrase, only one scale of the image is used to retain the relative scales of different ingredients. But the scale invariance property of the food classifier is investigated during further classification phrase. In this way, we can define relative small elements and large elements.

**Modification 2: using a mixed model of part and texture.** The complex visual composites of a food item is sometimes not differentiable only through shape attributes. Textures and colors are also important properties. But the color of a food ingredient is not reliable due to variant lighting conditions. A texture filter bank, Semantic Texton Forest (STF) in [30] is chosen to detect food textures. STF is an image segmentation and classification technique that generates soft labels for each pixel based on their local texture properties. This is achieved through learning from manually labeled sample images and building decision forests. Fig. 4.3 shows the results. After the part model detects ingredients, the texture model will verify the results.

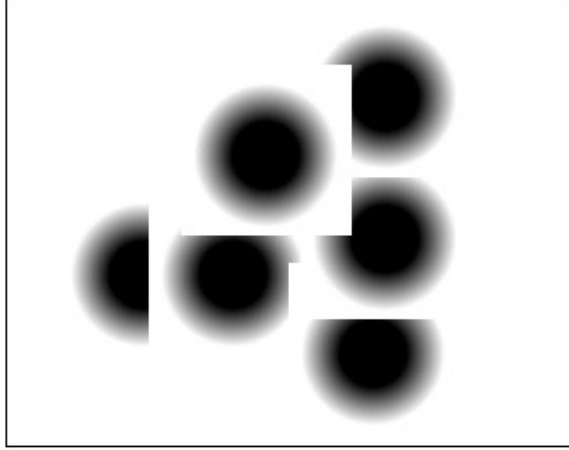
**Modification 3: using more flexible location model.** Foods are extremely deformable objects, and the ingredients do not have a certain pattern of geometric distributions. Therefore, a location model with higher degree of flexibility are more desirable. Fig. 4.4 of the position of the position filters. The degree of flexibility is learned from the sample images.



**Figure 4.3:** Extracted food ingredient textures of a plate of fried rice.

Some ingredients may have a ‘curvature bulb’ geometry distributions. For some certain food elements, their geometry distribution may be uniform inside the boundary of the food, such as pizza toppings.

With these modifications, the results of the ingredient detection is a histogram of the food ingredients appeared in the image. There are  $d$  food types, and we use a vector  $\mathbf{z}$  to represent the histogram,  $\mathbf{z} = (z_1, z_2, \dots, z_d) \in \mathbb{R}^d$ . The next step is to classify  $\mathbf{z} \in \mathbb{R}^d$ .



**Figure 4.4:** The part location model of chicken wings. The black dots represent the possible part locations. A part will have a higher probability to appear at a place with darker dots. The combination of the dots represents the location relationship of different parts.

## 4.4 Food classification

After detecting the ingredients, we combine the ingredients together as a ingredient histogram  $\mathbf{z}$ . Since some types of food contain only one ingredient, and a meal could include more than one food item, we use a visual phrase [34] to represent the classification results, such as “a chicken sandwich with a drink” or “a beef steak with rice”.

Visual phrases encode food structures and the relation between complex visual composites, which are the food ingredients. Given a list of the food ingredients, the visual phrase encodes which ingredients are together the composites of the same type of food. Therefore the visual phrases could be classified from the ingredient histograms.

In order to classify the ingredient histogram, we use a SVM. SVM is one of the most successful techniques in classification problems. It could find a unique global optimal solution and it has a solid mathematical derivations. SVM tries to solve the following classification problem.

Given  $n$  samples  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , with labels  $y_i = \pm 1$ , for every sample  $\mathbf{x}_i, (i = 1, \dots, n)$  is a vector in a  $d - dimensional$  space

$$\mathbf{x}_i = (x_{i_1}, x_{i_2}, \dots, x_{i_d}) \in \mathbb{R}^d \quad (4.1)$$

Assume the samples are linearly separable in the feature space. The best hyperplane (the classifier) to separate the samples is defined in the form as

$$\mathbf{w}^T \mathbf{x} = b \quad (4.2)$$

The geometric margin of the resulting classifier is  $\frac{1}{\|\mathbf{w}\|}$

The SVM tries to maximize the margin in order to obtain the best classifier. Then we have such an optimization problem,

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \frac{1}{\|\mathbf{w}\|} \\ \text{s. t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i - b) \geq 1 \quad i = 1, \dots, n. \end{aligned}$$

It is equivalent to maximizing the quadratic term  $\frac{2}{\|\mathbf{w}\|^2}$  for calculating efficiency. However, typically the training samples are not linearly separable. Then a soft margin is added to the margins.

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i^2 \\ \text{s. t.} \quad & y_i(\mathbf{w}^T \mathbf{x}_i - b) \geq 1 - \xi_i \quad i = 1, \dots, n. \end{aligned}$$

Then it is converted to a convex optimization problem. We define the Lagrangian function as

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi) = \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ & + C \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i - b) - 1 + \xi_i) \end{aligned}$$

where  $\alpha$  is a vector of length  $n$ ,  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  and  $\alpha_i \geq 0$  are the Lagrange multipliers. The optimal is achieved when  $\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \mathbf{w}} = \mathbf{0}$ ,  $\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \alpha} = 0$ , and  $\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \xi} = \mathbf{0}$ .



$$\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i = \mathbf{0} \quad (4.3)$$

$$\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \alpha} = \sum_{i=1}^n y_i \alpha_i = 0 \quad (4.4)$$

$$\frac{\partial L(\mathbf{w}, \alpha, \xi, b)}{\partial \xi} = C\xi - \alpha = 0 \quad (4.5)$$

and substituting the relations, we obtain

$$\begin{aligned} \max_{\mathbf{w}, b, \alpha} \quad & L(\mathbf{w}, \alpha, b) = \sum_{i=1}^n \alpha_i \\ & - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (K'(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{i,j}) \\ \text{s. t.} \quad & \sum_{i=1}^n y_i \alpha_i = 0, \alpha_i \geq 0 \end{aligned}$$

where  $\delta_{i,j}$  is defined to be 1 if  $i = j$  and 0, and  $K'$  is the kernel function.

We define a new kernel function  $K$ ,

$$K(\mathbf{x}, \mathbf{z}) = K'(\mathbf{x}, \mathbf{z}) + \frac{1}{C} \delta_{\mathbf{x}}(\mathbf{z}) \quad (4.6)$$

where  $\mathbf{z}$  is the test sample to classify. Then the classification function will be in the form as

$$f(\mathbf{z}, \alpha^*, b^*) = \sum_{i=1}^n y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{z}) + b^* \quad (4.7)$$

where  $\mathbf{x}$  are the training samples,  $\alpha^*$  and  $b^*$  are the optimal  $\alpha$  and  $b$  learned from the training samples, and  $\mathbf{z}$  is the new sample to classify.

In the SVM, the kernel function  $K(\mathbf{x}_i, \mathbf{z})$  plays an important role in measuring the distance between two features. The most common linear kernel function is the inner product. However, not all the classification problem could be solved through a linear inner product kernel function. Non-linear and quasi-linear kernel functions have also been explored for different problems and promising results have been achieved.

Choosing a right kernel function for a specific problem is still a problem when using SVM for classification problems. The kernel function could be viewed as discriminative properties of a feature. The discriminative properties of a visual feature are encoded in several feature channels. The trade-off between the discriminative power and invariance distinguishes one feature from another. This trade-off varies from task to task. Therefore, no single kernel function can be optimal for all situations.

Motivated by recently appeared multiple kernel learning method [43], in this paper, an optimal combination of kernel functions are learnt, each of which captures a different feature channel. Our features include the contribution from different viewpoints, distribution of food ingredients, and these features at different spatial pyramid levels.

All these features are organized in a hierarchy of kernel functions. On the top level is a linear combination of multiple viewpoints. The weight of each viewpoint is calculated from the relations between views. Under each viewpoints, there is another level of kernel functions, which are a linear combination of element kernel functions at multiple scales.

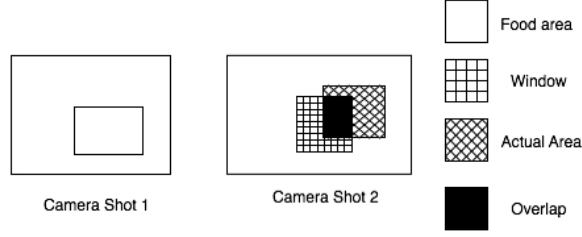
#### 4.4.1 Multiple viewpoints

Classifying food items from a single viewpoint with the feature points would be inaccurate, since occlusions could block key food ingredients. In order to get rid of the occlusions, we develop a multi-view kernel for food classification task, which considers food appearances from more than one perspective.

Given  $m$  viewpoints,  $V = \{v_1, v_2, \dots, v_m\}$ , kernel function for all the viewpoint is defined as,

$$K^{multi-view}(\mathbf{x}, \mathbf{z}) = \sum_{\ell=1}^m g_{\ell} K^{v_{\ell}}(\mathbf{x}, \mathbf{z}) \quad (4.8)$$

The weight of each viewpoint  $g_{\ell}$  is calculated through the relations between each views.



**Figure 4.5:** Geometric similarity between two viewpoints. The rectangle area in camera shot 1 is the food area. The corresponding window area of the food area in camera shot 2 is the grid window. At the same time in camera shot 2, we detect the food area is the grid actual area, which has an overlap with the corresponding window. The fraction of the overlap area in the corresponding window area represents the geometric similarity.

It is not learnt from the training sets, nor an average of all the viewpoints. Consider two views, the results of the second view is related to the results in the first image and the kernel is used to model the occlusions and fault recognitions. Intuitively, if a food is occluded in the first image, it shows a small probability that food exists in the scene. But in the other image, where the food is not or partial occluded, the probability that it exists is still high. On the other hand, if the food does not exist, but it is detected in the first image, which is a fault recognition, in the other images it still has a small probability to be detected. In this way, the recognition results will be enhanced.

Let's define  $g_\ell$  formally. If start from  $v_i$ , for  $v_j \in V$ ,  $g_j$  is defined by the geometrical similarity  $\tau(i, j)$  between viewpoints  $i$  and  $j$ .

$$g_j = \tau(i, j) = \frac{Area(j)}{Window(i, j)}. \quad (4.9)$$

Figure 4.5 defines  $\tau$ .  $Window(i, j)$  denotes the area of the projection of food item in image  $i$  if the food is not occluded. It is calculated with food ingredients detected in the first image. A set of correspondent bounding boxes could be found in the other image with the geometric relations. These bounding boxes compose a correspondent window in the images.  $Window(i, j)$  represents the area of the window.  $Area(j)$  represents the actual area of item appeared in image  $j$ . It is calculated from the actual food ingredients detected in the image.

#### 4.4.2 Multi-kernel

Under one viewpoint, multiple kernels are used to classify food from different feature channels. We use the technique in [42] to learn the optimal kernel function. If we have  $k$  element kernel functions, the optimal kernel is defined

$$K^{v_\ell}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^k d_i K_i^{v_\ell}(\mathbf{x}_i, \mathbf{z}) \quad (4.10)$$

where  $d_i$  is the optimal weight of the  $i$ th kernel.

Then the classifier is in the form of

$$f(\mathbf{z}, \boldsymbol{\alpha}^*, b^*) = \sum_{i=1}^n \sum_{j=1}^k \sum_{\ell=1}^m y_i \alpha_i^* d_j K_j^{v_\ell}(\mathbf{x}_i, \mathbf{z}) \quad (4.11)$$

The kernel weights  $d$  is learnt through optimization, which is carried out in a SVM so as to achieve the best classification results on the training set [42].

Since food recognition has many uncertainties, and it is not sure what kind of kernel is the best, we choose different kinds of element kernels. We consider most of the popular kernels used in object recognition, including linear kernel of the form

$$K_{linear}(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle \quad (4.12)$$

and quasi-linear kernels in the form of

$$K_{quasi-linear}(\mathbf{x}, \mathbf{z}) = \frac{1}{2}(1 - \chi^2(\mathbf{x}, \mathbf{z})) \quad (4.13)$$

and non-linear, RBF- $\chi^2$  kernel of the form,

$$K_{non-linear}(\mathbf{x}, \mathbf{z}) = e^{-\gamma \chi^2(\mathbf{x}, \mathbf{z})} \quad (4.14)$$

In addition, the food ingredients are detected in three pyramid levels, corresponding to one, four, and sixteen spatial subdivisions.

## 4.5 Dataset

Image datasets are a prerequisite to visual object recognition. Researchers have developed several image databases for general and specific purpose datasets for object recognition. However, there are few complete food image databases available. PFID [45] is one published image dataset we found for research purpose. However, it only has 4545 images for fast foods.

We develop a food image database consisting of 15262 images for both training and testing purposes. Training of the SVM consists of two parts, training of the food ingredient detector and training of the food classifier.

We collect a list of 55 popular american food categories, ranging from drinks, pies, to sandwiches and Hoppin’ Johns. We assign a difficulty index to each food category according to their ingredient composites and number of ingredients. The regular shape food and rigid shape food have the smallest difficulty indices, while those like Hoppin’ John, fried rice and noodles have the highest. Table 4.1 shows a complete list of the food classification difficulties.

For each food category, we use Google Image search engine to gather images. Downloaded images are manually validated. We rotate the images to create multi-view images. We also collect multi-view images from restaurants and home as the test images. The ground truth about an image, i.e. food category, positions in the image, food ingredient positions, and ingredient categories are manually labeled and stored in an xml file.

The database has been split into 50% for training and validation, and 50% for testing. The distribution of images, food objects, and food ingredients are approximately equal across the data sets. In total there are 55 food categories and 15262 food images in the database. On average each class has 277 images, with 466 examples. Table 4.2 shows the details of the database statistics.

**Table 4.1**  
Food Difficulty Statistics.

Difficulty	Ingredient Composition	Ingredient	Examples
1	Single dominant ingredient	1	lobster, steak, snow cone, drink, chicken wing
2	One same ingredient repeats a few times	1-5	pancake, corn bread, krispy kreme
3	One same ingredient repeats many times	1-n	shrimp, biscuits, French fry, musubi, cookie, fried onion
4	More than one dominant ingredient	2-n	steak dish, fish dish
5	Dominant ingredients with many small ingredients	n	pie, burger, taco, sandwich, pizza, sushi
6	Many small ingredients repeat many times	n	fried rice, baked beans, noodles, boils, fajitas, Hoppin' John

## 4.6 Experiment

We evaluate DietCam in two parts, ingredient detection and food classification. Results of these two steps are compared with baselines methods respectively. We found three methods used in food recognition and they are also popular pattern classification methods that have been widely used in object classification. The baseline methods include SIFT with a nearest neighbor classifier, texture classification and color histogram with a SVM classifier.

### 4.6.1 Ingredient detection

The goal of ingredient detection is to predict bounding boxes of each food ingredient in the image. DietCam will output a set of ingredient bounding boxes with scores. We threshold the scores at different points to finalize the bounding boxes. We use precision-recall curves across all the images in the dataset to evaluate the performance of DietCam ingredient detection.

The ground truth bounding boxes of an image are stored in an xml file. The ground truth is

parsed when detected bounding boxes are extracted from the image. A predicted bounding box is considered correct if it overlaps more than 50% with a ground truth bounding box. Otherwise, it is considered a false detection. According to this rule, a precision-recall curve could be plotted across a test set. We categorize the dataset into six groups, according to the difficulties. Every group has a precision-recall curve, as shown in Fig. 4.6 to Fig. 4.11.

We implemented the baseline methods for result comparison. SIFT had been widely used in general object recognition due to its invariance to transformations and illumination conditions. In the experiment, we represented every food ingredient as a histogram of occurrence frequencies of each ingredient type's features. The features were extracted and classified with a nearest neighbor classifier.

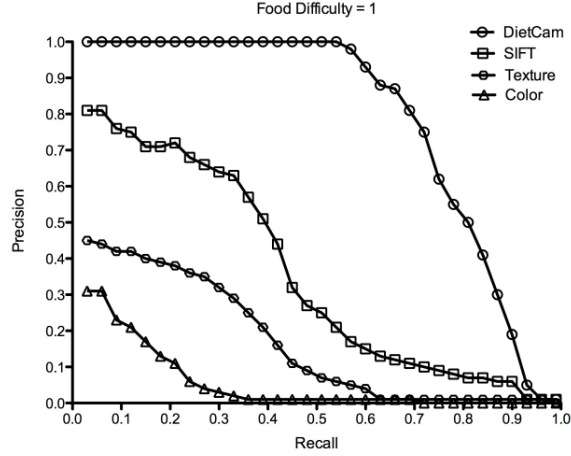
Another method used as the baseline was the texture recognition. We use the same texture model in DietCam for comparing. The difference is that DietCam uses a mixed model of both texture and part models. The texture of an image could be found through convolving the image with a texon bank. Then the texture is classified through decision forests. The image will be represented as the filter responses. Based on the textures, the image could be segmented and each segmented piece will be classified with a multi-class SVM.

For the color histogram baseline method, we employ a standard RGB 3-dimensional color histogram with four quantization levels per color band. Each pixel in the image was mapped to its closet cell in the histogram to produce a 64 dimensional histogram of the image. Then a multi-class SVM was used for classification. The poor performance of color histogram of multiple food recognition was predictable since it is inadequate for food segmentations and the complex lighting conditions make it inferior.

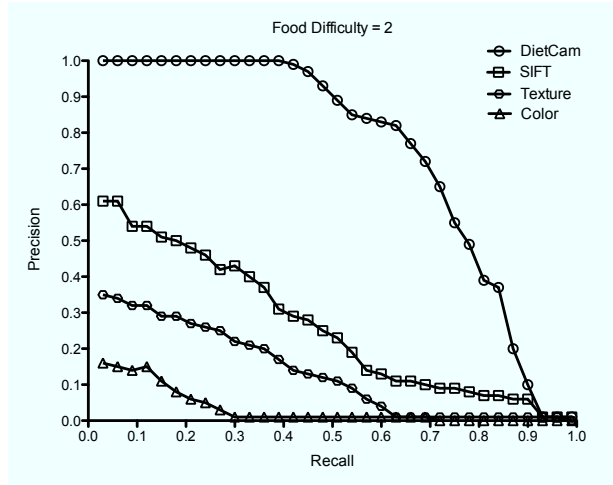
Fig. 4.6 shows the result precision-recall curves of DietCam and baseline methods for difficulty 1 food ingredient detection. In this class of food items, one item consists of one dominant ingredient. But an image could contain more than one food item. Since there is only one ingredient for one food type, it is expected to have the best result among all the difficulties. DietCam outperforms all the baseline methods in precision. The precision of the baselines methods are affected by the number of food items in the images. While there are few images that has only one food item in it, the baseline methods cannot detect all the ingredients.

Fig. 4.7 shows the results of food types with difficulty 2. In this category, there could be more than one appearance of the same dominant ingredient. Compared with food types of difficulty 1, DietCam detects more irrelevant ingredients when recall is larger than 0.4. The baseline methods have worse results, since the number of ingredients increase.

Fig. 4.8 shows the results of food types with difficulty 3. In this food type, a food item



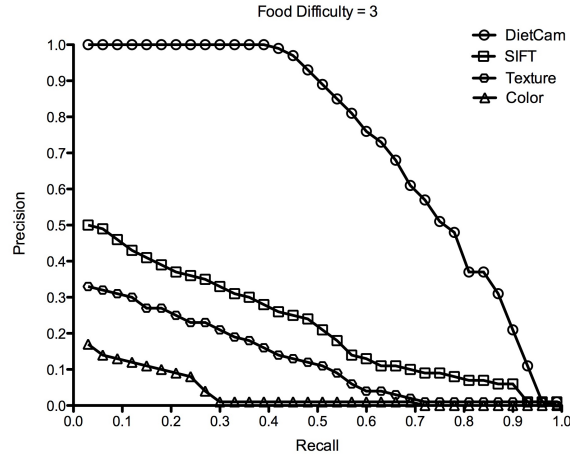
**Figure 4.6:** Precision-Recall curve for models trained on food categories with difficulty 1, single dominant ingredient. In this category, foods appear as a single dominant ingredient. The number of ingredient is 1. The results of DietCam and other three baseline methods are shown.



**Figure 4.7:** Precision-Recall curve for models trained on food categories with difficulty 2, one same ingredient repeats a few times. In this category, foods appear as a combination of the same large ingredient. The number of ingredient is 1 to 5. The results of DietCam and other three baseline methods are shown.

consists of one kind of ingredient, but there could be more than one appearance of this ingredient. The difference between difficulty 2 and 3 is the size of the ingredients. In difficulty 3, the size is much smaller than those in difficulty 2. Typical examples are chopped onions, chopped green peppers, and french fries. DietCam shows a similar results in this





**Figure 4.8:** Precision-Recall curve for models trained on food categories with difficulty 3, one same ingredients repeats many times. In this category, foods appear as a combination of the same small ingredient. The number of ingredient is 1 to infinity. The results of DietCam and other three baseline methods are shown.

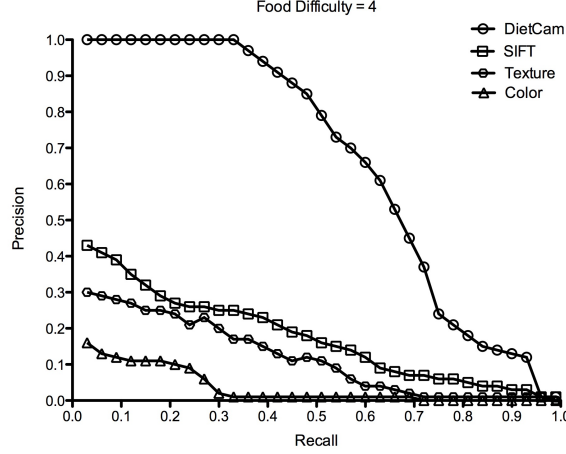
category compared with that in difficulty 2. But the baseline methods cannot detect the smaller ingredients effectively.

Fig. 4.9 shows the results of difficulty 4 food types. In this type of food, more than one type of dominant ingredient presents in the image. DietCam shows similar results compared with difficulty 3. However, the irrelevant number of predicted ingredients increased. The reason is that the number of ingredients needs to be detected increases. With the increasing ingredients in one image, the baseline methods show even worse precision.

Fig. 4.10 shows the results of difficulty 5 food types. In this food category, a food item contains dominant ingredients together with small ingredients. The precision of DietCam decreases due to the increased number of small ingredients. The baseline methods cannot detect those small ingredients effectively. But their precision is not affected too much since they could find the large dominant ingredients.

Fig. 4.11 shows the results of food types with ingredients that are the most difficult to detect. All the ingredients are small and with irregular shapes. DietCam shows a lower precision, while the baseline methods even fail.

The ingredient detection is important since DietCam uses these ingredient to classify food items. From Fig. 4.6 to Fig. 4.11 we can see that the precision decreases when the number of ingredients increases in the images. Then we will see how the ingredient detection



**Figure 4.9:** Precision-Recall curve for models trained on food categories with difficulty 4, more than one dominant ingredients. In this category, foods appear as a combination of different large dominant ingredients. The number of ingredient is more than one. The results of DietCam and other three baseline methods are shown.

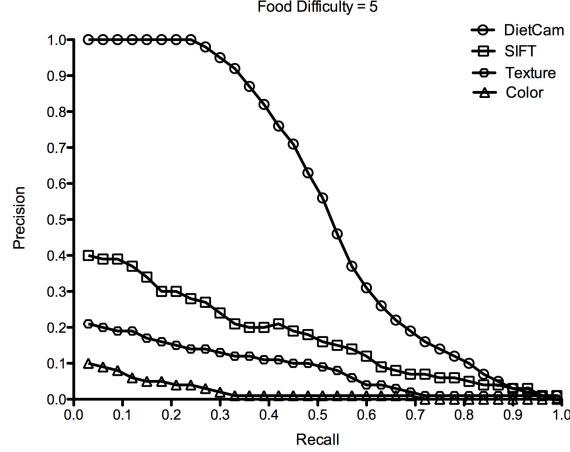
results will affect the performance of the multi-view classifier.

## 4.6.2 Food classification

The goal of this experiment is to evaluate the multi-kernel classifier. Given the precision of the last experiment result, we would like to see how the precision affects the final classification results.

In order to evaluate the multi-kernel classifier, we implement two baseline methods. Similar with the baseline methods in the last experiment, SIFT classifier and texture classifier are used here. In addition, we also implement a version of DietCam that has only a single view kernel to see the contribution of the multi-view kernel.

We use similar baseline methods, which are SIFT with nearest neighbor classifier and texture with SVM. The difference is that we train and test the classifiers with the whole images, rather than with segmented food ingredients in bounding boxes. The indicator we use to evaluate the classifiers is their accuracy, which is defined as the fraction of correct recognition in all the test cases.



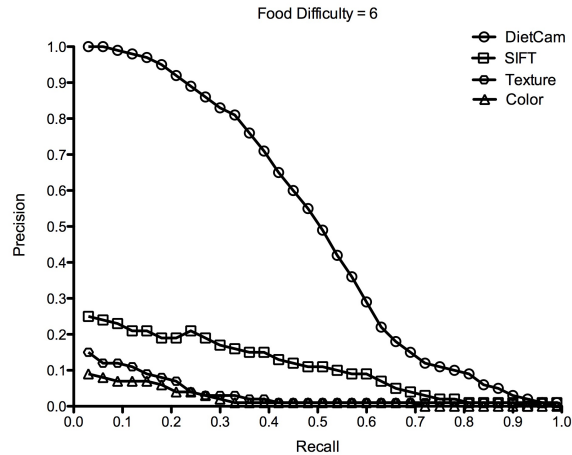
**Figure 4.10:** Precision-Recall curve for models trained on food categories with difficulty 5, dominant ingredients with many small ingredients. In this category, foods appear as a combination of a small number of large dominant ingredients and many small ingredients. The number of ingredient is more than one. The results of DietCam and other three baseline methods are shown.

Fig. 4.12 shows the comparison between DietCam and the SIFT classifier. DietCam presents a constant accuracy of 90%. Even when the difficulty is 5 and 6, DietCam also shows an accuracy higher than 85%, which is much higher than literature methods used in food recognition. SIFT shows an accuracy about 60% for food with difficulty 1. It is the similar results with that used in fast food classification. However, when the food composition is complex, i.e. difficulty is from 2 to 6, SIFT shows an inferior accuracy. When the difficulty is 6, SIFT shows an accuracy of only about 20%, which is unusable in real applications.

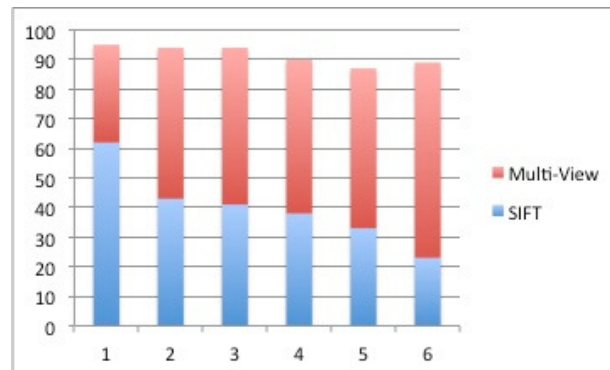
Fig. 4.13 shows the accuracy comparison between DietCam and texture classifier. The accuracy of DietCam is the same with that in Fig. 4.12. The texture classifier shows an accuracy even lower than SIFT. The reason is that food ingredients usually do not have distinctive textures. But DietCam uses the texture information as an verification for part models.

Fig. 4.14 show the accuracy comparison between multi-view kernel and single-view kernel. Without the multi-view consideration, the single-view kernel only achieves an accuracy of 60% to 80%. While multiple viewpoints are used, the accuracy is increased by 10% for difficulty 1 and 2 foods, and about 20% for difficulty 3 to 6 foods.

Fig. 4.15 shows a group of typical images that the food items inside are not recognized correctly. They are from every difficulty group, and they show some common reasons that

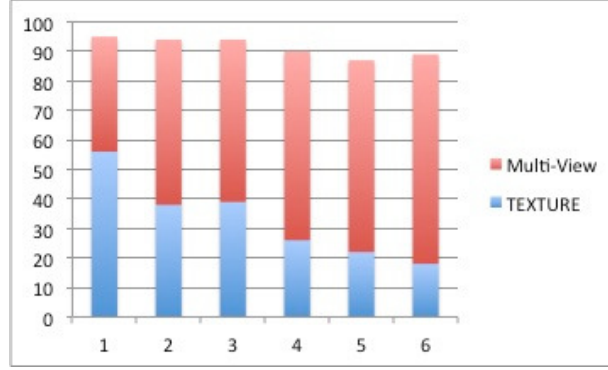


**Figure 4.11:** Precision-Recall curve for models trained on food categories with difficulty 6, a large number of small ingredients repeat many times. In this category, foods appear as a combination of a large number of different small ingredient. The number of ingredient is more than one. The results of DietCam and other three baseline methods are shown.

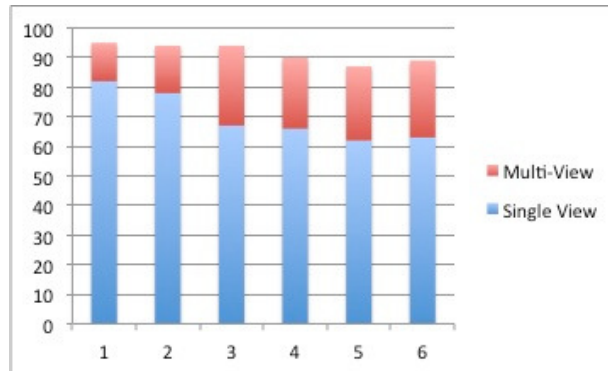


**Figure 4.12:** Food recognition accuracy comparison between DietCam and SIFT classifier. The accuracy statistics are obtained under food types from difficulty 1 to 6. Each bar corresponds to each difficulty. The blue bars show the accuracy of SIFT classifier. The red bars show how DietCam outperforms SIFT, i.e. the accuracy of DietCam is the sum of the blue bar and red bar.

make them hard to recognize. There are four typical reasons that DietCam cannot recognize them. 1. The key ingredients are covered by sources, creams or decoration ingredient. 2. The foods are prepared in irregular shapes. 3. The foods are cooked with special recipes. 4. The food in the image is bitten or left-over.



**Figure 4.13:** Food recognition accuracy comparison between DieCam and texture classifier. The accuracy statistics are obtained under food types from difficulty 1 to 6. Each bar corresponds to each difficulty. The blue bars show the accuracy of texture classifier. The red bars show how DietCam outperforms texture classifier, i.e. the accuracy of DietCam is the sum of the blue bar and red bar.



**Figure 4.14:** Food recognition accuracy comparison between DieCam and single view classifier. The accuracy statistics are obtained under food types from difficulty 1 to 6. Each bar corresponds to each difficulty. The blue bars show the accuracy of single view classifier. The red bars show how DietCam outperforms single view classifier, i.e. the accuracy of DietCam is the sum of the blue bar and red bar. The red bars show the contribution of the multi-view kernel.

## 4.7 Discussion

Food classification is an extremely difficult problem in pattern recognition. The challenges come from the nature of human food compositions. The modern civilization enables human beings to cook and prepare foods with different combinations of ingredients. The cooking methods develop over so long a time that the appearances of foods vary significantly from



**Figure 4.15:** Unrecognized food images. These images are part of those that are not recognized correctly. There are four typical reasons that DietCam cannot recognize them. 1. The key ingredients are covered by sources, creams or decoration ingredient. 2. The foods are prepared in irregular shapes. 3. The foods are cooked with special recipes. 4. The food in the image is bitten or left-over.

raw materials. Followed problem is that even human cannot recognize all the food types.

This paper tries to solve this problem with computers to help people assess their food intakes and foster healthy eating styles. Promising results and increased performance have been achieved compared with other food recognition methods. However, it still need improvements to be a successful field application.

One improvement is to fulfill a complete food image database. Currently, we have a database of 55 popular food classes. But it is still lack of most food categories. Actually collect a complete food list is a difficult problem. Even with these 55 food classes, we encountered difficulties in finding the ground truth in all the images. First, it is a time consuming task to label all the food ingredients. Second, sometimes it is not possible to label all the ingredients in an image. Some ingredients are so small and vague that it is hard to recognize. However, DietCam could detect them correctly, while it is not in the ground truth.

Another possible improvement is the classification method and the computing speed. At present, the classification method needs many computing resources, especially the ingredient detection part. This limits DietCam’s application in everyday life. We plan to implement DietCam on mobile devices such as smart phones that human could take with them almost everywhere. However, the computing resources on the smart phones do not allow food classification on board. Therefore, we implement a food image collection application on the smart phone, and process collected images on remote servers.

Another problem we meet is that some kind of food and ingredients are not naturally separable visually from images. For these types of foods, human beings use other information and experience to recognize them. For example, mayonnaise is usually appeared with salads or sandwiches, while yogurt does not. With the context, people could guess whether the white cream is mayonnaise or yogurt. Similar method will be integrated into DietCam. This is also the reason we use a multi-kernel SVM. We can integrate new food features, new classification methods, even new models into the kernel function.

Our future works will be exploring new features and new models for food classification. They could include ambient sound, human dictation record, geometric location from GPS, models from other sensors, and context modeling.

## **4.8 Conclusion**

Food intake assessment is important to control calorie intakes, which are sources for many public health problems. However, people are still not aware of their food intakes due to a lack of convenient and accurate food intake assessment method. In this paper, we present DietCam, an automatic food recognition method. Concerning the challenges caused by the uncertainties of food appearances, it develops a new food ingredient detector and a multi-view, multi-kernel based SVM to classify food items. We collect a food image database of 15262 food images, among which 50% are used as training images and the other 50% are used for testing. The results are promising compared with existing food classification methods. The accuracy are increased by 60% for the most complex food compositions.



**Table 4.2**  
Food Database Statistics.

Food	Training			Validation			Testing		
	Img	Food	Ing	Img	Food	Ing	Img	Food	Ing
Hoppin' John	62	103	1056	61	107	978	123	211	2011
Buttermilk biscuits	75	128	359	75	121	366	150	234	762
Whole lobster	67	110	130	67	113	143	134	239	242
Shrimp and hushpuppies	78	540	540	77	532	521	155	1009	112
Barbecue ribs steak	65	89	108	65	81	121	130	198	231
Krispy Kreme	65	76	556	64	73	567	129	172	1098
Tacos	73	98	435	73	101	441	146	210	987
Lime pie	66	69	219	66	70	217	132	154	445
Philly cheese steak sandwich	65	80	530	65	76	578	130	156	1231
Pork barbecue sandwich	73	101	880	72	99	760	145	200	1770
Lowcountry boil	72	73	354	72	76	334	144	161	720
Huckleberry pie	74	81	145	74	82	140	148	160	243
Clam chowder	65	76	204	64	69	198	129	155	345
Burger	80	123	549	80	119	567	160	154	989
Eggs Benedict	65	100	193	65	101	191	130	189	432
Pastrami on rye sandwich	63	108	121	63	87	129	126	199	231
Pancakes with syrup	65	68	68	65	67	76	130	123	123
Bagel	64	98	98	64	88	93	128	192	228
Soft pretzel	64	72	72	64	70	73	128	157	156
funnel cake	67	69	135	67	67	125	134	140	267
Snow cone	65	67	67	64	68	70	129	142	140
Smoked salmon	61	99	109	61	92	103	122	207	234
Persimmon pudding	76	97	99	76	98	98	152	211	254
Corn dog	79	86	89	79	91	108	158	178	202
French fry	61	98	109	60	77	112	121	187	215
Chicken wings	71	78	78	70	73	84	141	143	156
Drink	72	113	132	71	121	145	143	257	278
Chili dog	65	81	342	64	79	356	129	175	723
Spam musubi	72	231	681	71	240	665	143	460	1428
Fluffernutter sandwich	65	77	357	65	78	360	130	155	750
Cookie	73	78	82	73	76	90	146	156	150
BLT sandwich	69	80	459	69	85	450	138	166	924
Baked beans	65	68	680	65	71	657	130	133	1428
Pumpkin pie	60	64	134	60	66	135	120	143	266
Fajitas	58	59	335	57	58	340	115	122	680
Succotash	63	69	379	62	69	411	125	151	760
Cornbread	61	66	79	61	65	90	122	136	165
Barbecue chicken pizza	71	74	657	71	76	660	142	137	1328
Chicken fried steak	73	77	157	73	76	140	146	170	325
Burrito	75	79	349	75	78	350	150	155	766
Pecan pie	72	78	180	71	77	195	143	145	354
Catfish	74	90	213	73	91	231	147	185	454
Mashed potato	76	89	95	76	88	106	152	190	210
Meatloaf	76	87	95	76	85	113	152	185	207
Green bean casserole	75	79	130	75	79	143	150	157	257
French's fried onions	76	109	457	76	106	435	152	178	956
Sopaipillas	65	93	250	65	90	260	130	194	468
Cheesecake	66	75	88	65	70	99	131	159	533
Turkey sandwich	65	97	211	65	87	231	130	206	466
Salad	78	83	355	78	80	320	156	183	779
Fried rice	78	81	320	78	80	313	156	166	620
Pasta	79	83	449	79	86	446	158	178	993
Noodles	79	80	414	79	90	430	158	195	820
Steaks with broccoli	67	156	177	66	145	198	133	323	330
Sushi	75	83	166	75	85	210	150	177	340
Total	3824	5366	15726	3807	5275	15772	7631	10818	31582



# References

- [1] F. Kong and J. Tan, “Dietcam: Regular shape food recognition with a camera phone,” *Body Sensor Networks (BSN), 2011 International Conference on*, pp. 127 – 132, 2011.
- [2] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, “Food recognition using statistics of pairwise local features,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2249 – 2256, 2010.
- [3] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [4] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [5] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [6] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Computing Surveys*, vol. 35, no. 4, pp. 399–458, Jan 2003. [Online]. Available: <http://portal.acm.org/citation.cfm?id=954339.954342>
- [7] M. Weber, M. Welling, and P. Perona, “Unsupervised learning of models for recognition,” *Computer Vision, European Conference on*, pp. 18 – 32, Jan 2000.
- [8] R. Szeliski, “Computer vision: Algorithms and applications,” *Springer*, 2010.
- [9] C. Martin, S. Kaya, and B. Gunturk, “Quantification of food intake using food image analysis,” *Engineering in Medicine and Biology Society. Annual International Conference of the IEEE*, pp. 6869 – 6872, 2009.
- [10] 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,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 4, pp. 756 – 766, 2010.

- [11] W. Wu and J. Yang, "Fast food recognition from videos of eating for calorie estimation," *Multimedia and Expo, IEEE international Conference on*, pp. 1210 – 1213, Jan 2009.
- [12] G. Godin, A. Bélanger-Gravel, A. marie Paradis, M.-C. Vohl, and L. Pêrusse, "A simple method to assess fruit and vegetable intake among obese and non-obese individuals," *Can J Public Health*, vol. 99, no. 6, pp. 494–8, Jan 2008.
- [13] M. A. Murtaugh, K. ni Ma, T. Greene, D. Redwood, S. Edwards, J. Johnson, L. Tom-Orme, A. P. Lanier, J. A. Henderson, and M. L. Slattery, "Validation of a dietary history questionnaire for american indian and alaska native people," *Ethn Dis*, vol. 20, no. 4, pp. 429–36, Feb 2011.
- [14] N. D. Wright, A. E. Groisman-Perelstein, J. Wylie-Rosett, N. Vernon, P. M. Diamantis, and C. R. Isasi, "A lifestyle assessment and intervention tool for pediatric weight management: the habits questionnaire," *J Hum Nutr Diet*, vol. 24, no. 1, pp. 96–100, Feb 2011.
- [15] A. F. Smith, S. D. Baxter, J. W. Hardin, C. H. Guinn, and J. A. Royer, "Relation of children's dietary reporting accuracy to cognitive ability," *Am J Epidemiol*, vol. 173, no. 1, pp. 103–9, Jan 2011.
- [16] L. A. Mainvil, C. C. Horwath, J. E. McKenzie, and R. Lawson, "Validation of brief instruments to measure adult fruit and vegetable consumption," *Appetite*, vol. 56, no. 1, pp. 111–7, Feb 2011.
- [17] M. A. Cardoso, L. Y. Tomita, and E. C. Laguna, "Assessing the validity of a food frequency questionnaire among low-income women in são paulo, southeastern brazil," *Cad Saude Publica*, vol. 26, no. 11, pp. 2059–67, Nov 2010.
- [18] F. H. Esfahani, G. Asghari, P. Mirmiran, and F. Azizi, "Reproducibility and relative validity of food group intake in a food frequency questionnaire developed for the tehran lipid and glucose study," *J Epidemiol*, vol. 20, no. 2, pp. 150–8, Jan 2010.
- [19] A. E. Dutman, A. Stafleu, A. Kruizinga, H. A. Brants, K. R. Westerterp, C. Kistemaker, W. J. Meuling, and R. A. Goldbohm, "Validation of an FFQ and options for data processing using the doubly labelled water method in children," *Public Health Nutr*, pp. 1–8, Aug 2010.
- [20] M. Aubertin-Leheudre, A. Koskela, A. Samaletdin, and H. Adlercreutz, "Plasma alkylresorcinol metabolites as potential biomarkers of whole-grain wheat and rye cereal fibre intakes in women," *Br J Nutr*, vol. 103, no. 3, pp. 339–43, Feb 2010.
- [21] G. L. Bowman, J. Shannon, E. Ho, M. G. Traber, B. Frei, B. S. Oken, J. A. Kaye, and J. F. Quinn, "Reliability and validity of food frequency questionnaire and nutrient biomarkers in elders with and without mild cognitive impairment," *Alzheimer Dis Assoc Disord*, vol. 25, no. 1, pp. 49–57, Jan 2011.

- [22] P. B. Ryan, K. A. Scanlon, and D. L. MacIntosh, "Analysis of dietary intake of selected metals in the nhexas-maryland investigation," *Environ Health Perspect*, vol. 109, no. 2, pp. 121–8, Feb 2001.
- [23] M. R. Ritchie, M. S. Morton, N. Deighton, A. Blake, and J. H. Cummings, "Plasma and urinary phyto-oestrogens as biomarkers of intake: validation by duplicate diet analysis," *Br J Nutr*, vol. 91, no. 3, pp. 447–57, Mar 2004.
- [24] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination," *European Conference on Computer Vision*, Jan 2002.
- [25] C. Schmid, "Constructing models for content-based image retrieval," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. II–39 – II–45, 2001.
- [26] J. D. Bonet and P. Viola, "Texture recognition using a non-parametric multi-scale statistical model," *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 641 – 647, 1998.
- [27] M. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance," *Image Processing, IEEE Transactions on*, vol. 11, no. 2, pp. 146 – 158, 2002.
- [28] Y. Xu, X. Yang, H. Ling, and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 161 – 168, 2010.
- [29] M. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1447 – 1454, Jan 2006.
- [30] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 – 8, 2008.
- [31] P. Viola and M. Jones, "Robust real-time face detection," *Computer Vision, IEEE International Conference on*, vol. 2, pp. 747 – 747, Jul 2001.
- [32] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 746 – 751, Jun 2000.
- [33] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

- [34] M. Sadeghi and A. Farhadi, “Recognition using visual phrases,” *Computer Vision and Pattern Recognition, IEEE Conference*, pp. 1745–1752, 2011.
- [35] H. Jia and A. Martinez, “Support vector machines in face recognition with occlusions,” *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 136 – 141, May 2009.
- [36] N. Cristianini and J. Shawe-Taylor, “An introduction to support vector machines,” *Cambridge University Press*, 2000.
- [37] S. Maji, A. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 – 8, 2008.
- [38] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW ’06. Conference on*, p. 13, 2006.
- [39] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” *Conference On Image And Video Retrieval, Proceedings of the 6th ACM international conference on Image and video retrieval*, Jan 2007.
- [40] K. Grauman and T. Darrell, “The pyramid match kernel: discriminative classification with sets of image features,” *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1458 – 1465 Vol. 2, 2005.
- [41] L. Duan, D. Xu, I. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data,” *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1959 – 1966, 2010.
- [42] M. Varma and D. Ray, “Learning the discriminative power-invariance trade-off,” *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1 – 8, 2007.
- [43] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, “Multiple kernels for object detection,” *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 606 – 613, 2009.
- [44] P. Gehler and S. Nowozin, “Let the kernel figure it out; principled learning of pre-processing for kernel classifiers,” *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2836 – 2843, 2009.
- [45] M. Chen, K. Dhingra, W. Wu, and L. Yang, “PFID: Pittsburgh fast-food image dataset,” *Image Processing, IEEE International Conference on*, pp. 289 – 292, Jan 2009.

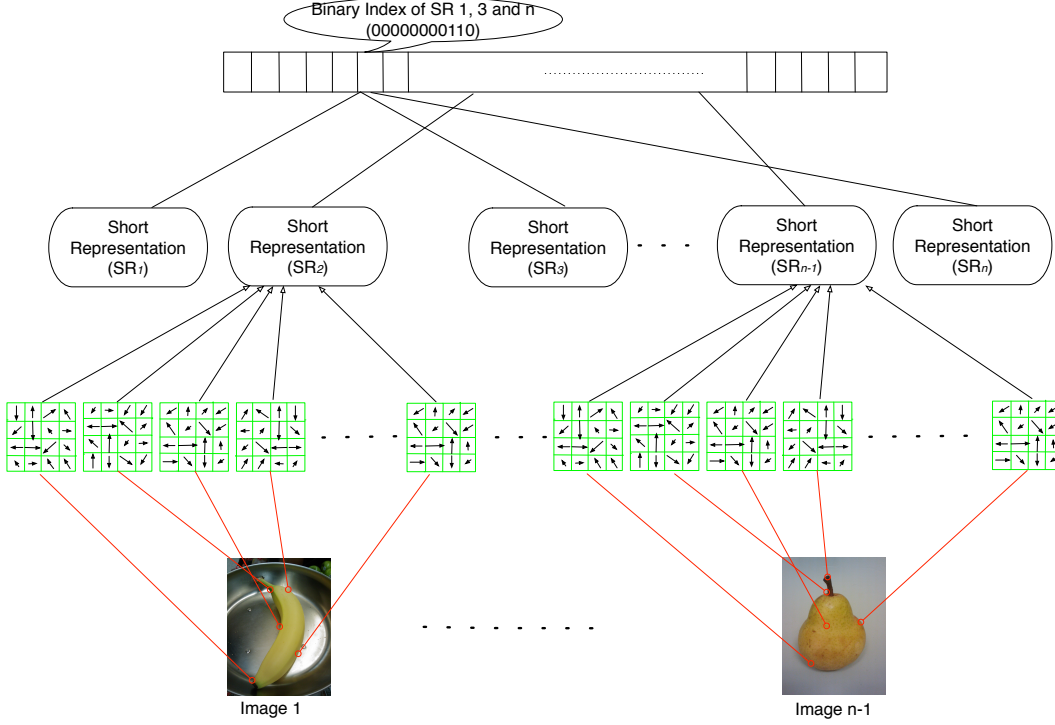


## **Chapter 5**

# **Indexing of Bags of Features: Efficient Image Retrieval from Large-Scale Database**

### **Abstract**

Efficient and accurate image retrieval from a large-scale database is a time consuming task and bag-of-features is a popular method for image retrieval and recognition. This paper presents an image indexing method for large-scale image retrieval and image recognition, based on the bags of features of the image. Existing image retrieval methods accelerating the retrieval speed by means of shortening the image features and developing fast nearest neighbor image feature search algorithms. Rather than indexing every feature, our approach indexes every image with a short binary string and retrieves images faster than feature indexing methods. Features of an image are combined into a short representation with vector quantization methods. Then spatial hashing method is developed to hash the short representations into binary indices of 8 to 16 bits, which makes it possible to perform near real-time image retrieval from millions of images. The experiments show a consistent fast and precise retrieval result.



**Figure 5.1:** The hierarchical structure of the index. From the bottom, every of the  $n$  images is represented by the SIFT descriptors. The descriptors of an image are quantized into a short representation. Then short representations will be hashed into binary indices. One binary index corresponds to more than one short representation. One short representation could represent more than one image.

## 5.1 Introduction

Image classification and augmented reality applications on networked hand-held smart devices with cameras are becoming more and more popular. Bag-of-features [1] has been proven one of the most effective and efficient methods for image classification tasks. As the storage of digital images grows significantly in the past years, it is more time consuming to retrieve and classify images from a large-scale image database through pair-wise feature matching. It is also difficult for an embedded device to query the image database with a large number of high dimensional features. Efficient indexing of the image and image classes in the database is a solution.

Current researches in image retrieval and indexing pay considerable attentions to work on the image descriptors. Researchers tend to simplify the retrieval problem through reducing descriptors dimensions [2–4] and developing fast descriptor classifiers [5–8]. However, those methods are still suffering from several problems such as high memory usage, low

training speed, infeasibility to include enough sample images and manual labeling.

Our method is motivated from the idea that in a database system the search time will not increase linearly with the database size with an efficient indexing mechanism. One example is a database with a tree structure index, where the speed of data retrieval operations is relative to the levels of the tree rather than the number of data entries. However, setting up such an index of the image descriptors could be difficult even impossible for an image database. The high dimensional feature descriptors will make the index tree too large in the memory and slow down the searching speed when comparing descriptors. Another fact is that every image has a various number of descriptors, which also makes the indexing of images harder. Rather than on the image descriptor level, we contribute our work on the object class level. Our goal is to develop a short binary representation of bags of features of images as the index meanwhile preserving the distinctiveness of the images.

Indexing bags of features will be more difficult than indexing a single feature. The binary indices of features could be generated through hashing, which has drawn attention in content-based image retrieval and computer vision communities. Both supervised [9–11] and unsupervised hashing [2] methods have been developed for simplified representation of image descriptors. A good example is Locality-sensitive hashing [2]. Based on these binary feature representations, various approximate nearest neighbor classifiers [7, 8] have been investigated for image retrieval. But generating binary representations for bags of features of images still draws few attentions and is still challenging. The generalization of different number of descriptors for different images and the conversion from the image generalization to the binary index will be difficult especially when considering the significant information lost.

In this paper, we explore efficient image retrieval from a large image database. The images are described with bags of SIFT descriptors [12]. All the SIFT descriptors belonging to the same image are quantized into a short representation of the object in the image. This representation has the same dimension as SIFT. Binary indices will be generated from the short representations through hashing. Fig. 5.1 shows the hierarchical structure of the new index mechanism. We found that the new SIFT quantization method and hashing function indexes the images effectively and this indexing method shortens the image retrieval time significantly and enhances the image retrieval precision.



## 5.2 Related Work

Bag-of-features has been widely used in image recognition and image retrieval. However, pair-wise descriptor comparison between images is time consuming especially in a large image database. The vocabulary tree [13] and vocabulary forest [14] are efficient descriptor matching methods in a database of tens of thousand images. The descriptors are indexed into a tree structure by hierarchical clustering. The major problem limiting vocabulary tree's application in large-scale image retrieval is the high memory usage, which is linear to the number of leaf nodes [13]. In the large-scale image database, recent works can be grouped mainly into two categories.

One category of the researches tries to shorten the pair-wise descriptor comparison time by means of reducing the descriptor dimensions. Take the popular SIFT descriptor for example. The dimension of an SIFT descriptor is 128, which is definitely too long for distance calculation. Principal component analysis has been used to shorten the descriptors [3, 4]. Another popular method is hashing. Unsupervised hashing such as Locality-Sensitive Hashing (LSH) [2] has been widely used to map an image descriptor to a randomly generated binary code. It works well for image descriptors since the result binary code preserves the metric distance property between descriptors. However, it is suffering from the high memory usage problem. Supervised hashing methods, such as BoostSSC and Restricted Boltzmann Machine, have been developed to reduce the memory usage with labeled images [9–11]. They are good at preserving the semantic similarities between images, but the low training speed and labeling process limit its usage in a large-scale database. Recently, semi-supervised hashing method [15] has been proposed to combine the advantage of metric similarity from LSH and semantic similarity from supervised hashing.

Another category works on developing efficient classification algorithms. One of the earliest methods applied on SIFT descriptors is the Best-Bin-First search [12], which clusters the descriptors of an image into a k-d tree structure. It reduces the pair-wise descriptor comparison times from the number of descriptors  $O(N)$  to the level of the trees  $O(\log(N))$ . K-d tree has been optimized for better space partitioning and better nearest neighbor classification performance [7, 8]. The nearest neighbor classifier has also been modified for efficient descriptor search among a large number of image descriptors [5, 6]. In order to reduce the manual labeling of the images, boosting has also been used for visual similarity learning [16]. The performance of these nearest neighbor and kernel methods is limited since it is infeasible to include enough training samples.

Some other related works also exist for image retrieval such as the development of global image descriptors instead of local descriptors [17], bundling the local image descriptors [18] and efficient matching through spatial pyramid matching [19]. However, together

with the two main categories, all the methods above work on the descriptor level. Our approach works on the image level. The binary index of an image is calculated with the bag-of-features in the image. The image index indicates which group of images most likely contains this image features.

### 5.3 Short Representations of Images

The image indexing mechanism starts from shortening the image features into an image short representation. An image could contain a large number of features and the quantity of the features could be different from image to image. Therefore, it will be difficult to calculate a binary index of the image directly from the features. In this section, a short representation of the image is calculated first from the image features. The length of the short representation is the same with a single image feature, i.e. 128 elements if SIFT is used. Then in next section, the binary index of an image will be calculated from the short representations.

The image features  $Feat$  is composed of a set of  $n$  feature points  $Feat = \{f_1, f_2, \dots, f_n\}$ . The goal of short representation is to compute a short vector  $SR \in \mathbb{R}^l$  from  $Feat$  to describe the image  $I$ . At the same time,  $SR$  is required to preserve the distinctiveness of  $Feat$ . One benefit  $SR$  brings to us is that every image could be represented as a feature vector of a certain length  $l$ , which makes it convenient in the next section to map  $SR$  to binary indices  $Index$  ( $Index \in \mathbb{R}^d$ , where  $d$  is the number of bits of the binary index). However, it is challenging to represent an image with such a short representation meanwhile preserve the distinctiveness of each image, since the image distinctiveness is identified as differentiable SIFT features.

The image features could be shortened through several ways, such as selecting the most important feature, classifying the features into a class, group all the features together, and extracting properties of the features. A representative feature could be selected randomly or purposely as the short representation of the image. It is simple however the information lost is considerable. Classification methods could be used to classify the image features into a group and the identification of the group would be the short representation of the image. Designing such a classifier will be non-trivial, since an image could contain non-salient features and the same feature could also appear in other images. In this section, three SIFT manipulation methods are proposed based on feature combination and feature property extraction. Their performance of image distinctiveness preservation will be compared in the experiment.

### 5.3.1 The average of SIFT features

A straightforward method to get the  $SR$  of an image is to cluster the image features in the feature space and choose the cluster center as this image's  $SR$ . Formally, in the database  $I$ , the  $k$ th image  $I_k$  is represented by a series of  $n$  SIFT features  $\{f_1^{(k)}, f_2^{(k)}, \dots, f_n^{(k)}\}$ . The short representation of image  $k$  is defined as

$$SR(I_k) = \frac{\sum_{i=1}^n f_i^{(k)}}{n}. \quad (5.1)$$

In this way, the image is represented by the average point of its SIFT features in the SIFT space and the result will be a vector of the same dimension with a single SIFT feature. The distinctiveness of the most dominant features would be preserved whereas that of the minor features could be pruned away. The algorithm is simple and fast. However the distinctiveness of individual features could be lost due to the average. It only takes the average quantity of image features into account rather than the importance.

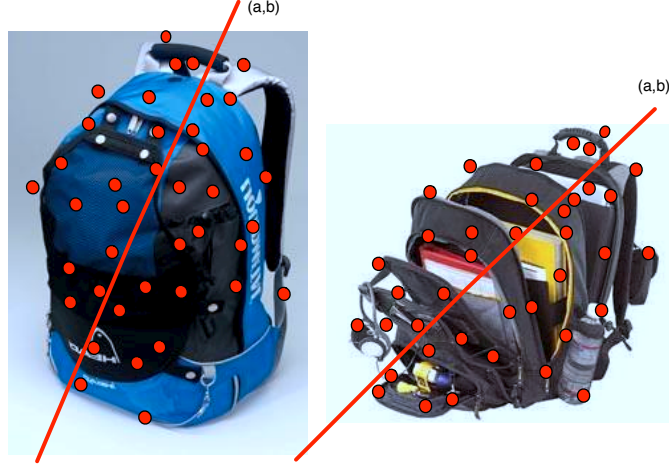
### 5.3.2 The weighted average of SIFT features

Since the average point of image features in the feature space does not take the importance of individual feature point into consideration, a weighted cluster center could be used. The weight  $w$  of a feature point is introduced into equation(5.1). The importance of a feature point  $w$  is directly related to the point's position  $(x, y)$  in the image. It is natural to see in most photographs, the photographer composites the most important subject in the center of the image. In few times photographers composite the most important subject in other places purposely. Therefore we can model the importance of the features points in an image as a two dimensional Gaussian distribution on the image coordinates. The mean is the center of the images and the variance is chosen as half of the image dimension in the experiment.

Hence, the short representation of image  $k$  is defined as

$$SR(I_k) = \frac{\sum_{i=1}^n w_i f_i^{(k)}}{n}, \quad (5.2)$$

where  $w$  is the weight of feature  $f_i^{(k)}$ , which is defined as a two dimensional Gaussian distribution



**Figure 5.2:** An example of using PCA to reveal the internal property of a set of points. These two images are from Caltech 256's Backpack dataset, and the red feature points are detected with SIFT. PCA calculates the predominant orientation of these points and represent the orientation with a line  $(a,b)$ , which could be used as a shorter representation of the points.

$$w(i) = e^{-\left(\frac{(x_i - x_o)^2}{2\sigma_x^2} + \frac{(y_i - y_o)^2}{2\sigma_y^2}\right)} \quad (5.3)$$

### 5.3.3 PCA of SIFT features

Another way to describe the image with a short representation is to combine all the image features together into a long vector and reduce the dimension of the result vector. Principal Component Analysis (PCA) [20] is a standard technique for dimension reduction. It has been applied to a broad class of problems in computer vision field, such as feature description [4], feature selection [21] and object recognition [22]. While PCA suffers from a number of known shortcomings such as the restriction to orthogonal linear combinations [4], it is still popular due to its simplicity. Considering its ability to reveal the internal structure of the data through explaining the variance with the first few principal components, we apply PCA here to reduce the dimension of the combined image features.

The problem is to classify *Feat* to a group according to the properties of feature points. This property is chosen as the orientation of the points in the feature space. Fig. 5.2 shows an example in two dimensional space. The predominant orientation of the points in the  $\{x,y\}$  space represents the most significant property of the distribution of the points with only a two-dimension vector. In the similar way, the *SR* of an image could be obtained from the

predominant orientation of  $Feat$  in the feature space. PCA is well suited for determining object orientation and rotation. According to the standard procedure of PCA

$$SR(I_k) = \mathbf{A}(\mathbf{Feat} - \mathbf{m}_{Feat}), \quad (5.4)$$

$\mathbf{Feat}$  is a vector of image features  $Feat$ ,  $\mathbf{m}_{Feat}$  is a vector of the mean of these features. The dimension of vector  $\mathbf{Feat}$  is as high as the number of features  $n$  extracted from the image. This high dimensional vector  $Feat$  is projected into a one dimension vector of image feature  $SR(I_k)$  through matrix  $\mathbf{A}$ , which is determined by the covariance matrix  $\mathbf{C}_{Feat}$  of  $\mathbf{Feat}$ .

$$\mathbf{A} = [e_1, e_2, \dots, e_n]^T \quad (5.5)$$

Rows in matrix  $A$  are eigenvectors  $e_j \in \mathbf{e}$  of the matrix  $\mathbf{C}_{Feat}$ , ordering according to the corresponding eigenvalues in descending order. In our implementation, we choose the first eigenvector in matrix  $\mathbf{A}$ , so the dimension of the short representation  $SR(I_k)$  is the same with an image feature.

### 5.3.4 Image distinctiveness of SRs

The image distinctiveness is preserved through the distance between image SRs. Let the function  $Similar(I_j, I_k)$  represent the similarity between image  $I_j$  and  $I_k$ , function  $D(SR_j, SR_k)$  represent the distance between the short representations  $SR_j, SR_k$  of image  $I_j$  and  $I_k$ . We have,

**Lemma 1:** the larger  $D(SR_j, SR_k)$  between two SRs, the smaller  $Similar(I_j, I_k)$  value, i.e. the more distinctive two images will be.

**Proof:** This can be proved from its converse-negative proposition, similar images would have similar short representations. Every image  $I$  is represented as a set of  $n$  features,  $I_j = \{f_1^j, f_2^j, \dots, f_n^j\}$ . Given two images, the image similarity is defined as the sum of the distance of the top  $r$  closest feature pairs. In order to remove the outliers in the image features,  $r$  is chosen as the 70% of the smaller  $n_j$  and  $n_k$  empirically. Formally,

$$Similarity(I_j, I_k) = \sum_{i=0}^{\lfloor r \times (n_j < n_k ? n_j : n_k) \rfloor} d(f_i^j, f_i^k), \quad (5.6)$$

where  $d(f_i^j, f_i^k)$  is the Euclidean distance between the feature pair  $f_i^j, f_i^k$ . It is clear that the smaller  $Similarity$  is, the more similar two images are.

Since the dimension of the short representation are the same, which is the dimension of an SIFT feature, we define the distance between the short representations as the Euclidean distance between them. Formally,

$$D(SR_j, SR_k) = d(SR_j, SR_k) \quad (5.7)$$

We will show the smaller the value of  $Similar(I_j, I_k)$ , the smaller the value will be achieved for  $Distance(SR_j, SR_k)$ . Consider three images  $I_j, I_k, I_l$ , if  $Similar(I_j, I_k) < Similar(I_k, I_l)$ , then  $Distance(SR_j, SR_k) < Distance(SR_k, SR_l)$ . The expressions of  $Similar(I_j, I_k)$  and  $Distance(SR_j, SR_k)$  could be expanded here.

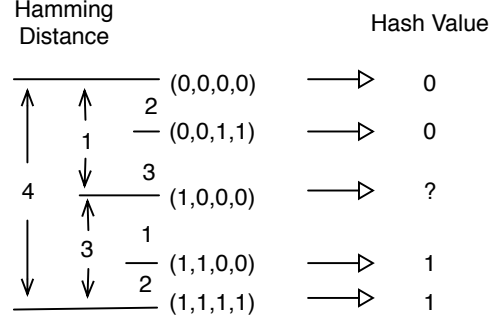
$$Similarity(I_j, I_k) = \sum_{i=0}^{\lfloor r \times (n_j < n_k ? n_j : n_k) \rfloor} \sqrt{\sum_{i=1}^{128} (f_{ni}^j - f_{ni}^k)^2} \quad (5.8)$$

$$D(SR_j, SR_k) = \sqrt{\frac{1}{128} \sum_{i=1}^{128} (SR_{ji} - SR_{ki})^2} \quad (5.9)$$

That becomes when the average distance increases, the distance of the average feature points will also grow up. It will be true when in the sampled feature points there are no outliers, which has been removed from the definition of  $Similarity(I_j, I_k)$ . The distinctiveness preservation property of the SRs has also been proved in the experiment 5.6.1.

## 5.4 Binary index of short representations

The image features are projected to image short representations, which is proved to be good at preserve the distinctiveness between images. The short representations will be mapped to binary indices, which index every image with an integer. It is obvious that one index could correspond to more than one image especially when a large number of images exist in the database. Features in these images with the same index will be used as vocabularies of this index. Therefore, the database is divided into several parts, each of which contains images of the same index. A vocabulary tree is built for a part and the index of the images is also the index of that vocabulary tree.



**Figure 5.3:** One example of bit assignment. When 0 is assigned to (0,0,0,0), (1,1,1,1) is naturally to be 1 to reflect their large hamming distance. Similarly, (1,1,1,0) and 0,1,0,0 are hashed to be 1 and 0 respectively. The problem is how to assign the value of (1,0,0,1), which has the same hamming distance to both groups.

An intuitive method to map a non-binary vector into binary is the hashing method. Hashing started in the database field and achieved extensive applications in computer vision society recently. Locality Similarity Hashing (LSH) [2] has been proven an efficient hashing scheme for approximate similarity search in high dimensional spaces. We present spatial hashing, which maps a binary set into another binary set of much fewer bits. The idea is to hash the SRs into binary codes of the same dimension based on LSH, and then to hash the binary codes into shorter binary indices.

### 5.4.1 Locality similarity hashing

The basic idea of LSH is to hash the SRs from the images so as to ensure a higher probability of collision for closer SRs than those that are far apart. Because of the uncertainty of this method, several hash functions are used to map the SR to different hash tables. More formally, a SR will be hashed by  $s$  different hash functions  $h_r (r \in [0, 1, \dots, s])$ . The results are stored in  $s$  hash tables.

Each hash function  $h_r$  is parameterized by two vectors,  $E_r = \langle E_1^r, E_2^r, \dots, E_{128}^r \rangle$  and  $T_r = \langle T_1^r, T_2^r, \dots, T_{128}^r \rangle$ . The dimension of vectors  $E$  and  $T$  is the same with SR's dimension, which is 128. Values of  $E_r$  are randomly drawn in  $[1, 2, \dots, 128]$ . Vector  $T_r$  stores thresholds for every single bit. Its element  $T_i^r$  is randomly drawn in  $[0, 1, \dots, C_i]$ , where  $C_i$  is the maximum value of the hashed vectors along the  $i$ th dimension.

Hash function  $h_r$  maps SR from  $\mathbb{R}^{128}$  to  $[0, 1, \dots, 2^{128} - 1]$ . The result  $h_r(SR)$  is computed

as a 128-bit binary string  $b_0^r, b_1^r, \dots, b_{127}^r$ , such that:

$$b_j^r = \begin{cases} 0, & \text{if } (SR(D_j^r) < T_j^r); \\ 1, & \text{otherwise.} \end{cases}$$

This 128-bit string is the hash index for the SR in the  $r$ th hash table. Modification of  $r$  can tune the accuracy and speed of the algorithm. When  $r$  is chosen high, the storage space required to store the  $r$  hash tables is very large. Therefore, even though the 128-byte SRs could be mapped to 128-bit binary indices, the results are still too long.  $r$  is chosen as 10 in the experiment and spatial hashing is developed in the next section to project the index shorter further.

### 5.4.2 Spatial hashing

As the final indices are desired to be short binary strings, a novel spatial hashing method is developed to reduce the dimension of binary codes. From the first bit of the binary code, spatial hashing picks a number of  $n$  bits, assigns them a binary value, then picks another  $n$  bits, assigns a value, and so on, until there are no bits left.

The hamming distance is used as the metric to measure image similarities. The value assigned to the group should maintain the distinctiveness between the binary codes. For example, when four consecutive digits are combined into a group, 0 and 1 could be assigned to (0000) and (1111) respectively to reflect their difference. However, not all the four-bit value could be assigned a one-bit value easily. Figure 5.3 shows such an instance.

Spatial hashing divides the space into two parts  $U$  and  $V$ , to whom 0 and 1 are assigned respectively. In order to maintain the distance between  $U$  and  $V$ , the space-dividing scheme should maximize the distinctiveness between them  $D(U, V)$ , which is modeled as the differences between inter-part hamming distances and intra-part hamming distances. More formally,

$$D(U, V) = 2 \times \sum_{i=1}^m \sum_{j=1}^n H(U_i, V_j) \quad (5.10)$$

$$- \sum_{i=1}^m \sum_{j=1}^m H(U_i, U_j) - \sum_{i=1}^n \sum_{j=1}^n H(V_i, V_j) \quad (5.11)$$



This will be an optimization problem. The space  $B$  is partitioned into two subsets  $U$  and  $V$  that have the same number of elements. The problem is to find a partition scheme  $P$ , satisfying

$$\operatorname{argmax}_{U,V} D(U,V), \quad (5.12)$$

such that  $U \cap V = \emptyset$  and  $U \cup V = B$ .

This problem could be solved through brute-force search when the number of bits per group is not too large, for example no more than sixteen. The result binary indices will have dimensions of 32, 16 or 8, when the number of bits in the group is 4, 8 or 16.

One property of the spatial hashing is that similar images will be hashed into the same index so that the nearest neighbor search algorithms will have a high retrieve precision when search an image with the same index in these images. Another benefit spatial hashing brings is that compared with constructing a large even impossible vocabulary tree for all the features in the database, it is possible to divide the large database into parts with semantic indices and vocabulary forests could be built. The results are presented in the experiment 5.6.2.

## 5.5 Food Database

An important application of this paper is food recognition and food image retrieval. We develop a hierarchy of two databases for efficient food recognition, a global database and a small personal database. The global database stores a large number of food types. Noticing the large size and the slow searching time in the global food database, a small personal database is developed as a cache in the image manager. The image manager has the image recording function besides extracting SIFT features. The images recorded will form a small personal food database. The fact that people are more likely to have a certain dietary style gives this feasibility. Considering the high possibility of food recurrence, it will be valuable to keep a record of what kind of food the users have eaten. When looking for the food types, this small database will have a higher hit rate compared with the large global database. In this way, before looking up in the large database, the personal database will be checked first.

The main contents of the food database are food types, visual descriptions of each food type, and their nutrition information. The database is built from the most popular food types including fast food, steak meals, fruits, and other high-calorie foods. The images are collected manually from the developers' input and from a food image website. Every type

of food is associated with SIFT features that describe its characteristics in images. The features are clustered into visual words with an efficient hierarchical k-means clustering algorithm. The visual words are stored in the database. The calorie density information of a type of food is another key content in the database. The USDA Food and Nutrient Database provides an accurate energy measure.

In the database, a food type will have multiple visual descriptions to cover more lighting and perspective possibilities. Food images taken in different settings are chosen as training images. Each of these training images contains only one food item. In this way, the features of this image will be a clean description of the food item contained, rather than being messed up by other food types.

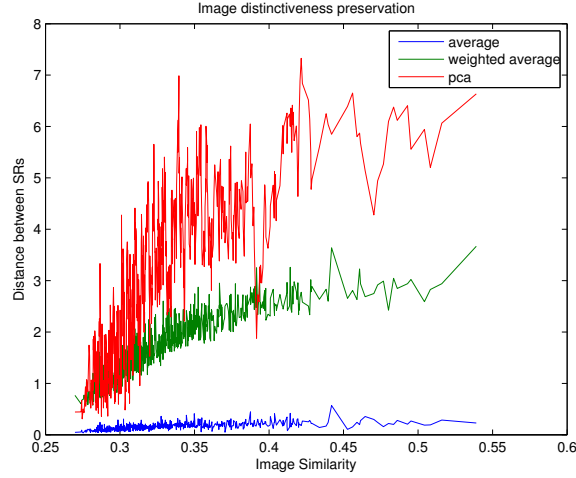
## 5.6 Experiment

In the experiments, image retrieval speed and precision will be evaluated. Given a large database of images, one goal is to evaluate the distinctiveness between the short representations of these images. Another goal is to find a set of images that are similar to a given input image with the help of binary index. We use two datasets, one of 30607 images from Caltech 256 [23] and another of 11 million from ImageNet [24]. The larger ImageNet is used as the image database and Caltech 256 provides the query images. The image features are extracted from both of them through the dense SIFT algorithm from VIFeat [25].

A vocabulary tree has been constructed based on features from Caltech 256. It takes up to 354 MB memories. It is believed the memory usage is linear in the number of leaf nodes of the vocabulary tree, which is decided by the dimension and level of the tree. Therefore the memory usage of the vocabulary tree of images in ImageNet would be huge, even too large for today's computers.

### 5.6.1 Short representation

The goal is to evaluate how three different SR algorithms preserve image distinctiveness. Since it is very slow to calculate the pair-wise distances between all the 11 million images in the database, we randomly select 10,000 images from the database as samples. The ground truth of the image similarity is defined by equation(5.8) and calculated pair-wise. The SRs are quantized from the image features with three algorithms, average, weighted average, and PCA. We process the results for easy comparison. The weights of the weighted



**Figure 5.4:** The distances between SRs vs. the similarity between images. 10000 images are sampled randomly from the ImageNet Database and their pair-wise distances are calculated and shown in the  $x$  axis. Accordingly, the SRs of these images are calculated with three algorithms and the pair-wise distances between SRs are shown in the  $y$  axis. The PCA algorithm preserves the distinctiveness of the images best.

average algorithm are converted from  $(0 - 1)$  to  $(0 - 100)$ , so that the results will not be too small. There could be negative values in the element of SRs from the PCA algorithm; we also convert the SR through adding a constant value to make all the elements in SRs positive.

Fig. 5.4 shows the results on these 10,000 images. Their pair-wise distances are calculated and shown in the  $x$  axis. Accordingly, the SRs of these images are calculated with three algorithms and the distances are shown in the  $y$  axis. Even though there are disturbances and noises on these curves, the trend is still that higher SR distances are more likely to have larger image similarities. Top performance is reached by PCA especially when the image similarity is large. Weighted average performs better than the average algorithm. When the similarity grows, the distance between SRs calculated with the average algorithm does not have an obvious change.

Therefore, the average algorithm for image short representation performs the worst when concerning its ability to preserve image similarities. The outcome of the weighted average algorithm seems more consistent than the average algorithm. The PCA algorithm is the best for image short representation. Even though there are disturbances along the curve of PCA due to information lost, it still preserve the image similarities best.

**Table 5.1**  
Maximum and minimum numbers of images per index.

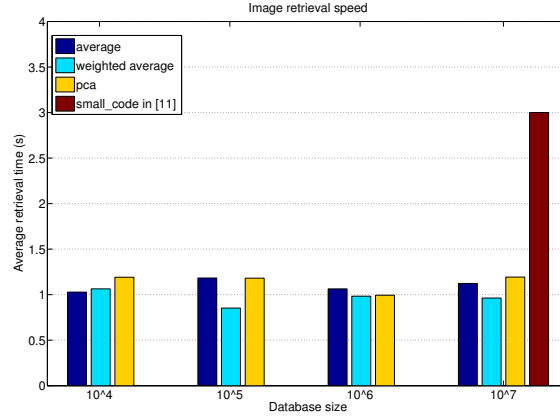
Bits	method	max	min	deviation
8	average	472,442	8,654	463,788
	w. average	365,720	11,497	354,223
	PCA	183,372	20,867	162,505
10	average	405,147	3,738	401,409
	w. average	288,072	4,245	283,827
	PCA	166,211	6,367	159,844
12	average	346,980	512	346,468
	w. average	208,014	951	207063
	PCA	155,231	1,498	153,733
14	average	353,909	361	353,548
	w. average	212,465	392	212,073
	PCA	109,558	435	109,123
16	average	399,331	33	399,298
	w. average	265,201	51	265,150
	PCA	95,788	97	95,691

In order to evaluate the SRs of a larger number of images, we also randomly select 100,000 images and compare their SRs pair-wise. If the distance between two SRs is less than 0.001, we consider them as the same SR. From these 100,000 images, we get 100,000 different SRs with any one of the three algorithms. The largest distance is 8.9512 from PCA algorithm and the smallest is 0.0022 from the average algorithm.

## 5.6.2 Image indexing

The goal is evaluate the performance of image indices. We index 11 million images in ImageNet as dataset and build the database. More than 30,000 Images in Caltech 256 are used as the test cases. When a test case is queried, the index of this image is calculated first. Then the vocabulary tree of this index will be loaded and the nearest neighbors of the test image will be retrieved from the vocabulary tree. We first evaluate the distance preserving of the binary index and then evaluate its performance in a real image retrieval application.

In the ImageNet database, the image short representations calculated with three different algorithms are mapped into binary indices of different lengths by spatial hashing. We set the index lengths to 8,10,12,14 and 16. Since there are a large number of images, the images should have a uniform distribution over all the indices if the short representation algorithm preserves the image distinctiveness well. Therefore, the longer the index, the



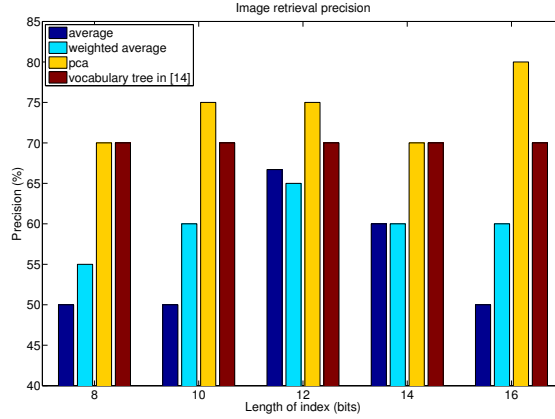
**Figure 5.5:** Image retrieval time evaluation. The average time from SR quantization to image retrieval are calculated. The PCA algorithm takes longer time than the other two.

smaller average number of images per index will have. One index with length of 16 is more likely to have a smaller number of images than that with length of 8. As a result, the vocabulary tree of a 16-bit index will take up less memories when looking for nearest neighbors in this index.

Table 5.1 shows the maximum and minimum number of images per index has for different bits and different SR algorithms. The largest deviations between the maximum and minimum values of the different length indices all come from the average algorithm. This proves again the bad performance of the average algorithm for preserving the image distinctiveness. Weighted average algorithm achieves better image distributions over the indices than the average algorithm. PCA has the smallest deviation between maximum and minimum numbers, which means it has the most uniform distribution of images.

The vocabulary trees are built for every index of images. According to [13], in order to achieve the best performance, we set the branch factor as 50, and the number of leaf nodes as the number of image features belonging to the index.

The image retrieval speed is evaluated based on different SR algorithms and different bits of indices. The dense SIFT features of 1,000 images are quantized into SRs first, then the indices are generated and the results are retrieved in the vocabulary tree. Every image has 700 SIFT features in average. The average time from SR quantization to image retrieval are calculated. The result of the small codes method in [11] is used for comparison. In [11], it takes  $6 \times 10^{-6}s$  to find 5 nearest neighbors of an image feature from 12.9 million features. In our case, their method will take about  $3s(6 \times 10^{-6} \times 700 \times 700)$  to compare an image with all the features in the database. Fig. 5.5 shows the results. In the results,



**Figure 5.6:** Image retrieval precision evaluation.

the retrieval for images with PCA SR algorithm takes about 1 seconds, with the average algorithm takes about 1.4 second and with weighted average algorithm takes less than 1 second. PCA spends longer time than the other two on calculating SRs, whereas Average spends most time on image searching. Even though it takes one second to retrieve images, the PCA still performs better than the small codes hashing methods considering the whole image is matched rather than a single feature. Another fact is that the running time is independent to the length of the indices.

From Caltech 256, 1000 images whose synsets also appear in ImageNet are selected as the input to the image retrieval engine. The ground truth is the synset name from Caltech 256. Given a query image, if the top 5 result candidates contain the ground truth, this image is considered as successfully retrieved. The precision is defined as the portion of successfully retrieved image in the total query images. Fig. 5.6 shows the results. The precision is consistent to the length of index, and the index generated by PCA outperforms the other two algorithms. It is not surprised that the average precision of PCA is higher than that reported in [13], because images in the same tree have the same index, images that interfere the performance of the vocabulary tree are excluded to other indices.

## 5.7 Conclusion

As the storage of digital images on the Internet grows significantly, bag-of-feature based object recognition and augmented reality applications on networked hand-held smart devices with cameras obtain more opportunities. But from a large-scale image database, it is also more time consuming to retrieve and classify images through pair-wise feature matching.

This paper presents an image indexing mechanism for efficient large-scale image retrieval and image recognition. It can represent an image as a binary index with as few as 16 even 8 bits. In order to obtain the binary index, the image features are quantized into a short representation, and then further mapped into a binary value with the novel spatial hashing. In the experiment, this method retrieves images faster than other feature-based indexing methods and the results promise a precision of 80%, which is higher than the standard vocabulary tree method.

# References

- [1] S. Helmer and D. Lowe, “Object class recognition with many local features,” *Computer Vision and Pattern Recognition Workshop, Conference on*, pp. 187–195, 2004.
- [2] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” *Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 518–529, Jan 1999. [Online]. Available: <http://portal.acm.org/citation.cfm?id=671516>
- [3] H. Jegou, M. Douze, C. Schmid, and P. Perez, “Aggregating local descriptors into a compact image representation,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3304 – 3311, 2010.
- [4] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 506–513, 2004.
- [5] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [6] H. Cevikalp, B. Triggs, F. Jurie, and R. Polikar, “Margin-based discriminant dimensionality reduction for visual recognition,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [7] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua, “Optimizing kd-trees for scalable visual descriptor indexing,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3392 – 3399, 2010.
- [8] C. Silpa-Anan and R. Hartley, “Optimised kd-trees for fast image descriptor matching,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [9] P. Jain, B. Kulis, and K. Grauman, “Fast image search for learned metrics,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [10] Y. Mu, J. Shen, and S. Yan, “Weakly-supervised hashing in kernel space,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3344 – 3351, 2010.



- [11] A. Torralba, R. Fergus, and Y. Weiss, “Small codes and large image databases for recognition,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [12] D. Lowe, “Object recognition from local scale-invariant features,” *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1150 – 1157, 1999.
- [13] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2161–2168, 2006.
- [14] T. Yeh, J. Lee, and T. Darrell, “Adaptive vocabulary forests br dynamic indexing and category learning,” *Computer Vision, IEEE International Conference on*, pp. 1 – 8, 2007.
- [15] J. Wang, S. Kumar, and S.-F. Chang, “Semi-supervised hashing for scalable image retrieval,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3424 – 3431, 2010.
- [16] C. Leistner, H. Grabner, and H. Bischof, “Semi-supervised boosting using visual similarity learning,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [17] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, Jan 2001. [Online]. Available: <http://www.springerlink.com/index/k62tg81w8352g71h.pdf>
- [18] Z. Wu, Q. Ke, M. Isard, and J. Sun, “Bundling features for large scale partial-duplicate web image search,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 25 – 32, 2009.
- [19] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 2169 – 2178, 2006.
- [20] I. Joliffe, “Principal component analysis,” *Springer-Verlag*, 1986.
- [21] K. Fukunaga and W. Koontz, “Application of the karhunen-loève expansion to feature selection and ordering,” *Computers, IEEE Transactions on*, vol. C-19, no. 4, pp. 311 – 318, 1970.
- [22] H. Murase and S. Nayar, “Detection of 3d objects in clustered scenes using hiearchical eigenspace,” *Pattern Recognition Letters*, vol. 18, no. 4, 1997.
- [23] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” *Technical Report 7694, Caltech*, Aug 2007.

- [24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 248 – 255, Jun 2009.
- [25] A. Vedaldi and B. Fulkerson, “Vlfeat: An open and portable library of computer vision algorithms,” *<http://www.vlfeat.org/>*, 2008.



## Chapter 6

# **DietVolume: Food Volume Estimation through Metric 3D Reconstruction on a Mobile Phone**

### **Abstract**

This paper presents DietVolume, a food volume estimation system for food intake assessment and obesity management. Obesity has been a severe health problem and the estimation and control of food intakes play an important role in obesity management. Food volumes could be estimated from food 3D structures reconstructed from multiple images. However, the metric scale of the structure is missing due to the projections. Techniques in this paper recover the metric scale of the 3D model and increase the 3D model accuracy through introducing inertial measurement units (IMU) i.e. gyroscope and accelerometer into the 3D reconstruction process. IMU could measure the metric scale distance it travels, which has the same scale with the 3D model. But the measurement from IMU is unreliable because of the bias. This paper uses Extended Kalman Filter (EKF) to reduce the bias of IMU and noises of the camera, meanwhile to estimate the scale of the 3D model. In the EKF, the measurements from IMU help decrease the noises from the camera, and the measurements from camera help estimate the bias of the IMU. The experiments show promising results in scale measurement and volume estimation.

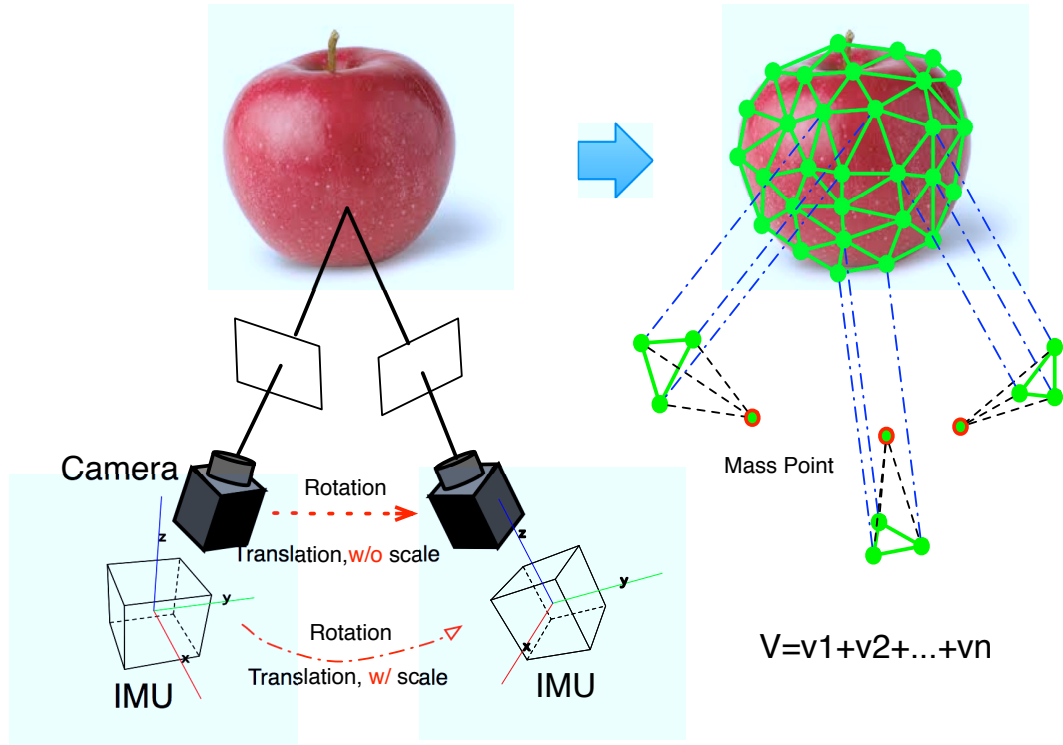
## 6.1 Introduction

In this paper, we develop a generalized markerless image-based metric 3D model reconstruction method on a camera phone aimed at food intake assessment applications. 3D model reconstruction has been an active research topic in computer vision for many years. With a metric scale 3D model, the volume of the object could be computed through geometric calculation in metric scale. Food volume estimation is a significant part of food intake estimation, which is important for obesity management and health related research. Obesity has been a severe public health challenge to the general population and social welfare[1, 2] in many developed countries. In the past three decades, the obesity rate in U.S. has increased significantly [3], resulting in serious consequences such as diabetes, stroke, heart disease even cancers. However, few people are aware of their food intakes and they are not willing to assess food intakes. The reason is the burdensome assessment methods and a lack of realtime feedback with these methods. Traditional food record or food diary methods require manual records of the food type and the portion of the food taken. The accuracy of this method is limited by the human rough estimation of the food portion. Image-based methods could automate food intake assessment. But in images, the dimension scale of the food is lost.

The scale information of an object in an image is directly relative to the depth of the object (the distance between the object and the camera), which is lost due to the projection from 3D positions to 2D images. Stereo vision could infer the depth of an object from image pairs. The stereo camera takes a pair of images from two slightly different perspectives. The distance between the cameras is known. The disparity between the image pair will be found through shifting and matching. Then the depth of the object will be estimated from the disparity. It is fast and easy for stereo vision to find the depth and it is widely used on robots and ground vehicles. But in many applications, such as augmented reality on camera phones, stereo cameras are not convenient or even not available.

With a single camera, the 3D structure of the environment could be perceived by moving the camera through it. In this process, structure from motion (SFM) techniques could be used to recover the 3D structure of the objects and the cameras from image sequences. However, the structure is up to an arbitrary scale and the volume of the structure is by no means to calculate. Methods appeared in structure from motion to infer the scale of the structure is to place an artificial reference with a known scale into the scene. But it limits its applications to place a marker before the 3D reconstruction.

Compared with stereo vision, SFM also uses multiple views of the object to reconstruct the 3D model. But the difference is that SFM could not estimate the metric distance between the view points directly from the images. From two views, the extrinsic parameters of the



**Figure 6.1:** Volume estimation of an apple. The camera could detect feature points and rebuild a 3D model for the apple. However, from the camera alone, the scale of the 3D model cannot be recovered. An IMU is attached to the camera, so that the motion of the camera, which has the same scale with the 3D model, could be recovered in metric scale. To estimate the volume of the 3D model, it is divided into individual tetrahedrons. The volume is the sum of those of the individual tetrahedrons, which could be calculated directly.

camera that encode the camera's orientation and position in the coordinate system could be calculated up to a scale [4]. The distance of the camera traveling between two views is directly related to the camera motion.

In robot odometry and egomotion, inertial sensors are widely used for detecting and tracking robot motion. The linear acceleration of the robot is measured by accelerometers, so that the velocity of the robot could be tracked. The angular velocity of the robot rotation is measured by the gyroscopes. The accelerometer measures the acceleration in a metric scale. Therefore the robot could be localized in a metric coordinate system, and the distance the robot has traveled could also be calculated in a metric scale.

Nowadays, with the development of high degree of integration, more and more camera phones are equipped with inertial sensors and cameras. This provides an opportunity to track the motion of the cell phone and camera in a metric scale. Compared with other

means that are also possible for the camera phone localization, such as GPS, cellular or wifi signal localization, inertial sensors are more accurate to detect small movements. But it is also reported that using inertial sensors alone will cause large deviations over time. This is caused by the drifts of inertial sensors and the integration over time enlarges this effect.

Concerning the drifts from inertial sensors, we treat the camera as the complement and correction to the inertial sensors. We present a metric 3D reconstruction with camera and inertial sensors, shown in Fig. 6.1, where the camera reconstructs the structure with SFM techniques and inertial sensors infer the metric scale of the structure. The camera motion estimated from the SFM will correct the measurements from inertial sensors. At the same time, the inertial sensors measurement could help correct the inaccuracy of the structure caused by the camera noises.

There are challenges in fusing these two types of sensors. First, an effective fusion method is needed to reduce sensor noises and drifts. Second, the non-linearity makes the fusion harder. Third, the inertial sensors and SFM operate on different rates. The inertial sensors divide the process into small slots and integrate these slots over time. While, SFM assumes large baselines, which indicate the distance between two consecutive frames should be large enough. This fact leads to a multi-rate sensor fusion.

An adaptive rate Extended Kalman Filter (EKF) is developed to fuse these sensors and estimate the scale of the 3D model. EKF is designed for non-linear process estimation and sensor fusion. The rate of the sensor fusion is adaptive and decided by the distance the camera traveled. Inertial sensors estimate the distance at a high rate. When the distance of the camera motion is far enough, SFM reconstructs the 3D model. Meanwhile the model and inertial sensor are fused. Otherwise, when the distance is not far enough, EKF does not fuse the results from SFM, but estimate the drifts of the inertial sensors only.

We present the related researches in the next section, followed by which, we formulate the image-based volume estimation problem. After the problem formulation, we solve the problem from 3D model reconstruction (section 6.4), scale estimation (section 6.5), volume calculation (section 6.6), and system implementation (section 6.7). Then the system is evaluated with field experiments (section 6.8). Finally, we conclude this paper in section 6.9.

## 6.2 Related work

The related work of this paper includes structure from motion, visual odometry, egomotion, and recently appeared image based food volume estimation.

### 6.2.1 Structure from motion

In computer vision, structure from motion (SFM) has been an active research topic for decades. It refers to the process of inferring 3D structures (up to a scale) from 2D observations from monocular vision or stereo vision[5]. The 3D structure includes the 3D model of the environment and the motion of the camera. Typical steps involved in SFM are extracting features, matching features from different views, calibrating the cameras, finding a dense representation of the scene, inferring geometric, textural and reflective properties of the scene. Challenges are noises from the camera and outliers of the matched features. There are several reviews from the earlier [6, 7], to the more recent literature [8]. Methods could be categorized into three kinds, optimization, fusing and filtering, and invariant-based approaches.

An optimization approach defines the “optimal” reconstruction as that minimizing an error function and searches for the optimal reconstruction by minimizing this error. The problem has been formulated and solved as a non-linear least square problem [9]. Bundle adjustment [10] is the problem of refining a visual reconstruction to produce jointly optimal structure and viewing parameter estimates. It could converge to the optimal solution when given a good starting point. Random sample consensus (RANSAC) [11, 12] has been a standard method for dealing with outliers arising from incorrect matched points. Compared with RANSAC, hypothesize-and-test framework (MLESAC) [13] optimizes the solution through maximizing the likelihood rather than just the number of inliers.

The fusing and filtering method computes a final multi-image reconstruction from intermediate reconstructions and estimates of the uncertainties in the images, rather than from image data directly. Typical method is Kalman filter (KF). Sequential Monte Carlo method has also been used [14]. Recently, inertial sensors have been imported to improve noise resistance, reduction of inherent ambiguities, and handling of mixed-domain sequences [15]. Extended Kalman filter (EKF) has been a standard fusion method for Inertial and visual sensors fusion [16]. Besides EKF, Unscented Kalman filter and EKF are compared in[17].

The invariant approach tries to find relations between images through deriving polynomial constraints on the image data by explicit algebra. Epipolar geometry [4, 18, 19] is an



example of this category.

One important information SFM loses is the absolute scale of the structures. It has been proposed to use accelerometers for scale estimation[20] simultaneously when reconstructing the structure through measuring the distance of the camera motion. However, without the orientation and rotation estimation, the scale estimation accuracy is still low. Our system is based on the SFM but the result is a structure with metric scales, including a metric 3D model of the environment, metric camera translations and rotations.

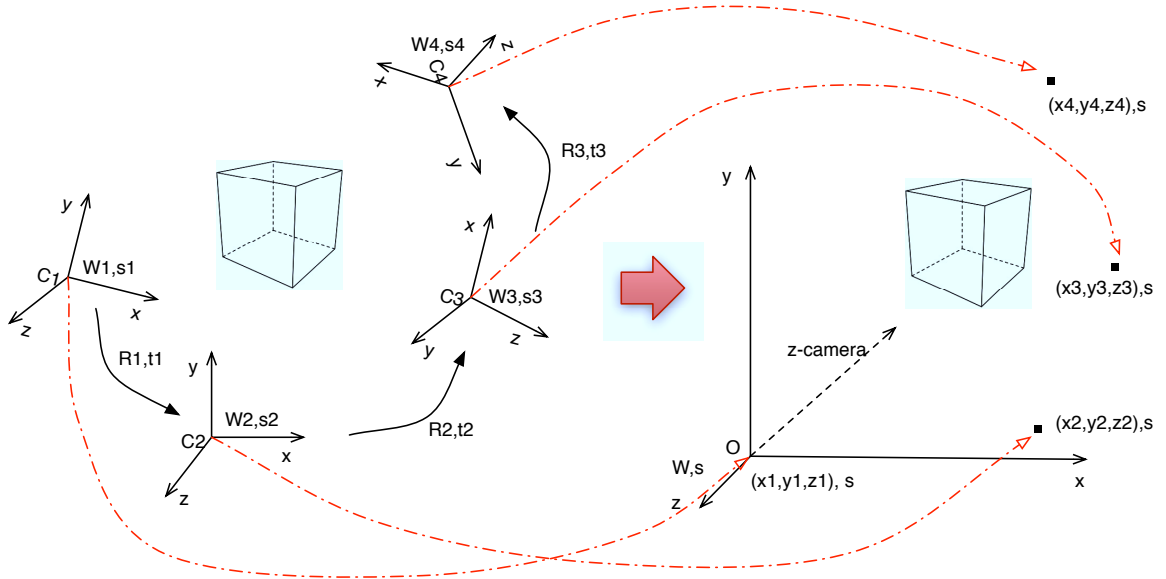
### **6.2.2 Visual odometry**

In robotics, the motion of the robots is also a research interest for many years. Visual odometry is the process to determine the location and orientation of the robots from corresponding images. It has been applied on planetary rovers[21–23], ground vehicles[24–29] and wearable systems[30]. The main devices are still but not limited to cameras. SFM techniques are usually used but the structure of the environment is not that important. The fast computation ability is a more critical factor. SFM techniques have been modified to increase the computation speed. For example, RANSAC has been modified to remove the outliers more efficiently[24, 27]. Non-SFM method has also been developed. For example, image displacement has been used[31]. Inertial sensors have also been used to facilitate cameras to decrease the growth of angular errors[26, 30].

Different from SFM, the absolute scale is critical in odometry, especially in ground vehicle navigation and indoor navigation. The scale is usually inferred from the absolute position of the robot through localization sensors such as GPS. When GPS is not available, i.e. indoor or on the Mars, motor drive speeds and stereo visions are utilized to provide scale estimation. The inertial sensors play a role as a complement to the vision system for corrections and noise reductions. In our system, the inertial sensors act an significant role for simultaneously scale estimation.

### **6.2.3 Egomotion**

In computer vision, camera egomotion is the tracking of the camera motion from the image sequences, without other localization devices, such as GPS. SFM techniques could be used directly for egomotion except environment model reconstruction. Epipolar geometry has been used for inferring the camera parameters between two views[32]. The egomotion could also be calculated with a rough environment model [33]. Egomotion also benefits



**Figure 6.2:** 3D reconstruction and scale recovery from videos. When a camera moves around a 3D object (in the left of the figure), part of the 3D model could be reconstructed with every pair of consecutive images. The coordinate systems and scales of the reconstructed model are different from each pair of images. We convert these coordinate systems into a unified system (in the right part), where the scale factor is constant and traceable.

from cameras with large field of views, such as omnidirectional cameras [34].

Inertial sensors have also been introduced into egomotion calculation. Through fusing visual and inertial sensors, the noises from cameras and drifts from inertial sensors could be reduced. EKF is usually used for this non-linear sensor fusion problem[35, 36]. Other types of sensor fusion techniques have also been explored, such as Unscented Kalman filter[37], Marginalized particle filter[38] and Expectation Maximization[39].

Similar with SFM, the absolute scale information is not so important in egomotion, since it only concerns the camera's relative movement in the structure. Inertial sensors have been used still as a complement to the vision system for corrections and noise reductions.

#### 6.2.4 Image based food volume estimation

Food volume estimation is an important aspect in food intake assessment that is our target application. There have been some related researches in image-based food volume estimation. The 3D model of the food has not been explored. Typical methods approximate the

food volume as the surface area of the food in the image. Methods to infer the scale of the food could be formulated as the calibration card method, which estimates the scale and the surface area of the foods through detecting the calibration card [40–42] or a physical reference [43] with a known scale. Its applications are limited since the mandatory appearance of the calibration card or reference. Compared with these methods, we provide a more accurate markerless food volume estimation method. The 3D model of the food items will be build with SFM techniques at the same time the scale of the 3D model is estimated through fusing inertial sensors.

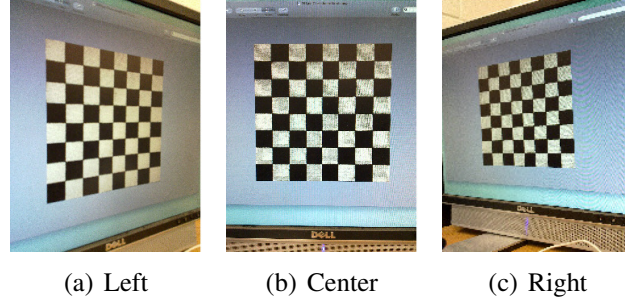
## 6.3 Problem Formulation

The volume of an object could be estimated from its 3D mesh model with known geometry properties. The geometry property of the 3D model could be obtained from the coordinates of the vertices. The scaleless coordinates could be reconstructed from images. The scale of the coordinates could be estimated from the IMU. The mesh model of an irregular shape could be divided into a number of small regular shapes, whose volume could be calculated directly from the coordinates of the vertices. Therefore, in order to estimate the volume of food items, we have three problems to solve, 3D model reconstruction, scale estimation, and volume calculation.

### 6.3.1 3D model reconstruction

The 3D model without a scale of the object could be reconstructed with two images. As shown in Fig. 6.2, when a camera moves around a 3D object (in the left of the figure), part of the 3D model could be reconstructed with every pair of consecutive images. In the first pair of images from camera C1 and C2, the 3D model of the object is reconstructed in the coordinate system  $W_1$ , with scale factor  $s_1$ , which are defined by the camera C1. In the second pair of images from camera C2 and C3, the 3D model of the object is reconstructed in the coordinate system  $W_2$ , with scale factor  $s_2$ , which are defined by the camera C2.  $W_2$  could be transformed to  $W_1$  with the rotation  $R_1$  and translation  $t_1$  from camera C1 to C2. The scale could also be unified through calculating the distances between the same points in the  $W_1$  and  $W_2$  respectively. We convert the 3D model reconstructed by each pair of images into a unified system with a constant scale factor, so that the scale would be traceable.

Given a series of images taken around a food item, the problem is to recover the 3D structure of the food item from these images. Consider an object  $\mathbf{O}$  in the 3D space with a 3D mesh model  $\mathbf{G}$  consisting of  $N$  vertices  $\mathbf{V} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ . The  $i$ th vertex ( $0 < i \leq N$ )



**Figure 6.3:** Intrinsic Parameters Calibration. Three different perspective chess-board images will give enough information to calibrate the camera, which is easy and convenient for the users.

has a scaled coordinate in the world coordinate system,  $\mathbf{X}_i = \{x_i, y_i, z_i, \lambda_i\}$ . The coordinates of all the vertices  $\mathbf{V}$  are necessary to calculate the volume of object  $\mathbf{O}$ .

In order to calculate the coordinates of  $\mathbf{V}$ , the object  $\mathbf{O}$  is sampled by a video camera from a series of viewpoints. At time  $k$ , the location in the world coordinate system of the camera is  $p_k$ . The parameter of the camera is  $P$ . The coordinate of  $\mathbf{X}_i$  in the image coordinate system is  $\mathbf{x}_i = P\mathbf{X}_i = \{x_i, y_i, t_i\}$ . At time  $k+1$ , the coordinate of the camera is  $p_{k+1}$ , and the parameter matrix is  $P'$ . The coordinate of  $\mathbf{X}_i$  in the image is  $\mathbf{x}'_i = P'\mathbf{X}_i = \{x'_i, y'_i, t'_i\}$ . Due to the camera projection, one dimension is lost and the scale of the 3D world is lost.

Through back projection of the vertices in the images, the scaleless 3D coordinates  $\mathbf{X}'$  of the vertices could be deduced,  $\mathbf{X}'_i = \{x'_i, y'_i, z'_i, \lambda'_i\}$ .  $\mathbf{X}'_i$  of a projected vertex is up to a scale factor  $s$  to the original coordinates  $\mathbf{X}$ , that is  $s \frac{x'_i}{\lambda'_i} = \frac{x_i}{\lambda_i}, s \frac{y'_i}{\lambda'_i} = \frac{y_i}{\lambda_i}, s \frac{z'_i}{\lambda'_i} = \frac{z_i}{\lambda_i}$ .

Therefore, the problem is, given an image sequence, and the correspondent points  $x$  on the images, to estimation the camera parameters  $P$ , which includes  $R$  and  $t$ . Then with  $P$ , the 3D coordinates  $X$  could be reconstructed from  $x$ .

### 6.3.2 Scale estimation

When a 3D object is projected to 2D image space, one dimension is lost, together with the scale. The scale of the object's projection is up to a scale factor to the metric scale of the object in the 3D space. The scale factor is directly related to the distance from the camera to the object and the camera coordinate system.

The scale factor maps the original 3D space to the reconstructed 3D space. Any distance in the new space is up to the same scale to the corresponding distance in the original space, such as the distance between two vertices, the distance from a camera to a vertex, and the distance between two cameras. Therefore, if any one of the distances could be measured and the corresponding distance in the reconstructed space is calculated, the scale factor would be estimated. The distance between two vertices and that between the camera and a vertex cannot be easily obtained without user defined references or range finders. Whereas the distance between cameras could be inferred from the motion sensors.

The inertial measurement unit, including accelerometers, gyroscopes and magnetometers measures the camera scaled motion. However, the distance calculated directly from integrating the acceleration over time is not accurate due to the drifts of the inertial sensors. We develop an EKF to fuse the visual sensor with the inertial sensors. The reconstructed model and the reconstructed camera position calculated from the visual camera are corrected with the measurements from IMU. At the same time, the IMU's drifts are estimated and reduced with the measurements from the camera. The scale factor will be estimated recursively from an initial value, and the estimation will be the value it converges to.

The IMU measures the camera scaled motion in the IMU coordinate system. At time  $k$ , the accelerometer measures the camera's scaled acceleration  $a_{sc,k}$  in the IMU coordinate system. The accelerometer has a bias  $b_{a,k}$ . The gyroscope measures the camera's angular velocity  $\omega_{sc,k}$  with a bias  $b_{\omega,k}$ . The magnetometer keeps the heading of the IMU in the 3D world. The magnetometer together with the gyroscope would track the orientation of the IMU and determine a transformation matrix  $M_{es}$  that maps the IMU coordinate system to the world coordinate system. The scaled distance  $d$  that the camera traveled in the world coordinate system could be calculated through integrating the acceleration and its orientation over time.

The problem is to find a way to compute the scale factor  $s$  with the help of IMU, estimate the coordinates of  $\mathbf{X}$ , and calculate the volume of  $\mathbf{G}$ . The scale factor  $s$  could be computed from the scaleless camera location  $p_k, p_{k+1}$  calculated with SFM techniques and scaled location  $p_k^m, p_{k+1}^m$  calculated from IMU.

$$\begin{aligned}
p_{k+1} &= s \times p^{m_{k+1}} \\
&= s \times (p_k^m + v_{ec,k} \times T + \frac{T^2}{2} a_{ec,k}) \\
&= s \times p_k^m + s \times v_{ec,k} \times T + s \times \frac{T^2}{2} a_{ec,k} \\
&= p_k + s \times v_{ec,k} \times T + s \times \frac{T^2}{2} a_{ec,k}
\end{aligned} \tag{6.1}$$

where  $v_{ec,k}$  is the velocity of the camera at time  $k$  in the world coordinate and  $T$  is the time

span.

### 6.3.3 Volume calculation

Given a 3D model  $\mathbf{G}$  with a known scale  $s$ , the problem is how to calculate its volume  $V$  from the coordinates of  $N$  vertices  $\mathbf{V} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ .

## 6.4 3D model reconstruction

The 3D model reconstruction process could be divided into three steps, feature points tracking, camera calibration and back projection. The feature points in the image are detected with Harris corner detector[44]. Then the corners are tracked with Lucas-Kanade-Tomasi (LKT) feature tracker [45].

A user friendly and general method to calibrate a camera's intrinsic parameters is developed for different cell phones. Many camera calibration methods have been proposed in the computer vision literature. The flexible camera calibration method[46] is well-suited for our requirements. It does not require any professional knowledge other than the user to shoot a planar pattern from two or more perspectives. We provide a chessboard pattern online, which is not only convenient for the users to access, but a known standard pattern to calibrate different types of cameras.

The parameters of a camera can be represented as

$$P = A[R \ T] \quad (6.2)$$

where

$$A = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6.3)$$

is the intrinsic parameter matrix,  $R$  and  $T$  are extrinsic rotation and translation parameters. In the intrinsic matrix  $A$ ,  $(u_0, v_0)$  is the coordinate of the principal point,  $\alpha$  and  $\beta$  are the scale factors in the  $u$  and  $v$  axes,  $c$  is the parameter represent the skewness of the axes  $u$  and  $v$ . The intrinsic matrix  $A$  is calibrated through finding correspondences between multiple views and solving the constraint equations established by the correspondences[46]. The users take three pictures of the chessboard under different orientations by moving the mobile phone as shown in Fig. 6.3. In the images, the inner corners of the chessboard will

be detected. Once calibrated, it will be stored on the hard drive and no longer needs to calculate again.

Unlike the static intrinsic parameters  $A$ , the extrinsic parameters  $R$  and  $T$  are always changing when taking pictures. Consequently, they need to be estimated every time right after the food pictures have been taken. We assume these three images are taken by three different cameras at the same time and group them into two pairs. The image in both pairs defines the world coordinate and this camera is defined as the reference camera. The other two cameras are calibrated pair by pair.

In order to make the users unconcerned with the calibration process, epipolar geometry is used because it only requires correspondences in the image to calibrate the extrinsic parameters. The correspondences between a pair of images are already extracted in the feature matching process. In a pair of images, the camera matrix of the reference image can be chosen as

$$P = A[I \ 0] \quad (6.4)$$

where  $I$  is a  $3 \times 3$  unit matrix. By doing this, the world coordinate system is decided. After the mobile phone is moved to take another picture, the new camera matrix related to the world coordinate system is decided as

$$P' = A[R \ T]. \quad (6.5)$$

The extrinsic parameters  $R$  and  $T$  can be estimated with intrinsic matrix and correspondences between these two views. In epipolar geometry, the essential matrix  $E$  encapsulates the projection relationship between two intrinsically calibrated cameras. It has the property

$$pEp' = 0 \quad (6.6)$$

where  $p$  and  $p'$  are correspondent points in two views. Hence,  $E$  can be estimated with the correspondent points. According to its definition

$$E = [T]_{\times} R \quad (6.7)$$

where  $[T]_{\times}$  is the skew-symmetric matrix of  $T$ , the extrinsic parameters  $R$  and  $T$  can be estimated by singular value decomposition[47].

After obtaining the camera matrix  $P$  and  $P'$ , the 3D coordinates of the corresponding points could be calculated through linear triangulation.

Since equation (6.6) still holds if it is multiplied by an arbitrary constant  $\lambda$ ,

$$\lambda \times pEp' = 0 \quad (6.8)$$

the extracted  $E$  and  $T$  will have an arbitrary scale. More over, the reconstructed  $P$ ,  $P'$  and the 3D point coordinates will also up to an arbitrary scale.

## 6.5 Scale factor estimation

The absolute scale of the structure is calculated through the fusion of a camera and inertial sensors, including an accelerometer, a gyroscope, and a magnetometer.

Since inertial sensors are introduced and they will be fused with the camera, the coordinate system between them should be defined first. There are three 3D coordinate systems connected to our vision-inertial system. Fig.6.4 illustrates how these coordinate systems are related.  $c$ ,  $s$  and  $e$  denote the Camera, Sensor and Earth coordinate frames respectively. Furthermore, the relative transformations between each two frames are represented by unit quaternion.

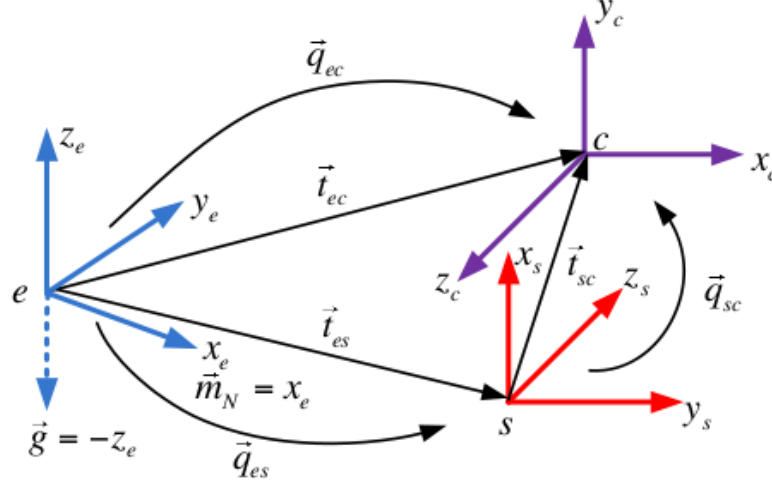
- † Earth frame ( $e$ ) It is fixed to earth with  $x_e$  axis pointing to local magnetic north and the  $z_e$  axis pointing opposite gravity.
- † Camera frame ( $c$ ) It is attached to the moving camera. The origin of the camera frame is located in the optical center of the camera, with the  $z_c$  axis pointing along the optical axis. The camera pose which includes the rotation  $\vec{q}_{ec}$  and the translation  $\vec{t}_{ec}$  is estimated relative to the earth frame  $e$ .
- † Sensor frame ( $s$ ) It is attached to the moving inertial sensors. The sensor readings directly obtained from the IMU are with respect to the sensor frame  $s$ . The sensor pose which includes the rotation  $\vec{q}_{es}$  and the translation  $\vec{t}_{es}$  is estimated relative to the earth frame  $e$ .

In the earth coordinate system, an EKF is formulated for the sensor fusion. The state vector  $x_k$  consists of the position of the camera  $p_{ec,k}$ , its velocity  $v_{ec,k}$ , acceleration  $a_{ec,k}$ , angular velocity  $\omega_{e,k}$ , the scale factor  $s_k$ , camera's orientation  $q_{e,k}$ , the accelerator bias  $b_{a,k}$ , and the gyroscope bias  $b_{\omega,k}$ .

$$x_k = \{p_{ec,k}, v_{ec,k}, a_{ec,k}, \omega_{e,k}, s_k, q_{e,k}, b_{a,k}, b_{\omega,k}\}. \quad (6.9)$$

The camera position  $p_{ec,k}$  is a four-element vector that encodes the camera's coordinates in





**Figure 6.4:** Three coordinate systems associated with the alignment procedure and the respective transformation based on unit quaternion

the reconstructed earth coordinate system.  $p_{ec,k}$  is reconstructed from the cameras. Therefore it is up to a scale to the metric world coordinate. The velocity  $v_{ec,k}$  is the camera's velocity in the metric world coordinate system. The angular velocity of the camera has three components,  $\omega_{e,k} = \{\omega_{ex,k}, \omega_{ey,k}, \omega_{ez,k}\}$ . The scale factor between the camera reconstructed coordinate and the metric world coordinate system is represented by  $s_k$ , which is estimated every time step. The camera orientation  $q_{e,k}$  is represented in unit quaternion  $q_{e,k} = \{q_{e0,k}, q_{ex,k}, q_{ey,k}, q_{ez,k}\}$ . The biases of accelerometer and gyroscope are also estimated and tracked in the state vector.

The process to be estimated is the camera motion with acceleration and rotation inputs together with noises,

$$x_k = f(x_{k-1}, u_{k-1}, n_{k-1}) \quad (6.10)$$

with a measurement  $z_k$  of camera position, acceleration and angular velocity that is

$$z_k = hx_k + m_k \quad (6.11)$$

We assume the camera has a random walk motion. Therefore, the control input  $u_{k-1}$  is empty. The random variables  $n_k$  and  $m_k$  represent the process and measurement noises respectively.

$$p(n) \sim N(0, Q)$$

$$p(m) \sim N(0, R)$$

We assume the camera has a uniformly accelerated linear translation at time  $k - 1$  and the time span between  $k - 1$  and  $k$  is  $T$ . The translation of the camera can be modeled by an equation set. The camera orientation is modeled by a uniformly rotation with angular velocity  $\omega_{e,k-1}$ . The rotation is represented by quaternion. We use a random walk model to estimate  $s_k$  and the biases  $b_{a,k}, b_{\omega,k}$  based on the value and a white noise at time  $k - 1$ . The random noises are  $n_{k-1}^s, n_{k-1}^a$ , and  $n_{k-1}^\omega$  for  $s_{k-1}, b_{a,k-1}$ , and  $b_{\omega,k-1}$  respectively. Therefore, the dynamic model of the state is defined as

$$\begin{aligned}
p_{ec,k} &= p_{ec,k-1} + \frac{T}{s_{k-1}} v_{ec,k-1} + \frac{T^2}{2s_{k-1}} a_{ec,k-1} \\
v_{ec,k} &= v_{ec,k-1} + T a_{ec,k-1} \\
a_{ec,k} &= a_{ec,k-1} + n_{k-1}^a \\
\omega_{e,k} &= \omega_{e,k-1} + n_{k-1}^\omega \\
s_k &= s_{k-1} + n_{k-1}^s \\
q_{e,k} &= \exp(\omega_{e,k-1} T) \otimes q_{e,k-1} \\
b_{a,k} &= b_{a,k-1} + n_{k-1}^{ab} \\
b_{\omega,k} &= b_{\omega,k-1} + n_{k-1}^{\omega b}
\end{aligned}$$

where  $\otimes$  is defined as the quaternion multiplication, and

$$\exp(\omega T) = \left[ \cos \frac{\|\omega\|T}{2}, \sin \frac{\|\omega\|T}{2} \frac{\omega^T}{\|\omega\|} \right]^T \quad (6.12)$$

$$\|\omega\| = \sqrt{(\omega_x)^2 + (\omega_y)^2 + (\omega_z)^2} \quad (6.13)$$

At time  $k = 1$ , the scale factor is initialized as the direct computation from the distance of IMU traveled and camera reconstructed. It does not concern the noises and drifts in the sensors therefore it is not accurate. Then at each time  $k$ , the scale factor will be updated to a more accurate value together with the state vector  $x_k$ . At last, the scale factor  $s_k$  will converge to  $\hat{s}$ . The state vector  $x_k$  is updated following the time update rule that is modeled by the dynamic model. The time update equations are

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (6.14)$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (6.15)$$

where  $A$  is the Jacobian matrix of partial derivatives of  $f$  with respect to  $x$ , and  $W$  is the

Jacobian matrix of partial derivatives of  $f$  with respect to  $n$ ,

$$A_k = \frac{\partial f}{\partial x}(\hat{x}_{k-1}, 0) \quad (6.16)$$

$$W_k = \frac{\partial f}{\partial n}(\hat{x}_{k-1}, 0) \quad (6.17)$$

The measurement update model includes the measurements of the camera position  $p_{ec,k}^m$ , acceleration  $a_{ec,k}$ , angular velocity  $\omega_{e,k}$  and scale factor  $s_k^m$ .  $p_{ec,k}^m$  is reconstructed from two images in the reconstructed coordinate system. Measured scale factor  $s_k^m$  is measured by  $s_k^m = \frac{D_{s,k}}{D_{v,k}}$ , where  $D_{s,k}$  and  $D_{v,k}$  are distance from the inertial sensors and cameras respectively.  $a_{ec,k}$  and  $\omega_{e,k}$  are measured from the inertial sensor,

$$a_{ec,k} = M_{es}(a_{s,k}^m - b_{a,k} - n_k) - g \quad (6.18)$$

$$\omega_{e,k} = \omega_k^m - b_{\omega,k} - n_k \quad (6.19)$$

where  $a_{s,k}^m$  is the accelerometer measurements in the IMU coordinate system at time  $k$ ,  $M_{es,k}$  is the transformation matrix from the sensor coordinate to the earth coordinate at time  $k$ . Similarly, the control input of the angular velocity is from the reading of gyroscope. The transformation matrix  $M_{es,k}$  changes with each time step. It is the matrix representation of the quaternion of the orientation of the accelerometer.

The measurement update equations are

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (6.20)$$

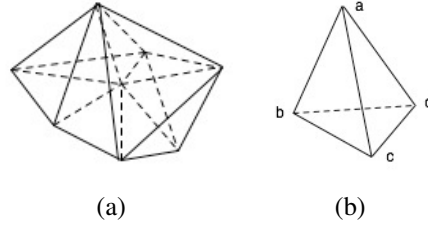
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \quad (6.21)$$

$$P_k = (I - K_k H_k) P_k^- \quad (6.22)$$

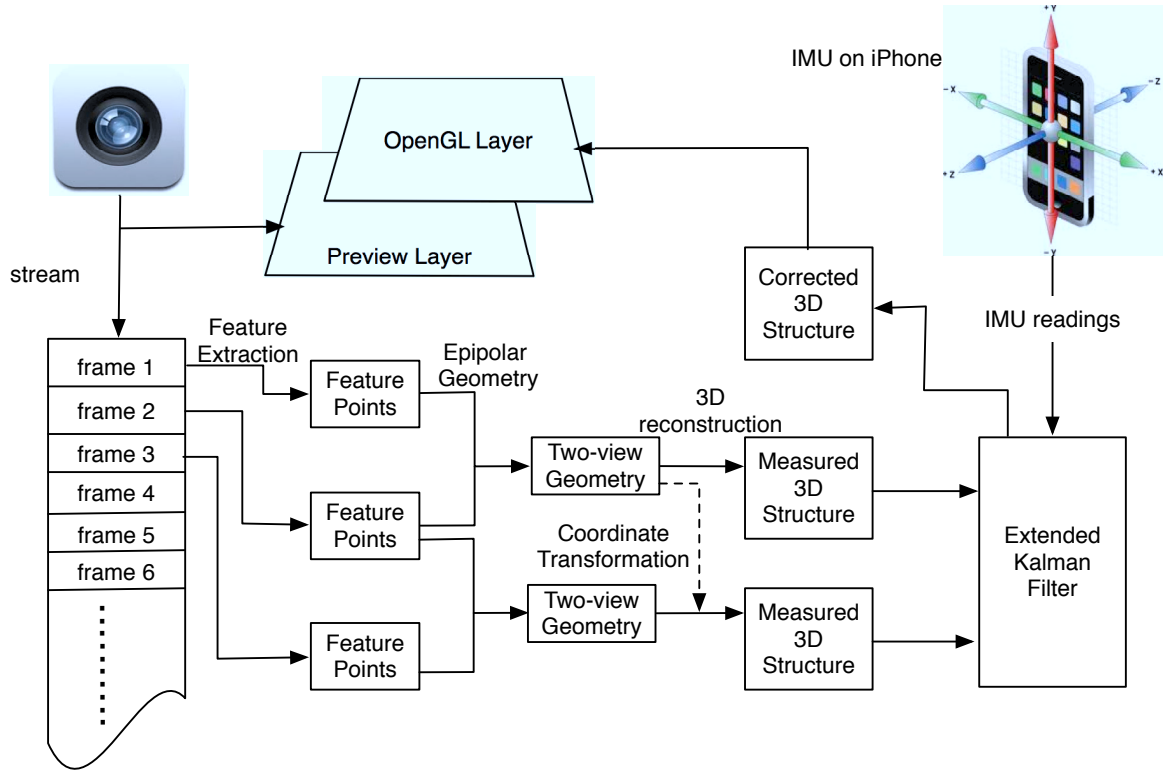
where  $H$  is the Jacobian matrix of partial derivatives of  $h$  with respect to  $x$ , and  $V$  is the Jacobian matrix of partial derivatives of  $h$  with respect to  $m$ ,

$$H_k = \frac{\partial h}{\partial x}(\hat{x}_k^-, 0) \quad (6.23)$$

$$V_k = \frac{\partial h}{\partial m}(\hat{x}_k^-, 0) \quad (6.24)$$



**Figure 6.5:** Tetrahedrons of an arbitrary shaped food item. 6.5(a) shows all the tetrahedrons, the point in the center is the estimated mass point. 6.5(b) shows a single tetrahedron with point coordinate  $a, b, c$  and  $d$ . The volume of this tetrahedron is calculated with equation (6.25).



**Figure 6.6:** DietVolume architecture.

## 6.6 Volume Calculation

We assume the food shapes are convex and the 3D model could cover the surface of the food. Then, the volume of an arbitrary shape model is calculated by dividing the whole model into small elements. Based on the idea of finite element analysis[48], a 3D object can be divided into a finite number of arbitrary shaped parts. A meal is divided into several

food items based on the classification information and a food item is divided up further. The mass point of the item could be used as the center for dividing the food item and it is estimated by averaging coordinates of all the points. It is connected to each 3D point, forming a group of tetrahedrons, as Fig. 6.5 shows. The volume of the food item is the sum of the volume of every single tetrahedron. With the coordinates of four points of the tetrahedron, the volume can be calculated with a dot product and a cross product as

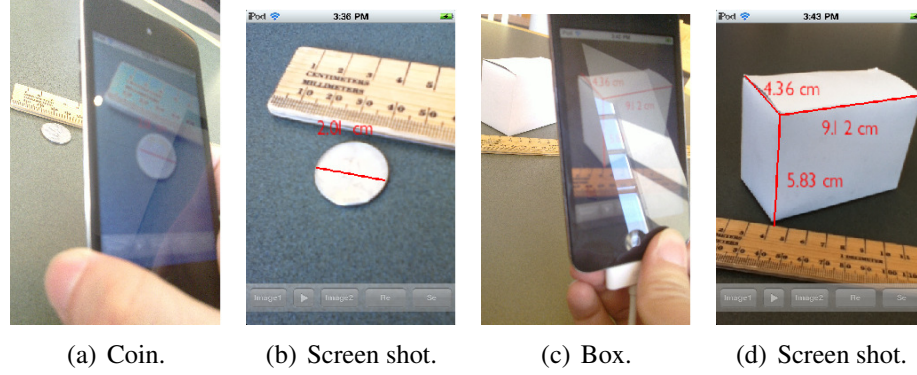
$$V_i = \frac{(a-d) \cdot ((b-d) \times (c-d))}{6} \quad (6.25)$$

where  $a, b, c, d$  are the coordinate vectors of the points.

The accuracy of the volume calculation is directly affected by the vertices of the 3D model. The vertices are from the feature points in the images. On smooth convex surfaces, where only few feature points would be detected, the sparse features could not cover the contour precisely. The result would be less than the real volume. On the contrary, on rough surfaces, the 3D model would cover the outermost layer of even beyond the food item. Consequently, the result will be larger than the real value. Generally, the more vertices the model has, and the more details the model covers, the more accurate the result will be.

## 6.7 Implementation and Visualization

We design and implement an iPhone virtual reality application, DietVolume, to evaluate the scale estimation with the EKF. As shown in Fig. 6.6, DietVolume has two view layers, a preview layer and an OpenGL rendering layer. The preview layer shows the camera contents directly to the user as the screen background. The OpenGL layer visualizes the reconstructed 3D structure of the scene on top of the preview layer. The camera stream is processed frame by frame. The feature points of every frame will be extracted by the feature extractor first. The feature points from two consecutive frames will be matched to find point correspondences, with which, the two-view geometry could be calculated through epipolar geometry. The resulting two-view geometry includes the camera parameters and correspondent points. Therefore, the 3D locations of the points could be reconstructed. Note, the coordinate systems between consecutive two-view geometries are different, in terms of original point, direction and scale. We convert the second coordinate system to the first one through camera parameters and feature points in common. The camera position of the 3D structure under unified coordinate system will be used as the measurement to the EKF. At the same time, the camera motion measured from the IMU will be used as another measurement. With the measurements, the EKF corrects its estimation of the 3D structure and scale. The result 3D structure will be rendered on the OpenGL layer and the volume of the reconstructed object will be calculated.



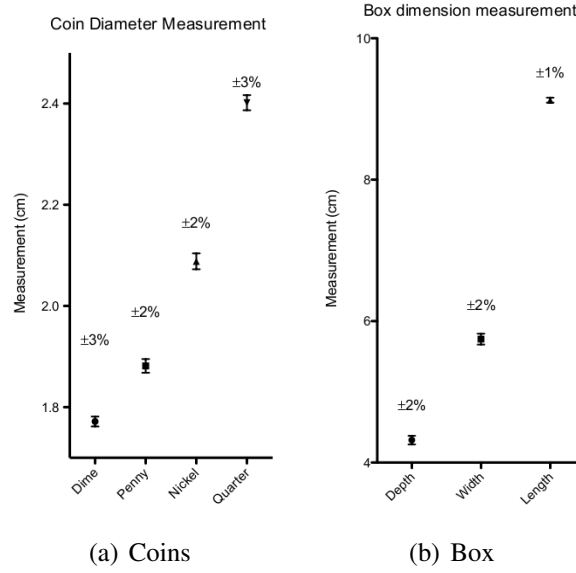
**Figure 6.7:** 3D object dimension measurement on iPhone.

Fig. 6.7 shows the visualization results of DietVolume. Two test cases are shown and the screen shots are presented. One test case is to measure the diameter of a coin. Two feature points are selected to constitute the diameter of the coin approximately. The other test case is to measure the dimension of a box, and to estimate its volume from the dimensions. The preview layer shows the object, i.e. a coin and a box, which comes from the unprocessed streams from the camera. The OpenGL layer renders selected feature points, i.e. red dots on the screen, and the distance between them.

## 6.8 Experiment

We are interested in two properties of DietVolume, the accuracy of the system and the time it needs to calculate the volume. The accuracy of the volume reconstruction will be evaluated through two steps, first object dimension estimation and volume estimation of 3D objects. The time the system needs for calculation is evaluated through tracking the scale factor in the image sequences and counting the number of frames the scale factor needs to converge. The results show promising reconstruction accuracy and a short converging time that the scale factor will converge in tens of frames.

The scale estimation accuracy is evaluated first with length measurements. In the experiments, we use the camera phone to reconstruct the 3D model of an object and measure the length between two vertices of the model. The typical scenarios include dimension measurement of non-food items such as coins and boxes. A common property of these types of objects is relative clear corners and edges to detect. Therefore, the scale estimation algorithm could be evaluated with least affections from 3D model reconstruction. For every object, we repeat the estimation for five times with different perspectives and motion patterns. Fig. 6.8 shows the results together. From the results, we can see the algorithm

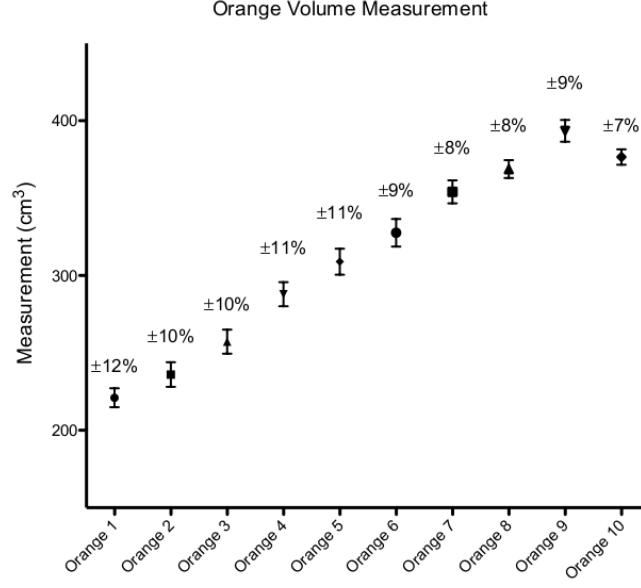


**Figure 6.8:** Coins diameter and box dimension measurements.

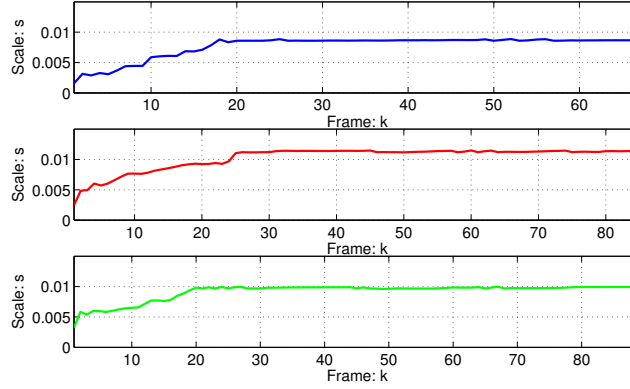
has a high accuracy over 98% and it is stable and consistent with the various lengths.

Then we evaluate the algorithm for real food volume estimation. The volumes of ten oranges are estimated with the algorithm. The true volume are measured through water displacement. We still perform five estimations for each orange. Fig. 6.9 shows the results. The accuracy is about 88%. The errors come from two sources, the scale estimation and the model reconstruction. The scale estimation of linear length is still of the same accuracy of 98%. But when calculating the volume, the error accumulates to cubic. Another factor is the feature point based 3D model reconstruction. The 3D model is reconstructed by the feature points, and its volume is defined by the position of the feature points. Therefore the volume of the 3D model could be inconsistent with that of the object if the feature points cover only part of the object or they include spaces outside the objects.

Another important factor of the system is the performance of the EKF. In the experiment, we track the outcome of the EKF and record the number of frames that the EKF needs to converge the scale factor. Fig. 6.10 shows the results of three different experiments. It shows the scale factors of different experiments are different and they will be converged in 25 frames.



**Figure 6.9:** Orange volume estimation.



**Figure 6.10:** Number of frames the scale factor needs to converge.

## 6.9 Conclusion

This paper presents a metric scale 3D model reconstruction method implemented on a cell phone for food volume estimation. The applications of this method are not limited to food volume estimation, but also include arbitrary object dimension estimation. The advantage is scale estimation and accuracy improvement by means of integrating camera and inertial sensors. The inertial sensor detects camera movement in metric scale but with drifts and noises. The camera recovers the motion in arbitrary scales. Through sensor fusion, the



drift from inertial sensors could be estimated and reduced; the noise from the camera could also be decreased. The experiment shows promising results that the scale factor could be estimated in tens of frames and the volume estimation is of about 90% accuracy.

# References

- [1] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [2] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [3] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [4] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision, 2nd ed,” *Book, Cambridge University Press, 2004*, Jan 2004.
- [5] T. Jebara, A. Azarbayejani, and A. Pentland, “3d structure from 2d motion,” *IEEE Signal Processing Magazine*, vol. 16, no. 3, pp. 66–84, Jan 1999.
- [6] C. Jerian and R. Jain, “Structure from motion-a critical analysis of methods,” *Systems, Man and Cybernetics, IEEE Transactions on DOI - 10.1109/21.97478*, vol. 21, no. 3, pp. 572–588, 1991.
- [7] T. Huang and A. Netravali, “Motion and structure from feature correspondences: a review,” *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, 1994.
- [8] J. Oliensis, “A critique of structure-from-motion algorithms,” *Computer Vision and Image Understanding*, vol. 80, no. 2, Nov 2000.
- [9] R. Szeliski and S. Kang, “Recovering 3d shape and motion from image streams using nonlinear least squares,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 752–753, 1993.
- [10] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment—a modern synthesis,” *Vision algorithms: theory and practice*, pp. 153–177, 2000.
- [11] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, Jan 1981.

- [12] D. Nistér, “Preemptive ransac for live structure and motion estimation,” *Machine Vision and Applications*, vol. 16, no. 5, Dec 2005.
- [13] P. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, Jan 2000.
- [14] G. Qian and R. Chellappa, “Structure from motion using sequential monte carlo methods,” *International Journal of Computer Vision*, vol. 59, no. 1, Aug 2004.
- [15] G. Qian, R. Chellappa, and Q. Zheng, “Robust structure from motion estimation using inertial data,” *J Opt Soc Am A Opt Image Sci Vis*, vol. 18, no. 12, pp. 2982–97, Dec 2001.
- [16] P. Germeiner, P. Einramhof, and M. Vincze, “Simultaneous motion and structure estimation by fusion of inertial and vision data,” *The International Journal of Robotics Research*, vol. 26, pp. 591–605, Jan 2007.
- [17] G. Bleser, “Towards visual-inertial slam for mobile augmented reality,” *PhD Dissertation, Technical University Kaiserslautern, Dr. Hut Verlag*, Mar 2009.
- [18] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–195, Apr 1998.
- [19] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” *Artificial Intelligence*, vol. 78, pp. 87–119, 1995.
- [20] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of IMU and vision for absolute scale estimation in monocular slam,” *International Conference on Unmanned Aerial Vehicles, Dubai*, 2010.
- [21] P. Corke, D. Strelow, and S. Singh, “Omnidirectional visual odometry for a planetary rover,” *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, vol. 4, pp. 4007–4012 vol.4, 2004.
- [22] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers: Field reports,” *J. Field Robotics*, vol. 24, no. 3, Mar 2007.
- [23] Y. Cheng, M. Maimone, and L. Matthies, “Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 54–62, 2006.
- [24] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, “Real-time monocular visual odometry for on-road vehicles with 1-point ransac,” *Robotics and Automation, IEEE International Conference on*, pp. 4293–4299, 2009.

- [25] A. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," *Robotics and Automation, IEEE International Conference on*, pp. 40–45, 2007.
- [26] K. Konolige, M. Agrawal, and J. Sola, "Large scale visual odometry for rough terrain," *Proc. International Symposium on Robotics Research*, 2007.
- [27] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3946–3952, 2008.
- [28] Nister, "Visual odometry," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, pp. 652–659, 2004.
- [29] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan 2006.
- [30] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2007.
- [31] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, "A robust visual odometry and precipice detection system using consumer-grade monocular vision," *Robotics and Automation, IEEE International Conference on*, pp. 3421–3427, 2005.
- [32] M. Brooks, W. Chojnacki, and L. Baumela, "Determining the egomotion of an uncalibrated camera from instantaneous optical flow," *J Opt Soc Am A*, vol. 14, no. 10, pp. 2670–2677, Jan 1997.
- [33] O. Koch and S. Teller, "Wide-area egomotion estimation from known 3d structure," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2007.
- [34] J. Gluckman and S. Nayar, "Ego-motion and omnidirectional cameras," *Computer Vision, 1998. Sixth International Conference on DOI - 10.1109/ICCV.1998.710838*, pp. 999–1005, 1998.
- [35] G. Bleser and D. Stricker, "Advanced tracking through efficient image processing and visual-inertial sensor fusion," *Computers & Graphics*, vol. 33, pp. 59–72, Jan 2009.
- [36] J. Hol, T. Schön, H. Luinge, P. Slycke, and F. Gustafsson, "Robust real-time tracking by fusing measurements from inertial and vision sensors," *Journal of Real-Time Image Processing*, vol. 2, pp. 149–160, Jan 2007.
- [37] L. Armesto, J. Tornero, and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *The International Journal of Robotics Research*, vol. 26, pp. 577–589, Jan 2007.

- [38] G. Bleser and D. Strickery, "Using the marginalised particle filter for real-time visual-inertial sensor fusion," *Mixed and Augmented Reality, Proceedings of the 7th IEEE/ACM International Symposium on*, Sep 2008.
- [39] G. Dubbelman, W. van der Mark, and F. Groen, "Accurate and robust ego-motion estimation using expectation maximization," *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3914–3920, 2008.
- [40] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fernstrom, J. DeLany, and R. Scabassi, "Determination of food portion size by image processing," *Engineering in Medicine and Biology Society, Annual International Conference of the IEEE*, pp. 871 – 874, 2008.
- [41] R. Weiss, P. J. Stumbo, and A. Divakaran, "Automatic food documentation and volume computation using digital imaging and electronic transmission," *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 42–44, Apr 2010.
- [42] 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," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 4, pp. 756 – 766, 2010.
- [43] Y. Yue, W. Jia, J. Fernstrom, R. Scabassi, M. Fernstrom, N. Yao, and M. Sun, "Food volume estimation using a circular reference in image-based dietary studies," *IEEE Annual Northeast Bioengineering Conference*, pp. 1 – 2, 2010.
- [44] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conference*, vol. 15, p. 50, 1988.
- [45] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, Jul 1981. [Online]. Available: <http://citeseer.ist.psu.edu/180224>
- [46] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Computer Vision, IEEE International Conference on*, pp. 666 – 673, 1999.
- [47] G. Strang, "Introduction to linear algebra, 3rd ed," *Wellesley-Cambridge Press*, 1998, 1998.
- [48] R. Taylor and O. Zienkiewicz, "The finite element method for solid and structural mechanics," *Butterworth-Heinemann*, 2005.

# Chapter 7

## Conclusion

This chapter presents a summary of the major findings and results in this work, including a summary of conclusions, a summary of primary contributions and a discussion of future works.

### 7.1 Conclusions

Currently, image based food recognition is still under development. Existing pattern recognition and classification algorithms show their insufficiency when applied in food recognition. Food volume estimation still relies on markers to indicate scales. As a result, the goal of this work presented in this thesis is to develop novel automatic food intake assessment methods through investigating new visual food recognition algorithms and marker-less volume calculation algorithms. This work explored the feasibilities to achieve these goals.

A prototype using existing pattern classification method was implemented. In the prototype, a feature based food classification approach and a multiple-view method to obtain the calorie values of food items through 3D model reconstruction (to calculate the volume) and occlusion reductions were developed. Food databases consisting of personal and global databases were constructed. The prototype was implemented on the iPhone platform. In the experiment, fruits, fast food, steaks and home-made food with certain shape patterns were tested. The evaluation results showed that the overall segmentation accuracy dropped when the number of food increased, and even though food in the test images had a certain shape pattern, the accuracy of recognition was 70% when number of references in the database is 20. The recognition accuracy increased to 90% when the number of references

of the same kind of food in the database was more than 50. But the prototype cannot be used for deformable shape foods.

The prototype was improved through three aspects, food recognition algorithms for deformable shape foods, database indexing mechanisms, and volume estimation methods.

The food recognition algorithms were improved through two steps. All the deformable food items were divided into two groups, regular shape food and arbitrary shape food. A multi-view food segmentation and recognition framework was developed for regular shape food items. Compared with standard object recognition algorithms, the results exhibit its accuracy of 84% and 91% when recognizing arbitrary number of or single food item respectively. For arbitrary shape food items, a food ingredient detector and a multi-view, multi-kernel based SVM food classifier was developed to classify food items. A food image database of 15262 food images was collected, among which 50% are used as training images and the other 50% are used for testing. The results are promising compared with existing food classification methods. The accuracy are increased by 60% for the most complex food compositions.

From a large-scale image database, it was time consuming to retrieve and classify images through pair-wise feature matching in the prototype system. An image indexing mechanism for efficient large-scale image retrieval and image recognition was developed to improve the prototype system. The indexing mechanism represented an image as a binary index with as few as 16 even 8 bits. In order to obtain the binary index, the image features were quantized into a short representation, and then further mapped into a binary value with the novel spatial hashing. In the experiment, the new method retrieved images faster than other feature-based indexing methods and the results promise a precision of 80%, which is higher than the standard vocabulary tree method.

A metric scale 3D model reconstruction method implemented on a smart phone was developed to reduce the credit card marker in the prototype system for food volume estimation. The applications of this method were not limited to food volume estimation, but also include arbitrary object dimension estimation. The advantage of this method was scale estimation and accuracy improvement by means of integrating camera and inertial measurement units. The inertial measurement units detected camera movement in metric scale but with drifts and noises. The camera recovered the motion in arbitrary scales. Through sensor fusion, the drifts from inertial sensors were estimated and reduced; the noises from the camera were decreased. The experiment shows promising results that the scale factor was estimated in tens of frames and the volume estimation is of about 90% accuracy.

## 7.2 Summary of contributions

In summary, the main contributions of this work to the field of visual food image recognition, image based object volume estimation and large-scale database indexing are:

1. A client-server food intake assessment system architecture was designed and implemented in the prototype system.
2. Traditional pattern classification algorithms were reviewed and implemented for food classification.
3. A multi-view object recognition framework was designed, and applied for food recognition.
4. Part-based models were improved to recognize food ingredients.
5. A multi-kernel support vector machine classifier was developed for arbitrary food image classification. With this method, the accuracy of arbitrary shape food recognition increased to 90%.
6. A food image database of 55 food types and 15262 food images were collected, labeled for food image recognition research.
7. An image feature indexing method was developed to increase image recognition accuracy and image retrieval efficiency in a large-scale image database with 11 million images.
8. A hierarchical database structure comprising of personal databases and global database was proposed to increase the speed of image search.
9. A credit card method was proposed in the prototype system for metric scale calculation and food volume estimation.
10. Inertial measurement units and cameras on smart phones are used together for markerless scale estimation to calculate distances between and volumes of objects.

## 7.3 Future works

This work has opened new and promising research perspectives that may lead to field applications in the future. Future research should address the following challenges:



1. Currently, the smart phone is used to gather meal information and transmit information. Images and data are processed on the server side. Light weight food recognition algorithms that are executable on commercial smart phones will make smart phone based food intake assessment more convenient and more popular.
2. A combination of visual food recognition and speech recognition will enhance the integrity and accuracy of the food intake assessment records. Moreover, the combination has potential for food amount estimation, based on the speech input of the user.
3. Recognition and volume estimation of food residues is still not solved. Food residues are more difficult to recognize than the whole meals, since the appearance of the residues does not have certain patterns. But recognizing food residues will be useful for accurate food intake assessment.
4. An automatic food image collection and labeling method will solve the problem that there is not any complete food image database. A food image database is the prerequisite for the development of mature food image recognition algorithms.

# References

- [1] “World Health Organization [internet]. Obesity and overweight; 2011 [Updated March 2011, cited October 2011]. Available from: <http://www.who.int/mediacentre/factsheets/fs311/en/>.”
- [2] “Anne Collins [internet]; Obesity Statistics; Available from: <http://www.annecollins.com/obesity/statistics-obesity.htm>.”
- [3] “Centers for Disease Control and Prevention [internet]. U.S. Obesity Trends; 2011 [Updated July 21, 2011, cited October 2011]. Available from: <http://www.cdc.gov/obesity/data/trends.html>.”
- [4] “At a glance 2009 - Obesity, halting the epidemic by making health easier,” *Center for Disease Control and Prevention [Online]*. Available: <http://www.cdc.gov/nccdphp/dnpa/obesity/>.
- [5] E. Finkelstein, I. Fiebelkorn, and G. Wang, “National medical spending attributable to overweight and obesity: How much, and who’s paying?” *Health Affairs Web Exclusive*, vol. 5, no. 14, 2003.
- [6] A. Ershow, J. Hill, and J. Baldwin, “Novel engineering approaches to obesity, overweight, and energy balance: public health needs and research opportunities,” *Engineering in Medicine and Biology Society, IEEE Annual International Conference of*, pp. 5212–5214, Jan 2004.
- [7] G. Godin, A. Bélanger-Gravel, A. marie Paradis, M.-C. Vohl, and L. Périusse, “A simple method to assess fruit and vegetable intake among obese and non-obese individuals,” *Can J Public Health*, vol. 99, no. 6, pp. 494–8, Jan 2008.
- [8] M. A. Murtaugh, K. ni Ma, T. Greene, D. Redwood, S. Edwards, J. Johnson, L. Tom-Orme, A. P. Lanier, J. A. Henderson, and M. L. Slattery, “Validation of a dietary history questionnaire for american indian and alaska native people,” *Ethn Dis*, vol. 20, no. 4, pp. 429–36, Feb 2011.
- [9] N. D. Wright, A. E. Groisman-Perelstein, J. Wylie-Rosett, N. Vernon, P. M. Diamantis, and C. R. Isasi, “A lifestyle assessment and intervention tool for pediatric

- weight management: the habits questionnaire,” *J Hum Nutr Diet*, vol. 24, no. 1, pp. 96–100, Feb 2011.
- [10] A. F. Smith, S. D. Baxter, J. W. Hardin, C. H. Guinn, and J. A. Royer, “Relation of children’s dietary reporting accuracy to cognitive ability,” *Am J Epidemiol*, vol. 173, no. 1, pp. 103–9, Jan 2011.
  - [11] L. A. Mainvil, C. C. Horwath, J. E. McKenzie, and R. Lawson, “Validation of brief instruments to measure adult fruit and vegetable consumption,” *Appetite*, vol. 56, no. 1, pp. 111–7, Feb 2011.
  - [12] M. A. Cardoso, L. Y. Tomita, and E. C. Laguna, “Assessing the validity of a food frequency questionnaire among low-income women in são paulo, southeastern brazil,” *Cad Saude Publica*, vol. 26, no. 11, pp. 2059–67, Nov 2010.
  - [13] F. H. Esfahani, G. Asghari, P. Mirmiran, and F. Azizi, “Reproducibility and relative validity of food group intake in a food frequency questionnaire developed for the tehran lipid and glucose study,” *J Epidemiol*, vol. 20, no. 2, pp. 150–8, Jan 2010.
  - [14] A. E. Dutman, A. Stafleu, A. Kruizinga, H. A. Brants, K. R. Westerterp, C. Kistemaker, W. J. Meuling, and R. A. Goldbohm, “Validation of an FFQ and options for data processing using the doubly labelled water method in children,” *Public Health Nutr*, pp. 1–8, Aug 2010.
  - [15] M. Aubertin-Leheudre, A. Koskela, A. Samaletdin, and H. Adlercreutz, “Plasma alkylresorcinol metabolites as potential biomarkers of whole-grain wheat and rye cereal fibre intakes in women,” *Br J Nutr*, vol. 103, no. 3, pp. 339–43, Feb 2010.
  - [16] G. L. Bowman, J. Shannon, E. Ho, M. G. Traber, B. Frei, B. S. Oken, J. A. Kaye, and J. F. Quinn, “Reliability and validity of food frequency questionnaire and nutrient biomarkers in elders with and without mild cognitive impairment,” *Alzheimer Dis Assoc Disord*, vol. 25, no. 1, pp. 49–57, Jan 2011.
  - [17] P. B. Ryan, K. A. Scanlon, and D. L. MacIntosh, “Analysis of dietary intake of selected metals in the nhexas-maryland investigation,” *Environ Health Perspect*, vol. 109, no. 2, pp. 121–8, Feb 2001.
  - [18] M. R. Ritchie, M. S. Morton, N. Deighton, A. Blake, and J. H. Cummings, “Plasma and urinary phyto-oestrogens as biomarkers of intake: validation by duplicate diet analysis,” *Br J Nutr*, vol. 91, no. 3, pp. 447–57, Mar 2004.
  - [19] O. Amft, “Automatic dietary monitoring using on-body sensors, detection of eating and drinking behaviour in healthy individuals,” *PhD dissertation, Swiss Federal Institute of Technology Zurich*, 2008, Jan 2008.
  - [20] K. Patrick, W. Griswold, F. Raab, and S. Intille, “Health and the mobile phone,” *American Journal of Preventive Medicine*, vol. 35, no. 2, pp. 177–181, Aug 2008.

- [21] “USDA’s center for nutrition policy and promotion, mypyramid,” [Online]. Available: <http://www.mypyramid.gov/>.
- [22] C. Tsai, G. Lee, F. Raab, G. Norman, T. Sohn, W. Griswold, and K. Patrick, “Usability and feasibility of pmeb: A mobile phone application for monitoring real time caloric balance,” *Mobile Networks and Applications*, vol. 12, no. 2-3, pp. 173–184, Jun 2007.
- [23] “My food phone,” [Online]. Available: <http://www.mycanutrition.com/>.
- [24] “Sensei diet program,” <http://www.sensei.com/sensei/>.
- [25] “My food diary,” [Online]. Available: <http://www.myfooddiary.com/>.
- [26] T. Toscos, A. Faber, S. An, and M. Gandhi, “Chick clique: persuasive technology to motivate teenage girls to exercise,” *Human factors in computing systems, CHI extended abstracts on*, pp. 1873–1878, 2006.
- [27] R. Oliveira and N. Oliver, “Triplebeat: enhancing exercise performance with persuasion,” *Human computer interaction with mobile devices and services, International conference on*, pp. 255–264, 2008.
- [28] S. Reddy, A. Parker, J. Hyman, and J. Burke, “Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype,” *Embedded networked sensors, workshop on*, pp. 13–17, Jan 2007.
- [29] Ø. Trier, A. Jain, and T. Taxt, “Feature extraction methods for character recognition—a survey,” *Pattern recognition*, vol. 29, no. 4, pp. 641–662, Jan 1996.
- [30] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2161–2168, 2006.
- [31] I. Woo, K. Otsmo, S. Kim, D. Ebert, E. Delp, and C. Boushey, “Automatic portion estimation and visual refinement in mobile dietary assessment,” *Computational Image VIII, Proceedings of the SPIE*, vol. 7533, pp. 1–10, Dec 2010.
- [32] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” *Computer Vision, IEEE International Conference on*, pp. 666 – 673, 1999.
- [33] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision, 2nd ed,” *Book, Cambridge University Press*, 2004, Jan 2004.
- [34] F. Kong and J. Tan, “A 3d object model for wireless camera networks with network constraints,” *Distributed Smart Cameras, Third ACM/IEEE International Conference on*, pp. 1–8, Aug 2009.

- [35] "U.s. department of agriculture, agricultural research service. 2009." *USDA National Nutrient Database for Standard Reference, Release 22. Nutrient Data Laboratory Home Page*, <http://www.ars.usda.gov/ba/bhnrc/ndl>, 2009.
- [36] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, Jan 2004.
- [37] K. Mikolajczyk, B. Leibe, B. Schiele, M. Syst, and G. Darmstadt, "Local features for object class recognition," *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1792 – 1799, 2005.
- [38] D. Lowe, "Object recognition from local scale-invariant features," *Computer Vision, IEEE International Conference on*, vol. 2, pp. 1150 – 1157, 1999.
- [39] S. Helmer and D. Lowe, "Object class recognition with many local features," *Computer Vision and Pattern Recognition Workshop, Conference on*, pp. 187–195, 2004.
- [40] R. Bolle, J. Connell, N. Haas, R. Mohan, and G. Taubin, "Veggie vision: A produce recognition system," *Automatic Identification Advanced Technologies, IEEE Workshop on*, pp. 35–38, Feb 1997.
- [41] J. Salvi, X. Armangue, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern recognition*, vol. 35, pp. 1617–1635, 2002.
- [42] G. Strang, "Introduction to linear algebra, 3rd ed," *Wellesley-Cambridge Press*, 1998, 1998.
- [43] R. Taylor and O. Zienkiewicz, "The finite element method for solid and structural mechanics," *Butterworth-Heinemann*, 2005.
- [44] "Stockfood - the food image agency. food pictures for professionals," [online]<http://www.stockfood.com>.
- [45] "Photomodeler: Accurate and affordable 3d modeling-measuring-scanning," <http://www.photomodeler.com/index.htm>.
- [46] F. Zhu, A. Mariappan, C. Boushey, D. Kerr, K. Lutes, D. Ebert, and E. Delp, "Technology-assisted dietary assessment," *Computational Imaging, Proceedings of the IS&T/SPIE Conference on*, pp. 1–10, Jan 2008.
- [47] Y. Fujiki, K. Kazakos, C. Puri, and P. Buddharaju, "Neat-o-games: blending physical activity and fun in the daily routine," *Computers in Entertainment*, vol. 6, no. 2, pp. 1–22, 2008.

- [48] Z. Cheng, D. Devarajan, and R. Radke, "Determining vision graphs for distributed camera networks using feature digests," *Advances in Signal Processing, EURASIP Journal on*, vol. 2007, no. 1, pp. 220–231, Jan 2007.
- [49] C. Christoudias, R. Urtasun, and T. Darrell, "Unsupervised distributed feature selection for multi-view object recognition," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2008.
- [50] A. Yang, S. Maji, C. Christoudias, and T. Darrell, "Multiple-view object recognition in band-limited distributed camera networks," *Distributed Smart Cameras, Third ACM/IEEE International Conference on*, pp. 1–8, Aug 2009.
- [51] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Computer Vision, International Journal of*, vol. 47, no. 1/2/3, pp. 7–42, Jan 2002.
- [52] T. Jebara, A. Azarbayejani, and A. Pentland, "3d structure from 2d motion," *IEEE Signal Processing Magazine*, vol. 16, no. 3, pp. 66–84, Jan 1999.
- [53] S. Seitz, B. Curless, J. Diebel, and D. Scharstein, "A comparison and evaluation of multi-view stereo reconstruction algorithms," *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 1, pp. 519 – 528, 2006.
- [54] D. Kien, "A review of 3d reconstruction from video sequences," *Intelligent Sensory Information Systems technical report, University of Amsterdam*, 2005, 2005.
- [55] A. Saxena, M. Sun, and A. Ng, "Learning 3-d scene structure from a single still image," *Computer Vision, IEEE International Conference on*, pp. 1–8, 2007.
- [56] —, "3-d reconstruction from sparse views using monocular vision," *Computer Vision, IEEE International Conference on*, pp. 1–8, 2007.
- [57] W. Zhang and T. Chen, "A probabilistic framework for geometry reconstruction using prior information," *Image Processing, IEEE International Conference on*, vol. 2, pp. 529–532, Jan 2007.
- [58] R. Szeliski, "Computer vision: Algorithms and applications," *Springer*, 2010.
- [59] C. Martin, S. Kaya, and B. Gunturk, "Quantification of food intake using food image analysis," *Engineering in Medicine and Biology Society. Annual International Conference of the IEEE*, pp. 6869 – 6872, 2009.
- [60] W. Wu and J. Yang, "Fast food recognition from videos of eating for calorie estimation," *Multimedia and Expo, IEEE international Conference on*, pp. 1210 – 1213, Jan 2009.

- [61] 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," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 4, pp. 756 – 766, 2010.
- [62] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2249 – 2256, 2010.
- [63] P. Viola and M. Jones, "Robust real-time face detection," *Computer Vision, IEEE International Conference on*, vol. 2, pp. 747 – 747, Jul 2001.
- [64] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *Computer Vision, European Conference on*, pp. 18 – 32, Jan 2000.
- [65] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 746 – 751, Jun 2000.
- [66] D. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, Jan 1981.
- [67] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Apr 2005.
- [68] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Jan 2004.
- [69] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, Jan 2001.
- [70] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, pp. 761–767, Jan 2004.
- [71] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 506–513, 2004.
- [72] Y. Jia, J. Wang, G. Zeng, H. Zha, and X.-S. Hua, "Optimizing kd-trees for scalable visual descriptor indexing," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3392 – 3399, 2010.
- [73] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," *Computer Vision. IEEE International Conference on*, vol. 1, pp. 273–280, 2003.

- [74] A. Collet and S. Srinivasa, “Efficient multi-view object recognition and full pose estimation,” *Robotics and Automation, IEEE International Conference on*, pp. 2050–2055, 2010.
- [75] N. Naikal, A. Yang, and S. Sastry, “Towards an efficient distributed object recognition system in wireless smart camera networks,” *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1 – 8, 2010.
- [76] D. Williams, “Bayesian data fusion of multiview synthetic aperture sonar imagery for seabed classification,” *Image Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1239 – 1254, 2009.
- [77] M. Chen, K. Dhingra, W. Wu, and L. Yang, “PFID: Pittsburgh fast-food image dataset,” *Image Processing, IEEE International Conference on*, pp. 289 – 292, Jan 2009.
- [78] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 8, no. 6, pp. 679 –698, 1986.
- [79] F. Kong and J. Tan, “Dietcam: Regular shape food recognition with a camera phone,” *Body Sensor Networks (BSN), 2011 International Conference on*, pp. 127 – 132, 2011.
- [80] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Computing Surveys*, vol. 35, no. 4, pp. 399–458, Jan 2003. [Online]. Available: <http://portal.acm.org/citation.cfm?id=954339.954342>
- [81] M. Varma and A. Zisserman, “Classifying images of materials: Achieving viewpoint and illumination,” *European Conference on Computer Vision*, Jan 2002.
- [82] C. Schmid, “Constructing models for content-based image retrieval,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. II–39 – II–45, 2001.
- [83] J. D. Bonet and P. Viola, “Texture recognition using a non-parametric multi-scale statistical model,” *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 641 – 647, 1998.
- [84] M. Do and M. Vetterli, “Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance,” *Image Processing, IEEE Transactions on*, vol. 11, no. 2, pp. 146 – 158, 2002.
- [85] Y. Xu, X. Yang, H. Ling, and H. Ji, “A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 161 – 168, 2010.



- [86] M. Nilsback and A. Zisserman, “A visual vocabulary for flower classification,” *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1447 – 1454, Jan 2006.
- [87] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 – 8, 2008.
- [88] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [89] M. Sadeghi and A. Farhadi, “Recognition using visual phrases,” *Computer Vision and Pattern Recognition, IEEE Conference*, pp. 1745–1752, 2011.
- [90] H. Jia and A. Martinez, “Support vector machines in face recognition with occlusions,” *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 136 – 141, May 2009.
- [91] N. Cristianini and J. Shawe-Taylor, “An introduction to support vector machines,” *Cambridge University Press*, 2000.
- [92] S. Maji, A. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 – 8, 2008.
- [93] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, p. 13, 2006.
- [94] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” *Conference On Image And Video Retrieval, Proceedings of the 6th ACM international conference on Image and video retrieval*, Jan 2007.
- [95] K. Grauman and T. Darrell, “The pyramid match kernel: discriminative classification with sets of image features,” *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1458 – 1465 Vol. 2, 2005.
- [96] L. Duan, D. Xu, I. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data,” *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1959 – 1966, 2010.
- [97] M. Varma and D. Ray, “Learning the discriminative power-invariance trade-off,” *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1 – 8, 2007.

- [98] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, “Multiple kernels for object detection,” *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 606 – 613, 2009.
- [99] P. Gehler and S. Nowozin, “Let the kernel figure it out; principled learning of pre-processing for kernel classifiers,” *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2836 – 2843, 2009.
- [100] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” *Proceedings of the 25th International Conference on Very Large Data Bases*, pp. 518—529, Jan 1999.
- [101] H. Jegou, M. Douze, C. Schmid, and P. Perez, “Aggregating local descriptors into a compact image representation,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3304 – 3311, 2010.
- [102] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [103] H. Cevikalp, B. Triggs, F. Jurie, and R. Polikar, “Margin-based discriminant dimensionality reduction for visual recognition,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [104] C. Silpa-Anan and R. Hartley, “Optimised kd-trees for fast image descriptor matching,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [105] P. Jain, B. Kulis, and K. Grauman, “Fast image search for learned metrics,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [106] Y. Mu, J. Shen, and S. Yan, “Weakly-supervised hashing in kernel space,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3344 – 3351, 2010.
- [107] A. Torralba, R. Fergus, and Y. Weiss, “Small codes and large image databases for recognition,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [108] T. Yeh, J. Lee, and T. Darrell, “Adaptive vocabulary forests br dynamic indexing and category learning,” *Computer Vision, IEEE International Conference on*, pp. 1 – 8, 2007.
- [109] J. Wang, S. Kumar, and S.-F. Chang, “Semi-supervised hashing for scalable image retrieval,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 3424 – 3431, 2010.

- [110] C. Leistner, H. Grabner, and H. Bischof, “Semi-supervised boosting using visual similarity learning,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1 – 8, 2008.
- [111] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, Jan 2001.
- [112] Z. Wu, Q. Ke, M. Isard, and J. Sun, “Bundling features for large scale partial-duplicate web image search,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 25 – 32, 2009.
- [113] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, pp. 2169 – 2178, 2006.
- [114] I. Joliffe, “Principal component analysis,” *Springer-Verlag*, 1986.
- [115] K. Fukunaga and W. Koontz, “Application of the karhunen-loève expansion to feature selection and ordering,” *Computers, IEEE Transactions on*, vol. C-19, no. 4, pp. 311 – 318, 1970.
- [116] H. Murase and S. Nayar, “Detection of 3d objects in clustered scenes using hierarchical eigenspace,” *Pattern Recognition Letters*, vol. 18, no. 4, 1997.
- [117] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” *Technical Report 7694, Caltech*, Aug 2007.
- [118] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 248 – 255, Jun 2009.
- [119] A. Vedaldi and B. Fulkerson, “Vlfeat: An open and portable library of computer vision algorithms,” <http://www.vlfeat.org/>, 2008.
- [120] C. Jerian and R. Jain, “Structure from motion-a critical analysis of methods,” *Systems, Man and Cybernetics, IEEE Transactions on DOI - 10.1109/21.97478*, vol. 21, no. 3, pp. 572–588, 1991.
- [121] T. Huang and A. Netravali, “Motion and structure from feature correspondences: a review,” *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, 1994.
- [122] J. Oliensis, “A critique of structure-from-motion algorithms,” *Computer Vision and Image Understanding*, vol. 80, no. 2, Nov 2000.

- [123] R. Szeliski and S. Kang, "Recovering 3d shape and motion from image streams using nonlinear least squares," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 752–753, 1993.
- [124] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment—a modern synthesis," *Vision algorithms: theory and practice*, pp. 153–177, 2000.
- [125] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Jan 1981.
- [126] D. Nistér, "Preemptive ransac for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, Dec 2005.
- [127] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, Jan 2000.
- [128] G. Qian and R. Chellappa, "Structure from motion using sequential monte carlo methods," *International Journal of Computer Vision*, vol. 59, no. 1, Aug 2004.
- [129] G. Qian, R. Chellappa, and Q. Zheng, "Robust structure from motion estimation using inertial data," *J Opt Soc Am A Opt Image Sci Vis*, vol. 18, no. 12, pp. 2982–97, Dec 2001.
- [130] P. Germeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *The International Journal of Robotics Research*, vol. 26, pp. 591–605, Jan 2007.
- [131] G. Bleser, "Towards visual-inertial slam for mobile augmented reality," *PhD Dissertation, Technical University Kaiserslautern, Dr. Hut Verlag*, Mar 2009.
- [132] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–195, Apr 1998.
- [133] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, pp. 87–119, 1995.
- [134] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular slam," *International Conference on Unmanned Aerial Vehicles, Dubai*, 2010.
- [135] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, vol. 4, pp. 4007–4012 vol.4, 2004.

- [136] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers: Field reports,” *J. Field Robotics*, vol. 24, no. 3, Mar 2007.
- [137] Y. Cheng, M. Maimone, and L. Matthies, “Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging,” *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 54–62, 2006.
- [138] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, “Real-time monocular visual odometry for on-road vehicles with 1-point ransac,” *Robotics and Automation, IEEE International Conference on*, pp. 4293–4299, 2009.
- [139] A. Comport, E. Malis, and P. Rives, “Accurate quadrifocal tracking for robust 3d visual odometry,” *Robotics and Automation, IEEE International Conference on*, pp. 40–45, 2007.
- [140] K. Konolige, M. Agrawal, and J. Sola, “Large scale visual odometry for rough terrain,” *Proc. International Symposium on Robotics Research*, 2007.
- [141] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3946–3952, 2008.
- [142] Nister, “Visual odometry,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, pp. 652–659, 2004.
- [143] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan 2006.
- [144] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, “Visual odometry system using multiple stereo cameras and inertial measurement unit,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2007.
- [145] J. Campbell, R. Sukthankar, I. Nourbakhsh, and A. Pahwa, “A robust visual odometry and precipice detection system using consumer-grade monocular vision,” *Robotics and Automation, IEEE International Conference on*, pp. 3421–3427, 2005.
- [146] M. Brooks, W. Chojnacki, and L. Baumela, “Determining the egomotion of an uncalibrated camera from instantaneous optical flow,” *J Opt Soc Am A*, vol. 14, no. 10, pp. 2670–2677, Jan 1997.
- [147] O. Koch and S. Teller, “Wide-area egomotion estimation from known 3d structure,” *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 1–8, 2007.
- [148] J. Gluckman and S. Nayar, “Ego-motion and omnidirectional cameras,” *Computer Vision, 1998. Sixth International Conference on DOI - 10.1109/ICCV.1998.710838*, pp. 999–1005, 1998.

- [149] G. Bleser and D. Stricker, “Advanced tracking through efficient image processing and visual-inertial sensor fusion,” *Computers & Graphics*, vol. 33, pp. 59–72, Jan 2009.
- [150] J. Hol, T. Schön, H. Luinge, P. Slycke, and F. Gustafsson, “Robust real-time tracking by fusing measurements from inertial and vision sensors,” *Journal of Real-Time Image Processing*, vol. 2, pp. 149–160, Jan 2007.
- [151] L. Armesto, J. Tornero, and M. Vincze, “Fast ego-motion estimation with multi-rate fusion of inertial and vision,” *The International Journal of Robotics Research*, vol. 26, pp. 577–589, Jan 2007.
- [152] G. Bleser and D. Stricker, “Using the marginalised particle filter for real-time visual-inertial sensor fusion,” *Mixed and Augmented Reality, Proceedings of the 7th IEEE/ACM International Symposium on*, Sep 2008.
- [153] G. Dubbelman, W. van der Mark, and F. Groen, “Accurate and robust ego-motion estimation using expectation maximization,” *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3914–3920, 2008.
- [154] M. Sun, Q. Liu, K. Schmidt, J. Yang, N. Yao, J. Fernstrom, M. Fernstrom, J. DeLany, and R. Scabassi, “Determination of food portion size by image processing,” *Engineering in Medicine and Biology Society, Annual International Conference of the IEEE*, pp. 871 – 874, 2008.
- [155] R. Weiss, P. J. Stumbo, and A. Divakaran, “Automatic food documentation and volume computation using digital imaging and electronic transmission,” *Journal of the American Dietetic Association*, vol. 110, no. 1, pp. 42–44, Apr 2010.
- [156] Y. Yue, W. Jia, J. Fernstrom, R. Scabassi, M. Fernstrom, N. Yao, and M. Sun, “Food volume estimation using a circular reference in image-based dietary studies,” *IEEE Annual Northeast Bioengineering Conference*, pp. 1 – 2, 2010.
- [157] C. Harris and M. Stephens, “A combined corner and edge detector,” *Alvey Vision Conference*, vol. 15, p. 50, 1988.
- [158] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” *Proc 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, Jul 1981.