Pervasive and Mobile Computing 8 (2012) 147–163

Contents lists available at SciVerse ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

DietCam: Automatic dietary assessment with mobile camera phones

Fanyu Kong *, Jindong Tan

Department of Electrical and Computer Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI, 49931, United States

ARTICLE INFO

Article history:
Received 14 May 2010
Received in revised form 28 January 2011
Accepted 18 July 2011
Available online 26 July 2011

Keywords:
Food intake assessment
Calorie estimation
Mobile 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.

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 US 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. 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 US 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 almost 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

* Corresponding author.
E-mail addresses: fkong@mtu.edu (F. Kong), jitan@mtu.edu (J. Tan).

1574-1192/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2011.07.003

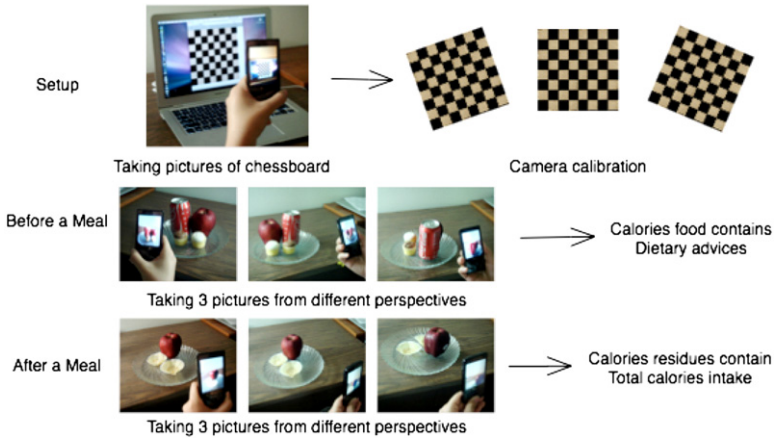


Fig. 1. Expected usage. The calorie information, which is a key to the obesity problem, will be extracted from three images or a short piece of video of the foods.

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. 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° 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 content 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 7 evaluates DietCam with field experiments. We discuss the related work in Section 9, and conclude the paper in Section 10.

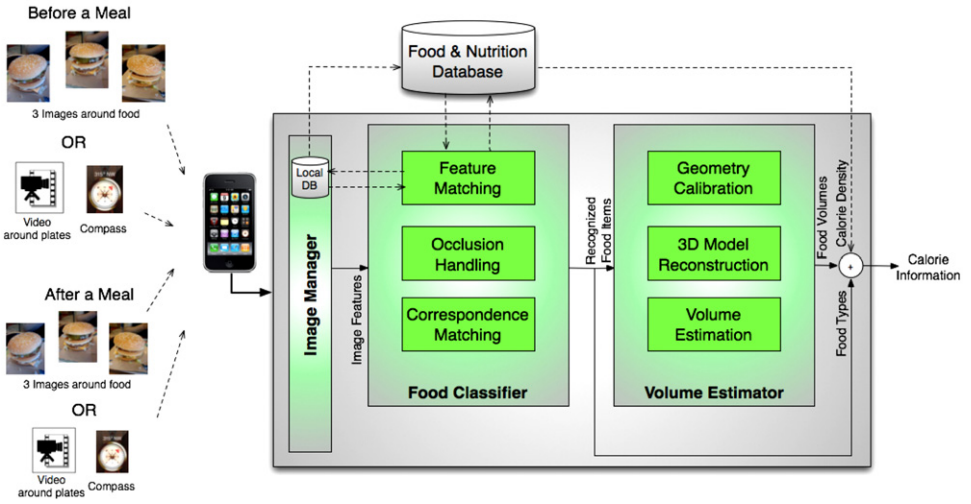


Fig. 2. System architecture. The types of food in a meal are classified by the food classifier. The volume of every food item is generated by the volume estimator.

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. 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 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.

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.

3.1. Food features

The image features used by the food classifier are extracted by the image manager, whose role is defined in Section 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 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].

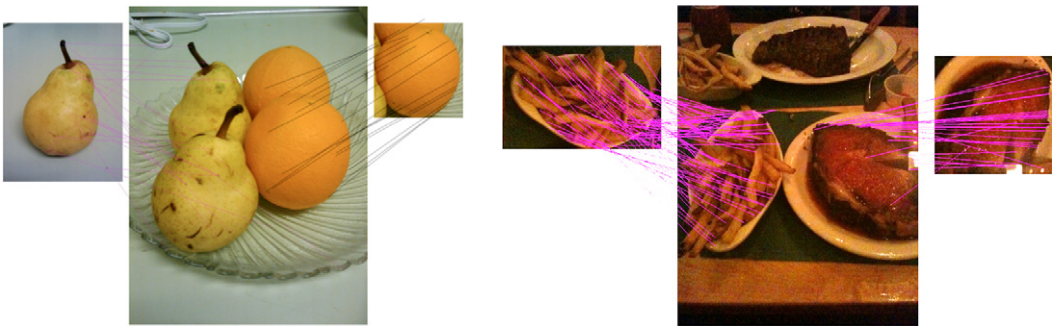
3.2. Food recognition

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

Fig. 3. Training set of cheeseburgers and apples from different perspectives.



(a) Segmentation of pears and oranges. (b) Segmentation of steaks and fries.

Fig. 4. Feature matching based food segmentation. (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. (b) shows the segmentation of a steak meal in dark lighting conditions.

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. 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 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, $\omega_1, \omega_2, \ldots, \omega_M$, and an unknown food item represented by a visual word vector x, the possibilities x belongs to $\omega_1, \omega_2, \ldots, \omega_M$ are $P(\omega_i|x)$, $i = 1, 2, \ldots, M$. The most probable food type of food item x is type ω_i which has the largest P value. P can be calculated through the Bayes rule. The class conditional probability density functions $p(x|\omega_i)$, $i = 1, 2, \ldots, M$ 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 ω_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.

3.3. Food segmentation

A meal always consists of more than one food item on the plate. Segmenting them 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 segmentation could be done at the same time as matching image features to the database. 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. 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.

Algorithm 1 Redundancy Reduction

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

```
n ⇐ 3, number of image pairs
for i = 1 to n do
    Find ith image pair I1, I2 from I
    Find 8 corresponding boundary SIFT feature point in I1, I2
    Calculate the fundamental matrix of the I1, I2
    Find all the food items {F1} in I1 with {S}
    for Every food item f in {F1} do
        Find correspondent searching window in I2
        if The same type of food exists in the window then
            Keep f
        else
            F = F − f
        end if
    end for
end for
```

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 item 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.

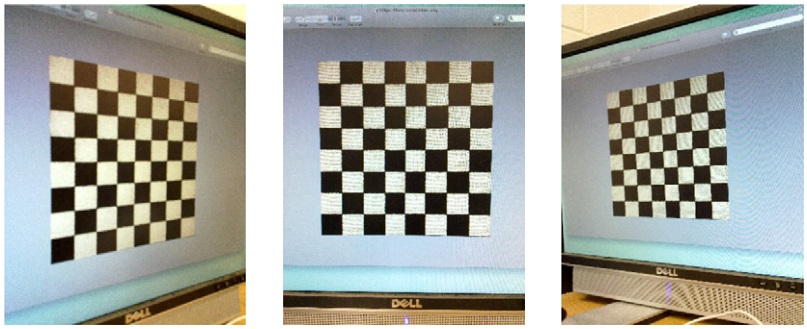
Algorithm 1 shows the whole procedure. It operates a pair of images one at a time. Every pair of images will be processed. For an image pair, the algorithm starts with initializing the geometry relation between them. The SIFT features are used again as the image descriptor, by matching which, the similarities 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.

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 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.

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



(a) Left. (b) Center. (c) Right.

Fig. 5. Intrinsic parameter calibration. Three different perspective chessboard images will give enough information to calibrate the camera, which is easy and convenient for the users.

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.

4.1.1. Intrinsic parameter 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

$$P = A\begin{bmatrix} R & T \end{bmatrix}, \tag{1}$$

where

$$A = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

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, α and β 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 calculates 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. 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.

4.1.2. Extrinsic parameters calibration

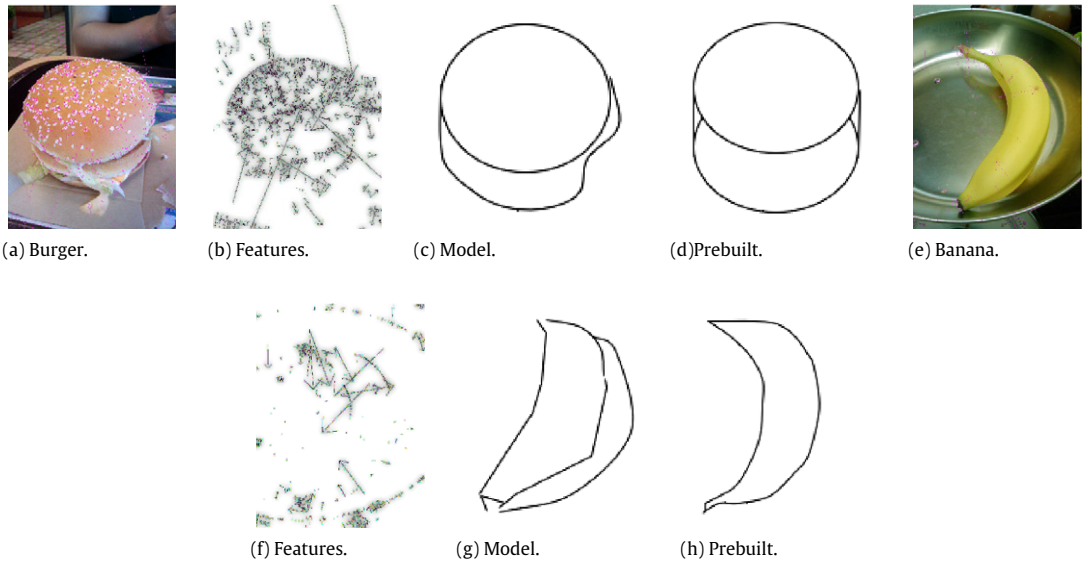
Unlike the static intrinsic parameters A, the extrinsic parameters R and T are always changing when taking pictures. Consequently, they need to be estimated every time right after the food pictures have been taken. We assume these three images are taken by three different cameras at the same time. Therefore, 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 can be chosen as

$$P = A\begin{bmatrix} I & 0 \end{bmatrix}, \tag{3}$$

where I is a 3×3 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

$$P' = A\begin{bmatrix} R & T \end{bmatrix}. \tag{4}$$



(a) Burger. (b) Features. (c) Model. (d)Prebuilt. (e) Banana.

(f) Features. (g) Model. (h) Prebuilt.

Fig. 6. (a) is the SIFT extraction of a burger. (b) shows these points in the 3D space. (c) shows the 3D model reconstructed directly from those points. (d) presents the supposed prebuilt model, where it is clear to see the differences. Similarly, (e)–(h) present an example of a banana, where the reconstructed model has an obviously larger volume than the prebuilt model.

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

$$pEp' = 0, \tag{5}$$

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,

$$E = [T]_{\times} R, \tag{6}$$

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].

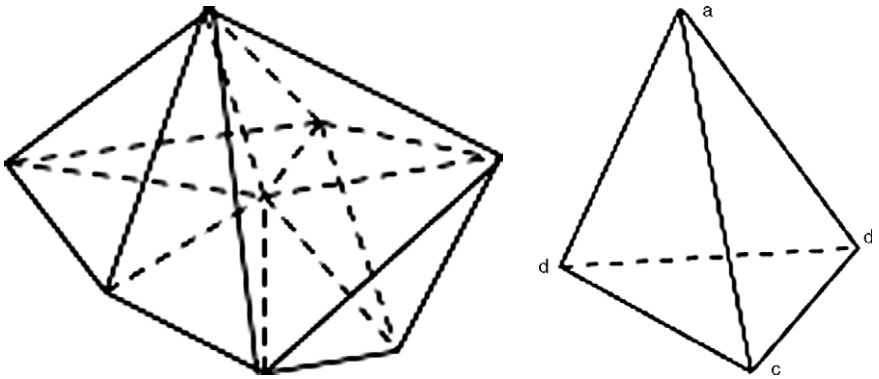
4.2. 3D model reconstruction

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 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. 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.

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.



(a) Arbitrary food. (b) Divided part.

Fig. 7. Tetrahedrons of an arbitrary shaped food item. (a) shows all the tetrahedrons: the point in the center is the estimated mass point. (b) shows a single tetrahedron with point coordinates a, b, c, and d. The volume of this tetrahedron is calculated with Eq. (7).

Table 1
Calorie density chart.

Food type	Total kcal in 100 g
Apple, raw, with skin	83
Wheat brand bread	248
Sliced sourdough bread	255
Cheeseburger	286
Steak	176
Sandwich, ham & cheese	241

For an arbitrary shaped model, whose volume is not computable directly from the coordinates 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. 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

$$V = \frac{(a-d)\cdot((b-d)\times(c-d))}{6} \tag{7}$$

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

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 1 shows part of the calorie density chart DietCam makes use of, which is from online resources [21]. With the volume v, mass density ρ, calorie density c, the number of calories is defined as

$$Cal = v \times \rho \times c. \tag{8}$$

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 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.

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.

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.

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, homemade 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 homes 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.

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. 8(a)). When the user taps inside the "Click to Take an Image" 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. 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.

The "Camera" tab leads the user to the camera calibration menu (as shown in Fig. 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. 8(d)). Since the application is currently developed for an iPhone, default camera information is

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

Fig. 8. Calorie calculation function.

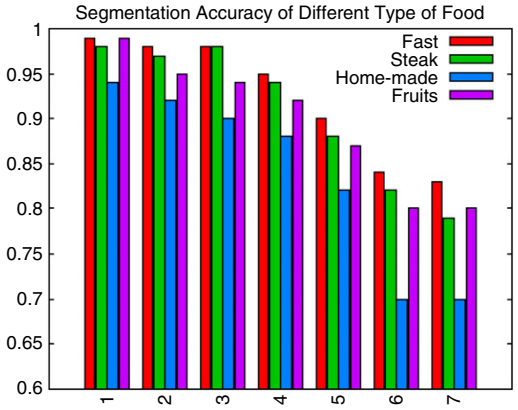


Fig. 9. The segmentation accuracy drops when the number of food items in the plate increases.

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.

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 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. 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

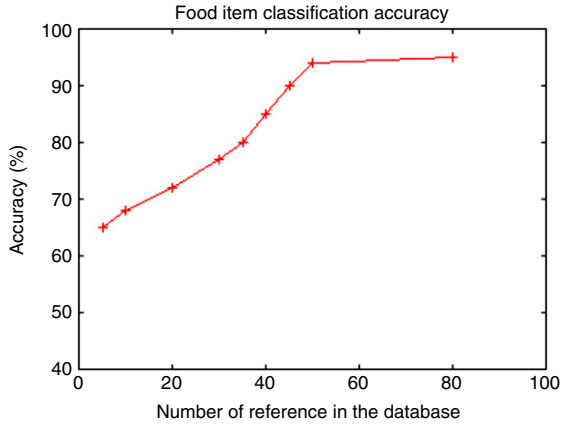


Fig. 10. Food item classification accuracy.

Table 2
Similar food classification accuracy (sample size: 20).

Hamburger	Cheeseburger	Double cheeseburger	Veggie subway	Philly subway
0.95	0.85	0.85	0.90	0.85

results are shown in Fig. 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 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.

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. 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. 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. (9).

$$D = \frac{\sum_{i=1}^{n} |V_i - M_i|}{n}, \tag{9}$$

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

Table 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.


(a) Apple.


(b) Banana.


(c) Burger.


(d) Steak.

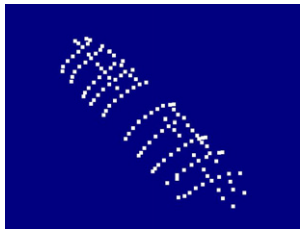

(e) Sandwich.


(f) Orange.


(g) Pear.

Fig. 11. Food types used to test the volume estimation.


(a) A sandwich with markers.


(b) 3D model built by PhotoModeler with markers.

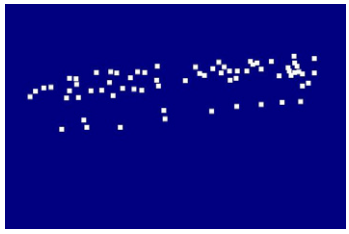

(c) 3D model built by the iPhone without markers.

Fig. 12. A sandwich in the experiment.

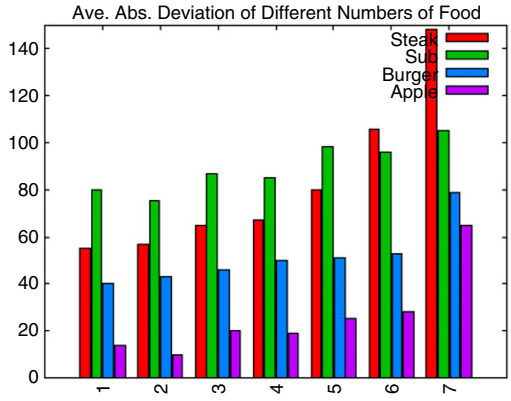


Fig. 13. Average absolute deviations with increased number of food item: cm^3.

Fig. 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 3
Measured and estimated food volumes: cm^3 (sample size: 10).

Measured value	Mean	Apple	Orange	Pear	Banana	Burger	Sub	Steak
		310.5	207	221	215.8	678.2	1280.1	288.5
Estimated without model	Mean	275.8	185.4	192	193.3	571.9	948.2	242.5
	Average absolute deviation	34.7	21.6	29	22.5	106.3	331.9	46
Estimated with model	Mean	286.7	198.4	194.2	204.1	623	1211.7	na
	Average absolute deviation	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.

Fig. 14. Calorie monitoring with OCR for labeled food items and drinks.

(a) Fruit residue.

(b) Fruit residue.

(c) Steak residue.

(d) Steak residue.

Fig. 15. Food residue recognition.

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. 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. 15. The volumes were estimated by the arbitrary model method. Arbitrary residue classification is one of our future works.

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, 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.

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. Woo et al. [17] and Zhu et al. [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 transmission, 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].

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, S. Intille, Health and the mobile phone, American Journal of Preventive Medicine 35 (2) (2008) 177–181.
[2] Obesity statistics, [Online]. Available: http://www.annecollins.com/obesity/statistics-obesity.htm.
[3] At a glance 2009 – obesity, halting the epidemic by making health easier, [Online]. Available: http://www.cdc.gov/nccdphp/dnpa/obesity/.
[4] E. Finkelstein, I. Fiebelkorn, G. Wang, National medical spending attributable to overweight and obesity: how much, and who's paying? Health Affairs Web Exclusive 5 (14) (2003).
[5] A. Ershow, J. Hill, J. Baldwin, Novel engineering approaches to obesity, overweight, and energy balance: public health needs and research opportunities, in: Engineering in Medicine and Biology Society, IEEE Annual International Conference of, 2004, pp. 5212–5214.
[6] O. Amft, Automatic dietary monitoring using on-body sensors, detection of eating and drinking behaviour in healthy individuals, Ph.D. Dissertation, Swiss Federal Institute of Technology Zurich, 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, K. Patrick, Usability and feasibility of pmeb: a mobile phone application for monitoring real time caloric balance, Mobile Networks and Applications 12 (2–3) (2007) 173–184.
[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, M. Gandhi, Chick clique: persuasive technology to motivate teenage girls to exercise, Human factors in computing systems, CHI Extended Abstracts (2006) 1873–1878.
[13] R. Oliveira, N. Oliver, Triplebeat: enhancing exercise performance with persuasion, in: Human Computer Interaction with Mobile Devices and Services, International Conference on, 2008, pp. 255–264.
[14] S. Reddy, A. Parker, J. Hyman, J. Burke, Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype, in: Embedded Networked Sensors, Workshop on, 2007, pp. 13–17.
[15] Ø. Trier, A. Jain, T. Taxt, Feature extraction methods for character recognition—a survey, Pattern Recognition 29 (4) (1996) 641–662.
[16] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Computer Vision and Pattern Recognition, IEEE Conference on, 2006, pp. 2161–2168.
[17] I. Woo, K. Otsmo, S. Kim, D. Ebert, E. Delp, C. Boushey, Automatic portion estimation and visual refinement in mobile dietary assessment, Computational Image VIII, in: Proceedings of the SPIE 7533, 2010, pp. 1–10.
[18] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, in: Computer Vision, IEEE International Conference on, 1999, pp. 666–673.
[19] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Book, Cambridge University Press, 2004.
[20] F. Kong, J. Tan, A 3d object model for wireless camera networks with network constraints, in: Distributed Smart Cameras, Third ACM/IEEE International Conference, 2009, pp. 1–8.
[21] US 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.
[22] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, computer vision, International Journal 60 (1) (2004) 63–86.
[23] K. Mikolajczyk, B. Leibe, B. Schiele, M. Syst, G. Darmstadt, Local features for object class recognition, in: Computer Vision, IEEE International Conference on, 2, 2005, pp. 1792–1799.
[24] D. Lowe, Object recognition from local scale-invariant features, in: Computer Vision, IEEE International Conference on, 2, 1999, pp. 1150–1157.
[25] S. Helmer, D. Lowe, Object class recognition with many local features, in: Computer Vision and Pattern Recognition Workshop, Conference on, 2004, pp. 187–195.
[26] R. Bolle, J. Connell, N. Haas, R. Mohan, G. Taubin, Veggie vision: a produce recognition system, in: Automatic Identification Advanced Technologies, IEEE Workshop on, 1997, pp. 35–38.
[27] J. Salvi, X. Armangue, J. Batlle, A comparative review of camera calibrating methods with accuracy evaluation, Pattern Recognition 35 (2002) 1617–1635.
[28] G. Strang, Introduction to Linear Algebra, third ed., Wellesley–Cambridge Press, 1998.
[29] R. Taylor, 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, E. Delp, Technology-assisted dietary assessment, computational imaging, in: Proceedings of the IS&T/SPIE Conference, 2008, pp. 1–10.
[33] Y. Fujiki, K. Kazakos, C. Puri, P. Buddharaju, Neat-o-games: blending physical activity and fun in the daily routine, Computers in Entertainment 6 (2) (2008) 1–22.
[34] Z. Cheng, D. Devarajan, R. Radke, Determining vision graphs for distributed camera networks using feature digests, Advances in Signal Processing, EURASIP Journal (1) (2007) 220–231.
[35] C. Christoudias, R. Urtasun, T. Darrell, Unsupervised distributed feature selection for multi-view object recognition, in: Computer Vision and Pattern Recognition, IEEE Conference on, 2008, pp. 1–8.
[36] A. Yang, S. Maji, C. Christoudias, T. Darrell, Multiple-view object recognition in band-limited distributed camera networks, in: Distributed Smart Cameras, Third ACM/IEEE International Conference on, 2009, pp. 1–8.
[37] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, computer vision, International Journal 47 (1–3) (2002) 7–42.

[38] T. Jebara, A. Azarbayejani, A. Pentland, 3d structure from 2d motion, IEEE Signal Processing Magazine 16 (3) (1999) 66–84.
[39] S. Seitz, B. Curless, J. Diebel, D. Scharstein, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Computer Vision and Pattern Recognition, IEEE Conference on, 1, 2006, pp. 519–528.
[40] D. Kien, A review of 3d reconstruction from video sequences, in: Intelligent Sensory Information Systems Technical Report, University of Amsterdam, 2005.
[41] A. Saxena, M. Sun, A. Ng, Learning 3-d scene structure from a single still image, in: Computer Vision, IEEE International Conference on, 2007, pp. 1–8.
[42] A. Saxena, M. Sun, A. Ng, 3-d reconstruction from sparse views using monocular vision, in: Computer Vision, IEEE International Conference on, 2007, pp. 1–8.
[43] W. Zhang, T. Chen, A probabilistic framework for geometry reconstruction using prior information, in: Image Processing, IEEE International Conference on, 2, 2007, pp. 529–532.