Food Region Segmentation in Meal Images Using Touch Points

Chamin Morikawa, Haruki Sugiyama, Kiyoharu Aizawa
Interfaculty Initiative in Information Studies
The University of Tokyo
ii.totoro@gmail.com, {aizawa,sugi}@hal.t.u-tokyo.ac.jp

ABSTRACT

We propose an interactive scheme for segmenting meal images for automated dietary assessment. A smartphone user photographs a meal and marks a few touch points on the resulting image. The segmentation algorithm initializes a set of food segments with the touch points, and grows them using local image features. We evaluate the algorithm with a data set consisting of 300 manually segmented meal images. The precision of segmentation is 0.87, compared with 0.70 for fully automatic segmentation. The results show that the precision of segmentation was significantly improved by incorporating minimal user intervention.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Human Factors

Keywords

Interactive segmentation, food image analysis, multimedia.

1. INTRODUCTION

With increased attention to obesity and dietary control, there has been a rapid growth in the use of dietary assessment methods. Conventional, paper-based dietary records are being replaced by applications that run on computers or mobile phones [20]. Some of these applications also include photos of meals, taken with camera phones or digital cameras, to record a user's meals [5, 18, 12]. Such systems are popular since they make it easier for a user to record their meals, without entering long descriptions.

Most of the existing image-based dietary assessment systems use images only as an input for human observation. The user, and/or a dietitian, looks at the images to identify dietary trends. However, there has been a considerable amount of recent researches on image-based automatic dietary assessment systems [4, 19, 6, 7]. A few working and experimental systems are already available [14, 13, 1].

An image-based automatic dietary assessment system analyzes a meal image to estimate its dietary composition. One important task in this process is to detect image regions that correspond to food, prior to further analysis. This task, which we call *food region segmentation*, is difficult due to a number of reasons. Food regions, in general, have a wide range of colors, textures, and shapes, making it hard for a machine learning system to model them. A meal can consist of one or more dishes. The backgrounds and lighting conditions are diverse. Meal images are generally captured using small cameras or camera phones, and have a wide range of compression levels and sizes. These result in lower accuracies in food region segmentation.

In this paper, we propose to improve the accuracy of food region segmentation by incorporating minimal user interaction. We design a smartphone application that allows a user to interact with the meal photos that he/she recorded. The user taps on a few places of the photo displayed on the screen, to quickly specify the food items. The locations of these *touch points* facilitate more accurate food region segmentation.

The paper presents the following main contributions. It proposes a scheme to add touch points to a meal image and send it to a remote server for further processing. It also presents a segmentation algorithm based on the touch points and other image features. Finally, it evaluates the proposed algorithm using a manually labeled data set and presents the results.

2. RELATED WORK

Food region segmentation is a sub-topic of the widely researched area of image segmentation. There is a large amount of research on the latter, intended for various applications. Some recent works demonstrate that incorporating user interaction can facilitate much higher accuracy [8, 15, 9].

There has been several researches on segmenting images of food. Martin et al. [10] segment food regions in an image using RGB color features and Gabor texture features. Chang et al. [2] propose a two-stage approach for segmenting images with food items on a plain background. The algorithm initializes segments using contrast, and refines them using a boundary searching process based on color features. Mery

CEA'12, November 2, 2012, Nara, Japan.

Figure 1: Functional overview

et al. [11] use a similar approach for segmenting images of fruit. De Silva et al. [3] propose a watershed algorithm based on the *CIE l*a*b** color space, for food region segmentation prior to food classsification. Woo et al. [19] use a controlled background, together with photos taken before and after the meal, to segment consumed food from a given meal.

Most of the above approaches are designed for segmenting one food item. Some others require controlled or simple backgrounds. Such restrictions are not suitable for segmenting images of daily meals. So far, to our knowledge, there has been no research on interactive segmentation of meal images.

3. APPROACH

Figure 1 illustrates the functional overview of the proposed scheme. A smart phone user captures and views a meal photo on the screen. The user taps on the image, to enter a set of touch points on regions that show food. The phone sends the image and the touch points to a server. The segmentation algorithm, running at the server, uses the points to identify image regions correponding to food. A dietary information database retains the results for further analysis. The following sections describe these functions in detail.

4. USER INTERACTION

Our objective here is to achieve more accurate segmentation with minimal user intervention. We designed a smartphone application to enable user interaction with meal photos. Figure 2 outlines the user interaction with this application for submitting an image to be segmented. The application provides two methods for a user to select an image. The user can either take a photo (labeled ‘1’), or select a previously-captured photo (labeled ‘2’). After selecting the photo, the user taps on it to specify the touch points. The “Send” button at the bottom right of the screen composes an email message for sending these data the server. This message contains the original photo as an attachment. The application encodes the touch points into a character string, and inserts the string as the subject of the message. The string confirms to the format

$foodtag\ W_D\ H_D\ N\ x_1\ y_1\ ...\ x_N\ y_N$

The keyword *foodtag* is a preamble that allows the server to identify messages with meal images. W_D and H_D are the width and height of the image when displayed on the smartphone by the application. N is the number of touch points, and (x_i, y_i) are the touch points according to the coordinate system of the image on the smartphone screen. This format ensures that the locations of the touch points can be accurately decoded despite different devices and screen sizes.

The server receives the message and decodes the image and the touch points. It then initializes the segmentation algorithm described in the following section.

5. SEGMENTATION

The input to the segmentation algorithm is a meal image and a set of touch points. The size and quality of the image can vary, depending on the device that sends the image. The touch points are located within food regions, which are sets of connected pixels. It is not essential that a touch point is located at or near the center of the corresponding food region. Different food regions may or may not be connected to each other. The segmentation algorithm has to be robust under different combinations of conditions mentioned above.

Figure 3 outlines the proposed algorithm. We partition the image to a fixed number of blocks. The segmentation algorithm initializes a segment for each touch point, with the block that contains the touch point. Thereafter, the algorithm recursively evaluates neighbors of each segment, and decides whether it can be added to the segment. The decision is made by extracting a set of features and parameters from each neighbor and comparing them with the characteristics of the current segment. The algorithm terminates when all neighbors of all segments have been visited. The following subsections describe these functions in detail.

5.1 Initialization

We first partition the image in to 768 rectangular blocks, by dividing its longer side by 32 and the other side by 24. We selected a fixed number of blocks instead of fixed-sized blocks, for the following reason. Images from smart mobile devices have a wide range of resolutions, from 320 × 160 to 3200 × 2400. However, for a meal image, the scene size is much less variable. The algorithm initializes by creating segments with blocks that contain touch points. It also immediately adds a 3×3 square neighborhood of blocks to each segment, to speed up segmentation. The neighborhood size is approximately equal to that of a fingertip on a smartphone screen.

The algorithm also approximates the radius of the plate or container that holds the food item corresponding to the touch point. This radius is used to ensure that a food segment does not grow beyond the actual region. Figure 4 visualizes this estimation. Figure 4a shows a meal image with touch points. After performing edge detection on this image and removing very small edges (Figure 4b), we create the radial projection of the remaining pixels for each touch point, along the line connecting the touch point to its near-

Figure 3: Segmentation algorithm

Figure 4: Estimating the plate radius

est neighbor. In the absence of neighbors, the projection is calculated along the shortest path to the image boundary. The peak in this projection (Figure 4c), created by the edge of the plate, provides a rough approximation for the plate radius. While this pattern is less visible with plates of different shapes and off-centered touch points, experimental results show that the patterns are sufficient to make an approximation.

5.2 Feature selection and extraction

A typical meal image consists of regions corresponding to different types of food, crockery and cutlery, glassware, and background (trays, table surfaces etc). We analyzed a collection of approximately 32,000 meal images to identify characteristics that distinguish food regions from the rest. The following were the dominant characteristics of food regions:

- Colors that are quite similar to each other
- Approximately circular shape
- Presence of a large number of irregular edges

The algorithm extracts a set of features that represent the above characterstics. Color is represented using a three dimensional RGB color histogram of the region. A total of 512 bins ($8 \times 8 \times 8$ bins with 8 levels for each color) are used. The presence of irregular edges is represented by the number of small circles detected by applying Hough transform on the gray-scaled original image. Figure 5a shows two meal images. Figure 5b shows the centers of circles detected by applying Hough transform to these images, superimposed on the originals. It is evident that the density of circle centers represents the likelihood of a food region.

The segmentation algorithm compares the segments with neighboring blocks, using the above features. It then determines whether the block can be merged with a segment.

5.3 Feature matching and merging

After initialization and feature extraction, the following input data are available for the segmentation algorithm:

- A set of *initial segments*, each consisting of nine blocks
- The possible plate radius R_p of each segment
- The coordinates of circles detected within the image
- A set of *unlabeled blocks* that do not belong to any initial segment
- The RGB color histograms of the blocks

We propose an iterative algorithm to form segments using the above data. The following algorithm is applied to each initial segment.

1. Select a neighboring, unlabeled block b_i of the current segment p.
2. Calculate *block similarities* s^h_{ij} between b_i and each neighboring block b_j that already belongs to p, using the percentage overlap of color histograms as the similarity measure.
3. Calculate *maximum histogram similarity* s^h_m such that $s^h_{ij} \geq s^h_m$ for the majority of b_j.
4. Calculate the distance d_{ip} from the center of the segment to the center of b_i.
5. Calculate the overall similarity S_ip as
$$S_{ip} = s^h_m \left[1 + \alpha(d_{ip} - R_p)/R_p\right]$$
6. Calculate C_i, the number of circle centers in b_i.
7. If ($c_i > 0$ and $S_ip > \beta$) or ($c_i = 0$ and $S_ip > \gamma$), merge b_i to p. Otherwise, label b_i as background

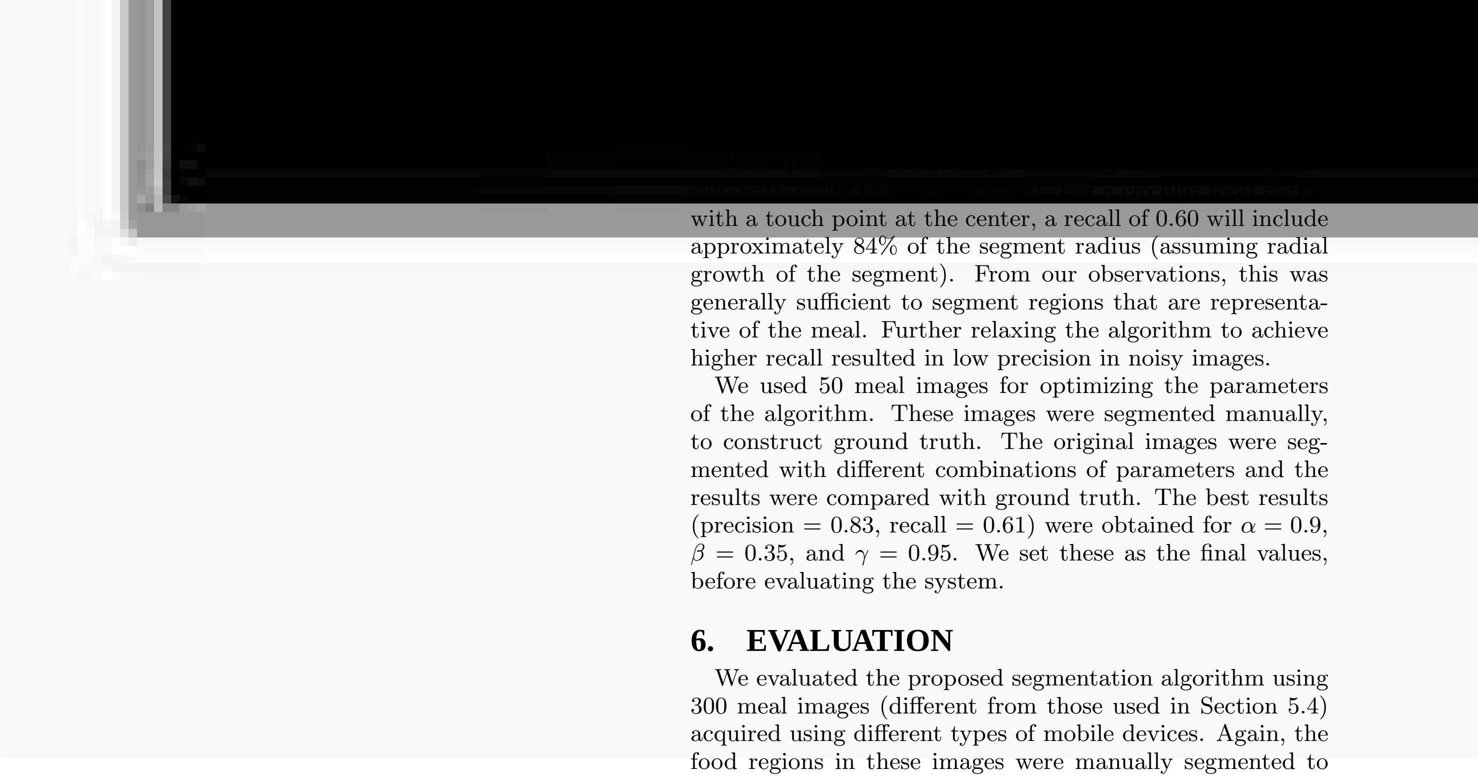

Figure 5: Results of circle detection on a meal image

8. Repeat steps 1 to 7 until all the neighboring block of p have been labeled.

Here, α is the weight of influence of R_p on block similarity, and β and γ are the respective similarity thresholds for presence and absence of circle centers. In case of a *collision* where a block is recognized as a member of multiple segments, the segment with maximum overall similarity that preserves connectedness is selected.

5.4 Parameter optimization

The performance of the algorithm varies according to the values of the parameters α, β and γ. The optimal values for these should be estimated to ensure that the algorithm produces good results on meal images from different devices, lighting conditions, and image compression levels.

In order to optimize the parameters, we first define the accuracy measures for segmenting meal images. Meal image segmentation can be treated as an information retrieval task where a collection of "food" regions are retrieved from a collection of image regions. Therefore, we can define precision P and recall R of segmentation as:

$$P = N_{TP}/(N_{TP} + N_{FP})$$

$$R = N_{TP}/(N_{TP} + N_{FN})$$

where N_{TP} is the no. of pixels correctly recognized as food, N_{FP} is the no. of pixels incorrectly recognized as food, and N_{FN} is the no. of pixels incorrectly recognized as non-food.

It should be noted that precision and recall do not have equal importance in this task; the original image has 100% recall. It is possible to achieve very high precision by retrieving small regions around touch points. However, this removes most of the food regions, and with them useful information for further analysis. Therefore, it is necessary to balance these two measures according to the purpose of segmentation.

with a touch point at the center, a recall of 0.60 will include approximately 84% of the segment radius (assuming radial growth of the segment). From our observations, this was generally sufficient to segment regions that are representative of the meal. Further relaxing the algorithm to achieve higher recall resulted in low precision in noisy images.

We used 50 meal images for optimizing the parameters of the algorithm. These images were segmented manually, to construct ground truth. The original images were segmented with different combinations of parameters and the results were compared with ground truth. The best results (precision = 0.83, recall = 0.61) were obtained for $\alpha = 0.9$, $\beta = 0.35$, and $\gamma = 0.95$. We set these as the final values, before evaluating the system.

6. EVALUATION

We evaluated the proposed segmentation algorithm using 300 meal images (different from those used in Section 5.4) acquired using different types of mobile devices. Again, the food regions in these images were manually segmented to create ground truth. Figure 6 shows some of the results from this evaluation. Images with touch points are shown on the left. The results of segmentation are shown on the right. The closed curves on the images indicate ground truth regions. It is evident that the results are generally accurate despite low image quality (Figures 6b and 6c), background clutter (Figure 6b) and the large number of plates (Figure 6c). In the presence of large food regions, the algorithm tends to return segments that are smaller than the actual regions. This is a result of distance based weights in the matching algorithm.

To evaluate the algorithm quantitatively, we calculated the precision and recall of segmentation for each image, according to the definitions in Section 5.4. For the set of 300 images, precision averaged 0.87 and recall 0.63.

We also compared the proposed method with a fully automated segmentation algorithm for meal images [16, 17]. This algorithm uses *GrabCut* [15] to segment the image. The seed for GrabCut is generated by creating the convex hull of the *star points* in the image. Star points in meal images occur mostly on food-plate boundaries. Therefore, the convex hull of such points is a reasonable, initial approximation for segmentation. Figure 7 outlines the functionality of this algorithm. Figure 7a shows the star points and their convex hull, superimposed on the image. It is evident that most of the points lie along the border of the food region and some lie on the border of the plate. Figure 7b shows the final result. While most of the background has been successfully removed, parts of the plate have been incorrectly segmented as food.

Average precision for fully automatic segmentation was 0.70, much lower than that for interactive segmentation. Average recall was 0.70. Figure 8 demonstrates the results of the comparison between the two approaches. Automatic segmentation, being a pixel-based method, returns smoother segment boundaries. However, it tends to misclassify parts of the background as food regions.

The proposed user interaction technique provides a fast method to provide valuable information for image segmentation. To measure the speed of entering touch points, one of the authors used a smartphone to mark touch points on

Figure 6: Example results

Figure 8: Comparison of segmentation algorithms

100 meal images. The average time consumed to mark touch points in an image was 2.6 seconds.

The meal images used in the experiments images presented challenging conditions for segmentation. Approximately 50% of the images had a resolution of 320×240 or less. Most of the larger images had lower compression quality. The other problems were improper exposure and the presence of noise. The use of a fixed number of blocks facilitated robust segmentation with different image sizes. With the user specifying food regions with touch points, compression quality caused less problems. The presence of noise and under-exposure caused over-segmentation, lowering precision.

7. CONCLUSION AND FUTURE WORK

We presented a scheme for user-assisted segmentation of meal images for dietary assessment. Touch-based interaction allowed fast acquisition of initialization points for segmenting food regions. Hierarchical segmentation based on color and texture allowed region growing on food regions with diverse shapes. Boundary estimation based on edge projection maintained precision of segmentation. An evaluation using 300 meal images demonstrated that the precision of segmentation is 0.87, compared to 0.70 using fully automatic segmentation.

The segmentation algorithm can be implemented on the smartphone itself, if further image analysis is done locally. The results of segmentation will be used as input for an algorithm that clusters large collection of meal images according to meal similarity.

8. ACKNOWLEDGMENTS

This research has been supported by a research grant from CREST-JST of Japan.

9. REFERENCES

[1] The Auri Group Tech Blog. NTT DoCoMo and KDDI; Keeping You Healthy. http://aurigroup.wordpress.com/2011/10/04/reporters-notebook-ceatec-2011-day-1/, 2011. [Online; accessed 6-July-2012].

[2] Y.-W. Chang and Y.-Y. Chen. An improve scheme of segmenting colour food image by robust algorithm. In *Proc. Algo2006*, 2006.

[3] L. C. de Silva, A. Pereira, and A. Punchihewa. Food classifications using color imaging. In *Proc. IVCNZ 2005*, pages 1–6.

[4] H. Hoashi, T. Joutou, and K. Yanai. Image recognition of 85 food categories by feature fusion. In *CEA*, 2010.

[5] Daily Burn Inc. Mealsnap. http://mealsnap.com, 2011. [Online; accessed 7-December-2011].

[6] K. Kitamura, T. Yamasaki, and K. Aizawa. Foodlog by analyzing food images. In *Proc. ACM Multimedia 2008*.

[7] K. Aizawa, G. C. de Silva, K. Kitamura, Y. Maruyama. Food Log: the Easiest Way to Capture and Archive What We Eat. Information Access for Personal Media Archives Workshop (IAPMA2010), pp.11-13.

[8] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23:303–308, 2004.

[9] D. Liu, Y. Xiong, L. G. Shapiro, and K. Pulli. Robust interactive image segmentation with automatic boundary refinement. In *ICIP*, pages 225–228, 2010.

[10] C. K. Martin, S. Kaya, and B. K. Gunturk. Quantification of food intake using food image analysis. In *Proc IEEE Eng Med Biol Soc*, pages 6869–72, 2009.

[11] D. Mery and F. Pedreschi. Segmentation of colour food images using a robust algorithm. *Journal of Food Engineering*, 66(3):353 – 360, 2005.

[12] MyPhotoDiet. My photo diet.

http://myphotodiet.com, 2006. [Online; accessed 7-December-2011].

[13] Electro Communications University of Japan. Foodimgbot. https://twitter.com/#!/foodimg_bot, 2011. [Online; accessed 7-December-2011].

[14] The University of Tokyo. Foodlog. http://foodlog.jp, 2008. [Online; accessed 6-July-2012].

[15] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004*, pages 309–314, 2004.

[16] H. Sugiyama, C. de Silva, and K. Aizawa. Food image segmentation using star points. In *MIRU 2011*. [In Japanese]

[17] H. Sugiyama, G. C. de Silva, and K. Aizawa Segmentation of Food Images by Local Extrema and GrabCut. In The Journal of The Institute of Image Information and Television Engineers, Japan. [In Japanese]

[18] Sukuta Systems. Photo diet - android market. https://market.android.com/details?id=sukuta.foodlog_en, 2011. [Online; accessed 7-December-2011].

[19] I. Woo, K. Ostmo, S. Kim, D. S. Ebert, E. J. Delp, and C. J. Boushey. Automatic portion estimation and visual refinement in mobile dietary assessment. In *Proc. Computational Imaging VIII*, volume 7533, page 75330, 2010.

[20] F. Zhu, A. Mariappan, C. J. Boushey, D. Kerr, K. D. Lutes, D. S. Ebert, and E. J. Delp. Technology-assisted dietary assessment. In *Computational Imaging*, 2008.

Figure 2: User interaction with the smartphone application

Figure 7: Fully automatic segmentation