Mobile Image Retrieval using Morphological Color Segmentation

David GAVILAN[†], Hiroki TAKAHASHI[‡], Suguru SAITO[†] and Masayuki NAKAJIMA[†]

[†] Graduate School of Information Science & Engineering, Tokyo Institute of Technology
2–12–1 W8–64 Ookayama Meguro-ku, Tokyo, 152–8552 Japan
E-mail: {david,suguru,nakajima}@img.cs.titech.ac.jp

[‡] Department of Human Communication, The University of Electro-Communications
1–5–1 Chofugaoka Chofu, Tokyo, 182–8585, Japan
E-mail: rocky@hc.uec.ac.jp

Abstract

An image retrieval method targeted to mobile devices is proposed in this paper. The method is based on an efficient segmentation method that obtains the most significant regions in the picture. These regions are found using a neural network-based color quantization over the morphological scale space of the images. The features of each segment are then used to index images in a database. The images also contain physical location context, by means of GPS, and the retrieval system is used as a part of an intelligent mobile interface. The retrieval method is compared with methods based on color histograms, showing the increase in speed, recall and precision, while maintaining interactive speeds in mobile devices.

Keywords: Image Retrieval, Mobile Applications, Color Segmentation

1 Introduction

The ever-growing collections of images available today create an increasing necessity for image retrieval systems. In recent years, cheaper memories and cheaper cameras have allowed the rapid expansion of digital imagery in mobile phones. Users take pictures almost daily and in a wide range of environments. While Computer Vision has solved a great variety of image classification and comparison problems in *narrow domains*, for *broad domains* the problem remains sparse. Broad domains have a large pictorial variety which is called the *sensory gap*, that is, the gap between the object in the world and the information in a description derived from a recording of that scene. For that reason, *search by association* is the most employed technique for broad domains [10].

Search by association usually requires highly interactive techniques. In order to decrease the feedback load from the user part, in this work we take two different countermeasures. The first one is to roughly classify images in basic categories to be able to get some of the advantages from *category search*, usually used to browse catalogs. The second measure is to improve the association capability by getting closer to the object level. This is accomplished by using scale space blobs that preserve geometric information.

The market of embedded devices has increased almost exponentially during the last decade. At the same time, the digital camera has become an inseparable part of a mobile phone. Applications have also been developed to make use of that camera, other than just storing pictures, such as 2D-codes readers [5], or even RPG games [20].

On the other hand, mobile search currently works with two different text-based approaches; messaging-based mobile search and browser-based mobile search. Messaging-based search is not intuitive to use because of the precise way that queries must be created, while browser-based search is more time consuming because it requires more user interaction. However, for most queries, the information we want to retrieve is more likely to be related to our current location. In such case, we can think of a single-click-based interaction, where the user press one button to take a picture, which triggers the information retrieval process. Moreover, locality will have a direct impact on the query size.

Our work contributes to the image retrieval field by providing a simple retrieval framework for mobile devices, along with its realization in a novel application. To the best of our knowledge, this is the first image retrieval system implemented in a mobile phone. Moreover, the image database naturally augments user's knowledge of the environment by just pressing one button in his phone, and thus transforming the so far passive camera into a means of interaction. For this purpose, we need to take into account some important requirements in image retrieval systems for mobile devices;

1. robustness against blurring, noise, and chromatic aberrations;

2. fast response, interactive speed;

3. clever search, that is, learn as much as possible from user's context (location, time, etc.).

In this work, the retrieval system relies on a color-based rough image segmentation algorithm based on the morphological scale space [1], robust under a wide variety of conditions. The image is then represented by a few region-based features, for fast and efficient retrieval. As opposed to other methods, our algorithm works with fixed parameters and works efficiently with local databases in a mobile framework, taking into account contextual features such as the GPS location to reduce the domain of the search.

This paper is organized as follows. In the next section, we will review some related work on image retrieval, as well as similar mobile applications. In Section 3 we will review our segmentation algorithm, while showing how it can be applied to low quality images. Then, in Section 4 we will present the structure of the database and how to access the images in it. In Section 5, the implementation of the system in a 3G mobile phone is described. Then, we will show some retrieval and speed results that will bring us to the conclusions of this work.

2 Related Work

The best well-known image retrieval system is probably the IBM's QBIC system [6]. This system works with color histograms and rough partitions of the image. However, the key for its success is the interactive query applet that allows the user to sketch his own queries. Although designed for broad domains, the best known working example works on the domain of an art gallery.

Kankanhalli et al.[11] proposed a combined color and spatial clustering algorithm to obtain regions whose features are used in image queries. However, they focus on searches on an iconographic narrow domain. A different approach for segmentation is shown in VisualSeek [19]. They use a backprojection algorithm to assign colors using a similarity measure between neighbors. Nevertheless, they rely on color histograms for representing each region, and we want to avoid heavy memory usage in our mobile application, in order to allow local processing.

The methods mentioned above either assume that we can find objects at the initial scale, or they do not care about objects at all. Works that use the scale space theory [13] for image retrieval try to overcome this. In the work of Mikolajczyk and Schmid, scale invariant interest points are used for indexing, without using color information [14]. The Blobworld system [2] proposes a new image representation trying to find regions which roughly correspond to objects, by defining a general-purpose image segmentation method. In this work, a similar but simpler segmentation algorithm, based on morphology and color categorization will be used, introduced in our previous work [9]. For a more detailed review on segmentation methods, please check the survey of Cheng et al. [3].

On the other hand, the main subtask of the framework presented in the introduction is to augment user's environment, providing, for instance, tourist information. Smart Sight [22] is a system that tries to do so. However, the system is not implemented in a mobile phone, but the user needs to carry a laptop computer in her bag, and wear an earphone set to give commands. Also, the information extracted from the environment depends on inefficient OCR techniques.

We believe that the touristic information database should be built by local agencies, and leave for the mobile terminal the task of optimizing the ways of querying such a database. Our system makes use of image categorization to select subtasks and limit queries on the sightseeing application. Moreover, the compact representation of the image using morphological color blobs minimizes the transmission load over the network.

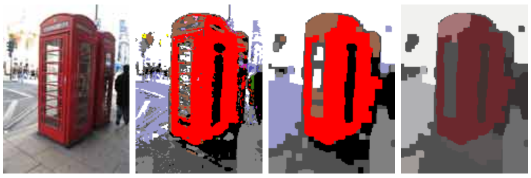


Figure 1: Segmentation example. Form left to right: original image; the result of color categorization; after applying the occlusion N-sieve; and the resulting regions represented by their mean color.

3 Color Segmentation

Blobs are low-level vision structures associated with objects in high-level vision. These structures appear in the linear scale space when convoluted with a Gaussian function [13]. Because of this isotropic diffusion, the geometric information is lost, and only information of size and position remains. To overcome this, in this paper the morphological scale space [1] will be used. By combining mathematical morphology we can keep the geometry of the objects, while the good properties of the scale space remain valid, such as causality, regularity, or affine invariance.

The segmentation algorithm is based on the *occlusion N-sieve* operator [9],

$$\mathscr{O}^r g = \mathop{{\bigcap}_{c}}_{i=1}^{n} \mathscr{N}^r g_i. \tag{1}$$

The operator is implemented by quantizing the image g in a few color categories using a neural network [7] and then applying N-sieves $\mathscr{N}^r$–a morphological closing followed by an opening– independently to each color category or image layer g_i. The resulting regions are merged together over a support background layer g_0 following the criteria of occlusion maximization c, and thus, preventing oversegmentation. Assuming a fixed color quantization network, the only parameter to set is the scale r, related to the minimum size of a relevant object.

Figure 1 displays an example image from the ICMU 2006 web site, the picture after being color quantized, and the final color regions. These regions will be stored in the image database and used for queries. The internal scale of all the images in the database is reduced by scaling the images to a thumbnail of 60×80 pixels, and the scale parameter r is empirically fixed to 3, corresponding to a square structuring element of 3×3.

All the images in the database have been taken with mobile phones with cameras with different resolutions, in the range of 300 Kpixels to 3.2 Mpixels. Notice that due to low quality lenses, chromatic aberrations and blurring may be present in the picture. Therefore, contours may not be clearly defined. However, our purpose is not to extract clean contours, but to locate meaningful regions that keep their original geometry as much as possible. Figure 2 shows some fireworks taken with a 5-year old mobile phone, at 120×160 pixels. Notice the appearance of green pixels and how they are ignored in the segmented image. Gray pixels do not belong to any region,

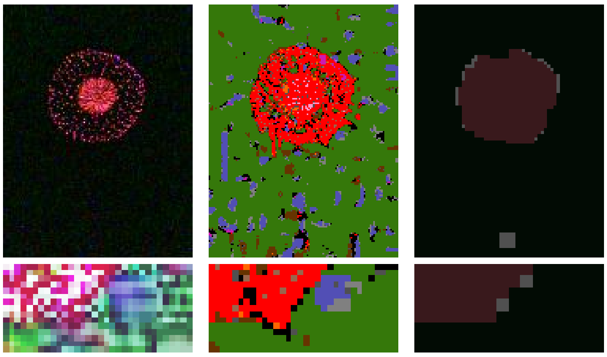


Figure 2: From left to right, original low resolution picture with noise, the result of color categorization, and the result of applying the occlusion sieve operator (showing the mean color of each region). A magnified detail of the picture is shown below. The color of the first zoomed image has been equalized to maintain color contrast when printing.

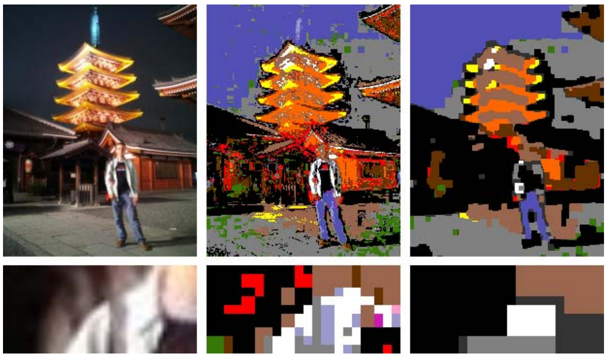


Figure 3: From left to right, original blurry picture, the result of color categorization, and the result of applying the occlusion sieve operator. A magnified detail of the picture is shown below.

but they are part of the support background layer g_0. On the other hand, Figure 3 shows a picture with blurred contours taken with a 3.2 Mpixel phone. Notice in the magnified parts how the blurring does not severely affect the already rough segmentation.

Finally, in Figure 4 several examples of segmentations are shown, taken in a wide range of environments with mobile phones. For details of the implementation of the occlusion N-sieve operator and a comparison with the mean-shift based segmentation [4], please refer to our previous work [9].

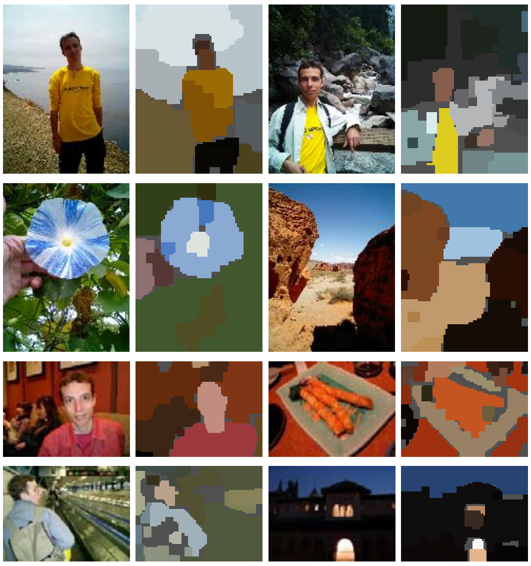


Figure 4: Segmentation examples of photographs taken under different lighting conditions –indoors, outdoors, at night, with backlight, etc.

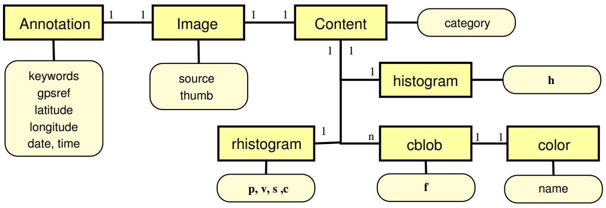


Figure 5: Database structure. f is the vector of features of every region; h contains the bins of the color histogram; and p, v, s, and c are the position, volume, shape and color of the region histogram.

4 Image Database Construction

Following the notation introduced in [21], we define an image as three different entities: source, annotation and content.

The contribution of this work to the automatic annotation of the image (semantics) is the categorization of the images in four categories and the categorization of each individual region in basic colors. The GPS location, date and time, as well as manually introduced keywords, are also part of the annotation. Contents are defined using the array of features of the blobs of the image. Figure 5 summarizes the schema of the image database, constructed using mySQL.

The web interface to the image database is a small Java applet that talks to some dynamic PHP documents, where the actual queries are performed. It allows loading image from disk or from the network, as well as drawing sketches in a small canvas. For the mobile application, a midlet[1] will be the interface to access the same documents, that will be rendered in lower resolution.

The web interface lets the user select individual blobs and search for similar objects in the database, or search for similar images. Similar images are defined using the composition of blobs in the picture, or its color histogram. Color categories used as excluding keywords, and another manually introduced keywords, can be used to restrict queries.

On the contrary, the mobile application is built under the philosophy of the single click. Therefore, by default all the blobs of the image are queried for scene matching, and the GPS location is used as a default “keyword”.

[1] *Midlet* is the name given to Java applications for embedded devices, while an *applet* is for the ones on web browsers.

4.1 Contextual Information

The context in which the user takes a picture can be used as a valuable information to restrict queries. The context is both the environment where the user is located, and user's intention. The environment can give us information such as the time of the day, the season of the year, or the geographical location. Such information can be obtained if the device provides such commodities. User's intention, however, refers to the purpose of taking the picture. Particularly, it is important to know whether she is interested in a person appearing in the picture, or the building behind that person. To approximate user's intention, we propose the use of image categorization.

4.1.1 Automatic Annotation: Image Categories.

Our image categories are defined in terms of task-oriented two-class classifiers. There are three main tasks we want to focus on. The first one is retrieving sightseeing information from a place, thus, it is important to distinguish between city views and natural scenes. The next task is to efficiently store the images in an album. It is important to distinguish between pictures of friends, and other pictures. Therefore, a portrait and non-portrait classifier is needed. The last task is related to reading text and codes from pictures. We need a classifier that tells us whether there is text or not in the picture. These three binary classifiers can be semantically expressed as four excluding basic image categories, by ordering them by task preference:

1. **Portrait**: Images that contain faces of relevant size;
2. **Text**: Images that contain a text region of relevant size;
3. **Artificial**: Scenes that contain mostly products of engineering, such as buildings, cars, etc.;
4. **Natural**: Scenes that contain natural objects, such as vegetation or food.

Notice that, when excluding text, this classification corresponds to the two major axes revealed in the experiments of [17], “human vs. non-human” and “natural vs. manmade”. Further subdivision to derive new semantic categories is possible, such as the set of 20 categories experimentally defined by Mojsilović and Rogowitz [16]. The learning of our four categories is done by training an intermediate backpropagation network with the features of each region, and a perceptron that uses the histogram of these to categorize the images [7].

4.1.2 Device Context: Using GPS.

When taking pictures with digital cameras, non-pictorial information is also stored with the picture. This *metadata* usually contains the configuration of the camera when the picture was taken, and the time. In some mobile phones, it is also possible to obtain the geographical location using the Global Positioning System (GPS), a satellite navigation system[2].

This information is important to restrict the queries for obtaining sightseeing information. If we take a picture of a building, the query will limit the scope to the *artificial* images around the area. The default range is 100 meters, value that corresponds to the positioning error. In our implementation, we just use the euclidean distance to the geographic coordinates, although it is also possible to extend the mySQL database with GIS extensions that allow spatial indexing. Moreover, the proposed system could be integrated with other location-based content search systems such as LocationWeb [18].

4.2 Indexing Image Contents

For every single region or color blob in the image, we extract position, shape and color features. These were used to train a neural network for image categorization. For the image retrieval system, the selected blob features are its center of gravity (x,y); its volume respect the image (v); its volume respect its bounding box (v_{bb}); its color, represented by the average, deviation and skewness on each band, (μ_c, σ_c, s_c), being c the band in the L*a*b* color model in this paper; and its shape, represented by an *elongation* measure $\mathscr{E}$, and its orientation θ. Every region is represented by a vector of these features f, and the image is represented by a set $\{f\}$. For later comparison, the image is also represented by a HSI color histogram of 166 bins; 18 bins for Hue, 3 for Saturation and 3 for Intensity, plus 4 additional gray levels, as proposed in VisualSeek [19]. These features, except for the shape, and a basic set of possible queries were presented in previous work [8].

In this work we introduce the *elongation* measure to indicate whether an object has a main growing direction, and it is defined as

$$\mathscr{E} = \min\left(\frac{|x'' - y''| + |x'y'|}{\max(x'', |x'y'|, y'')}, 1\right), \tag{2}$$

where x'', y'' and $x'y'$ are the second momentums of the object. The orientation of a region is given by

$$\theta = \operatorname{atan}\left(\frac{y'' - x'y'}{x'' - x'y'}\right). \tag{3}$$

The orientation distance d_o is then computed as

$$d_o = \frac{1}{2}\mathscr{E}_1\mathscr{E}_2(1 - \cos\alpha) + |\mathscr{E}_1 - \mathscr{E}_2| \tag{4}$$

where α is the angle between both regions, obtained with the equation

$$\cos\alpha = \cos\theta_1 \cdot \cos\theta_2 + \sin\theta_1 \cdot \sin\theta_2. \tag{5}$$

.

[2] Actually, most devices use the distances to antenna receivers to interpolate their position, instead of making real use of the satellite navigation system. The GPS system was designed by and is controlled by the United States Department of Defense.

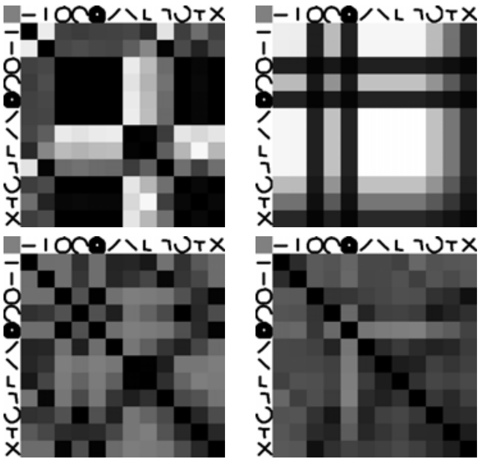


Figure 6: Distances between different sample objects. Left-right, up-down: angular distance $(1-\cos\alpha)$; elongation product $\mathcal{E}_1\mathcal{E}_2$; elongation and angle d_o; elongation, angle, and geometry.

The orientation is important to distinguish, for instance, the horizon from a vertical bar, assuming that the images are oriented parallel to the horizon. But it makes only sense for elongated regions, since for others there is no predominant orientation. The proposed orientation distance d_o tries to represent this fact. Figure 6 shows the table of distances between several sample regions; the darker the cell, the closer the objects. The table on top left shows only the angular distance. Because round regions end up with a default orientation of 45°, they match tilted lines in that direction. The last matrix shows all the combined features, except for color, showing the expected behavior.

In this work we also added a region histogram using the set of features $\{f\}$. This histogram contains nine spatial positions, five sizes, fourteen colors, and three shapes. Figure 7 shows the regions used as reference. Each segment of a segmented image is represented with the closest reference object. The histogram is then built by counting the number of pixels participating in a given position, shape, and color, and the number of regions of a given size, normalized by the image size and the total number of regions, respectively. For instance, the fourth picture in Figure 7 has a position histogram of nine bins, each valued 0.02, since there is a region of that relative size near each grid center. The shape histogram has a bin of $0.02\times 9 = 0.18$, the bin corresponding to the circle, and two zero bins, for the squares. The same value will appear in the white bin of the color histogram, since it's the only color in the drawing. As for the sizes, there are nine regions of the smallest size, and they are the total number of regions, so the bin representing that size will be valued 1, and zero for the rest.

Then, we define the distance between two images A and B as

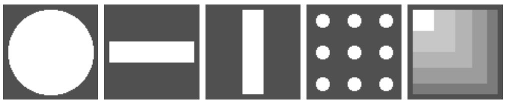


Figure 7: Reference objects for shape (circle and oriented rectangles), position (centers of a 3×3 grid), and size (5%, 20%, 40%, 60%, and 80% of the image size).

$$d_{image}(A,B) = w_p\langle h_p^A, h_p^B\rangle + w_s\langle h_s^A, h_s^B\rangle + w_c\langle h_c^A, h_c^B\rangle + w_v\langle h_v^A, h_v^B\rangle, \qquad (6)$$

where h_p^A is the position histogram of image A, h_s^A its shape histogram, h_c^A its color histogram, h_v^A its size histogram, $\langle\cdot\rangle$ is the euclidean distance, and w_p, w_s, w_c, and w_v are weights for each feature. In this paper, these weights are empirically set to 0.125, 0.125, 0.25, and 0.5, respectively.

For the web interface, an applet let us interact easily with the system. The applet is the same we introduced in previous work [8]. The system, with further explanations, can also be found online [3].

5 Low-level Vision on a 3G Mobile

To demonstrate the simple but efficient nature of our segmentation approach, we have implemented and tested the color blobs based algorithm for third generation (3G) mobile phones. The segmentation algorithm depends on the morphological scale space and the color categorization operator. Because the complexity of our N-Sieves implementation for indexed images is of $O(n^2)$, where n is the number of regions, we opted to apply only the color quantization step, and send the quantized image over the network.

The color categorization process depends on neural networks, having a sigmoidal activation function. Therefore, operations with real numbers are needed, which are not possible in most embedded CPUs. But since the process is static, it is also possible to store all the outputs for all the RGB color space in a look-up table. However, for true-color images the space is represented with 24 bits, so the table would have 16 million inputs. The maximum program size for the A5406CA phone is 150 KB, following the EZ Application Phase 3 specification [12], so this table can not be stored. Thus, we first map the image into a 16 bit color space, where the table will be just 64 KB. The quantization of an image of 80×60 pixels takes only 0.2 seconds, and it occupies only 2400 bytes, since two pixels can be encoded in a single byte (the color palette can be stored in 4 bits). This is then sent to the image server to process the query.

The image categorization process can also be stored as a look-up table. The categorization is used to automatically select tasks;

1) for *portraits*, the image is to be stored in the portrait folder;

[3] http://www.img.cs.titech.ac.jp/ ∼david/research.html.

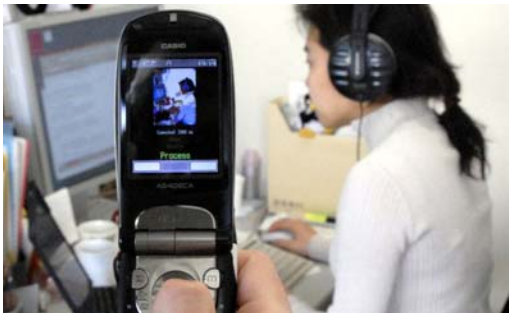


Figure 8: EZ mobile application for image retrieval. Color has been quantized and the image is to be sent to the database.

2) for *text*, the image is passed to the 2D code reader. If it fails, is stored selecting lossless compression (e.g. PNG format), or passed to a text recognition algorithm where available;

3) for *artificial* or *natural* pictures, the color quantized image, along with the GPS information, is sent to the database and a PHP page is rendered in the display.

The whole application, with the compressed look-up table, occupies 14 KB. Its interface is shown in Figure 8.

6 Results

In our experiments, we compare the retrieval capabilities of the region histogram against color histograms and our previous method. We use the *recall* and *precision* measures, defined as

$$recall = \frac{\#\{relevant \in \{retrieved\}\}}{\#\{relevant\}} \tag{7}$$

$$precision = \frac{\#\{relevant \in \{retrieved\}\}}{\#\{retrieved\}}. \tag{8}$$

The 1484 images in our database have been manually labeled with non-exclusive multiple keywords, that are used to tell which images resulting from a query are *relevant*. The selected keywords in our experiments are “toy”, ”flower”, ”portrait”, and ”building”, respectively having 9, 31, 122, and 136 relevant images. Figure 9 shows random samples of each category. To build the recall-precision graphs, every image from a given category is used as query, the precision computed for increasing recall values, and finally averaged by the total number of relevant images in that group.

Figure 10 shows the recall-precision graphs for every image category and for the four different methods tested. “Regions” refers to the distance between region histograms, as defined in equation (6). For the HSI color histogram, simply referred as “histogram” in the figure, the euclidean distance is used. These are compared with two composition-based metrics. One is the fuzzy logic query presented in Blobworld [2] and used in our previous work [8], defined as,

$$d_{image}(A,B) = \max_{i\in A}\{\min_{j\in B}(d_{blob}(i,j))\}, \tag{9}$$

where d_{blob} is the distance between each image segment and in this work also includes the orientation distance d_o. In the graphs, this query is referred as “maxmin”. The last type of query, referred in the graph as “summin”, is just the sum of minimum distances, that is,

$$d_{image}(A,B) = \sum_{i\in A}\min_{j\in B}(d_{blob}(i,j)). \tag{10}$$

In the graphs we can observe that the sum of minimum distances performs similar to the fuzzy query. Also, that our method gives better results for “broad domains”, that is, when not looking for a very specific object like the toy, in which the sum of minimum blob distances gives better precision in average. Color histograms also perform well in this case because the color of the object is constant across images (not so the background).

This means that it is difficult to tell apart particular objects using region histograms, but the method is good to tell big group categories apart, such as portraits or buildings. In fact, for particular objects, as discussed in previous experiments [8], the best approach it is to rely on user interaction: the user should select the most relevant target region, and start a query using regions distance. Figure 11 shows a query example for the “toy” object, where the user has manually selected the centered pink toy. Notice that by doing so it is easy to retrieve relevant images with completely different backgrounds.

In general, the precision of all queries is quite low. This is because in broad domains pictures are depicted in very varying ways. Even a photograph of the same object, taken at different times of the day or from different angles may easily alter the results of the query. This fact emphasizes the importance of the aid of contextual information such as the GPS location. For instance, the precision for the “building” query for all the methods triples if we restrict the query to images taken inside our university, since most of the buildings are plain gray and separated from each other. Moreover, by pre-classifying the images with an image categorization neural network, we can narrow the search easily in some cases.

Apart from the recall and precision, the speed of the queries has been also computed. Since the complexity of the fuzzy and sum queries is of $O(n^2)$, where n is the number of regions in all the database –$20,731$ objects–, they are the slowest methods. In our database, the average retrieval time of both types of queries is 3.57 minutes. In contrast, histograms retrieval time is linear to the number of images in the database –1484 images–, being the average retrieval time for color histograms 0.32 seconds, and 0.173 seconds for region histograms, the fastest of the methods. Figure 12 is a plot of the retrieval speed of the different methods for different database sizes, computed averaging random queries[4]. At the farthest point ($2^{10} = 1024$ images), our method, the fastest of the four, still responds at a fair speed. Since we can expect small queries when querying in context, that is, restricting by GPS location, the system can scale well for bigger databases.

[4] MySQL server v5, Debian Linux, CPU P-III 1GHz, 512 MB RAM

7 Conclusions and Future Work

In this paper, an efficient method for image segmentation using color categorization over the morphological scale space has been applied to image retrieval on a broad domain of digital photographs. The retrieval system has been implemented in a 3G mobile phone, in an application that aims to augment user's experience of the environment by providing local sightseeing information. The locality property of these devices, that is, GPS-capable phones, allows the reduction of the query load. Thus, the requirements for mobile image retrieval, robustness, efficiency, and locality, have been met.

Two new distance metrics have also been introduced. The first one is used to measure the orientation of elongated objects, and the second one is a region histogram used to represent the whole image. Using region histograms for queries results on faster and more precise queries for general images.

The proposed framework is compatible with other segmentation algorithms. However, the color categorization step adds knowledge and speed to the segmentation, while decreasing the load over the network. Subjects of future research include a further discussion on the choice of the color categories and its impact on different databases [15], a more precise image labeling [16], and the integration with GIS databases [18].

REFERENCES

[1] Andrew Bangham, Richard Harvey, Paul Ling, and Richard Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5(3):283–299, July 1996.

[2] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.

[3] H. D. Cheng, X. H. Jiang, Y. Sun, and Jing Li Wang. Color image segmentation: Advances & prospects. *Elsevier Science, Pattern Recognition*, 34(12):2259–2281, December 2001.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[5] DENSO Wave. QR code. ISO/IEC18004, June 2000. http://www.denso-wave.com/qrcode.

[6] M. Flickner, H. Sawhney, W. Niblack, and J. Ashley. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, September 1995.

[7] David Gavilan, Hiroki Takahashi, and Masayuki Nakajima. Image categorization using color blobs in a mobile environment. *Computer Graphics Forum (EG 2003)*, 22(3):427–432, September 2003.

[8] David Gavilan, Hiroki Takahashi, and Masayuki Nakajima. Color blobs-based image retrieval in broad domains. In *IWAIT*, pages 115–120, Jeju, Korea, January 2005.

[9] David Gavilan, Hiroki Takahashi, Suguru Saito, and Masayuki Nakajima. Morphological sieves on layered images for image segmentation. In *Proceedings of Nicograph International*, Seoul, June 2006.

[10] Th. Gevers and A.W.M Smeulders. Image search engines: An overview. In G. Medioni and S. B. Kang, editors, *Emerging Topics in Computer Vision*. Prentice Hall, 2004.

[11] Mohan S. Kankanhalli, Babu M. Mehtre, and Hock Yiung Huang. Color and spatial feature for content-based image retrieval. *Pattern Recognition Letters*, 20(1):109–118, January 1999.

[12] KDDI Profile. EZ Appli Phase 3, October 2003. http://www.au.kddi.com/ezfactory/-tec/spec/ezplus.html.

[13] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

[14] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, pages 525–531, 2001.

[15] Aleksandra Mojsilović. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image Progressing*, 14(5):690–699, 2005.

[16] Aleksandra Mojsilović and Bernice Rogowitz. Semantic metric for image library exploration. *IEEE Transactions on Multimedia*, 6(6):828–838, December 2004.

[17] Bernice E. Rogowitz, Thomas Frese, John R. Smith, Charles A. Bouman, and Edward Kalin. Perceptual image similarity experiments. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *SPIE*, 3299, San Jose, CA, January 1998.

[18] Misato Sasaki, Christian Noack, Hidetoshi Yokota, and Akira Idoue. Locationweb: Proposal and implementation of location-based web content search and creation using the mobile phone. In *2nd Int'l Conf. on Mobile Computing and Ubiquitous Networking*, 2005.

[19] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In ACM, editor, *Proceedings of the fourth ACM international conference on Multimedia*, pages 87 – 98, Boston, Massachusetts, United States, 1997. ACM Press.

[20] Square Enix. Final Fantasy VII: Before Crisis. NTT DoCoMo Foma i900, September 2004. http://www.square-enix.co.jp/mobile/bcff7.html.

[21] Mark G. L. M. van Doorn and Arjen P. de Vries. The psychology of multimedia databases. In *5th ACM Conference on Digital Libraries*, San Antonio, TX, USA, June 2-7 2000.

[22] Jie Yang, Weiyi Yang, Matthias Denecke, and Alex Waibel. Smart sight: A tourist assistant system. In *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 73. IEEE Computer Society, 1999.

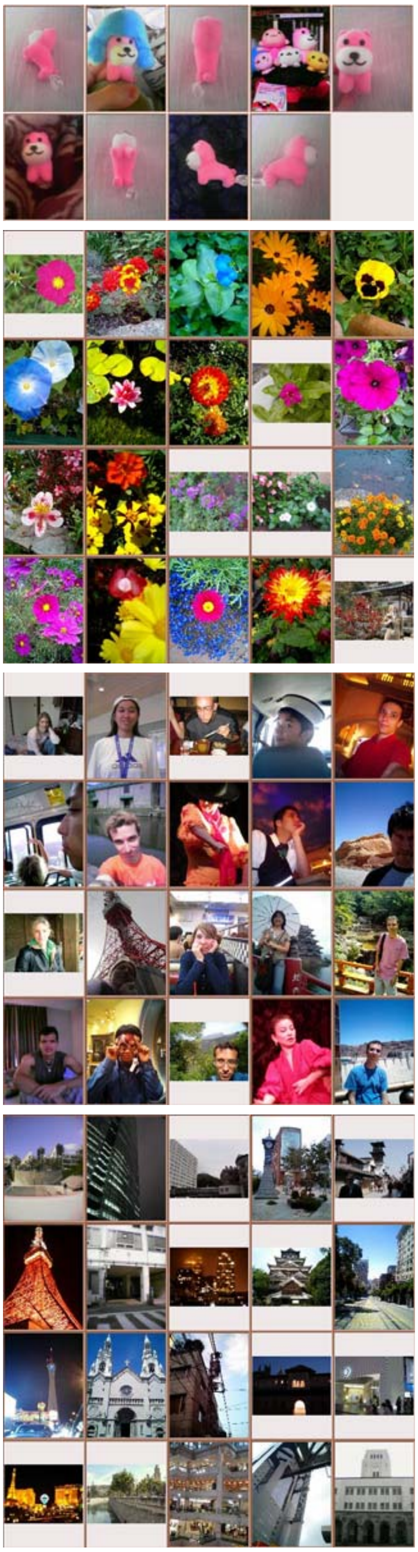


Figure 9: Sample images of the groups “toy”, ”flower”, ”portrait”, and ”building”, respectively.

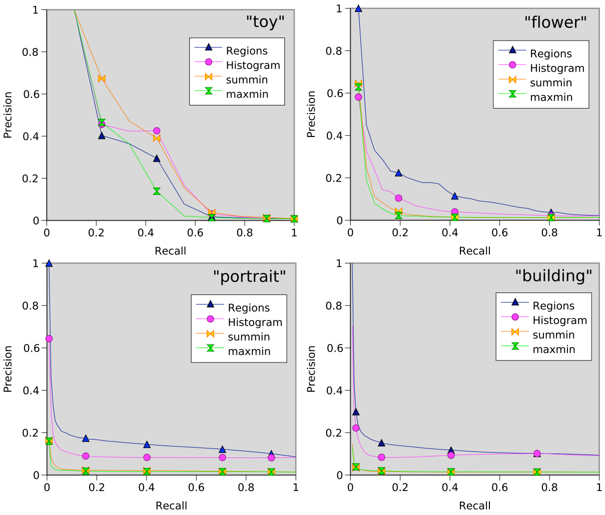
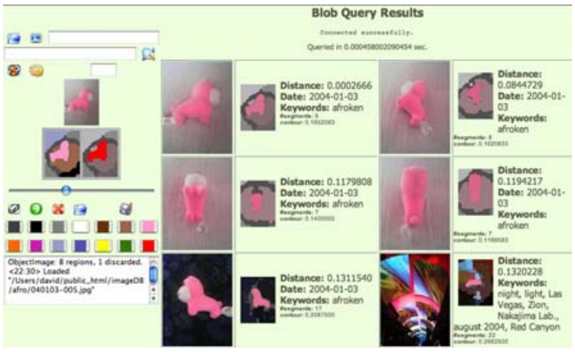


Figure 10: Recall-precision graphs.



Blob Query Results

Connected successfully.

Queried in 0.000458002090454 sec.

Distance: 0.0002666
Date: 2004-01-03
Keywords: afroken

Distance: 0.0844729
Date: 2004-01-03
Keywords: afroken

Distance: 0.1179808
Date: 2004-01-03
Keywords: afroken

Distance: 0.1194217
Date: 2004-01-03
Keywords: afroken

Distance: 0.1311540
Date: 2004-01-03
Keywords: afroken

Distance: 0.1320228
Keywords: night, light, Las Vegas, Zion, Nakajima Lab., august 2004, Red Canyon

```
ObjectImage: 8 regions, 1 discarded.
<22:30> Loaded
"/Users/david/public_html/imageDB
/afro/040103-005.jpg"
```

Figure 11: Interactive query example. The user selects the pink object and images with similar regions are retrieved.

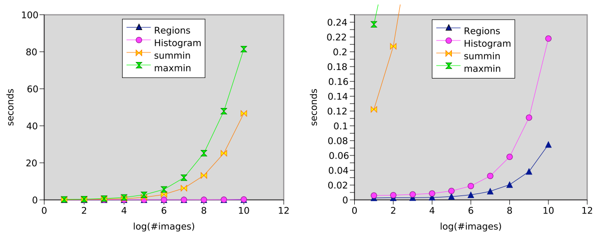


Figure 12: Retrieval speeds for different database sizes. The horizontal axis is the $\log_2$ of the number of images in the database, and the vertical one is the retrieval speed in seconds. The right graph is a scaled version of the left one.