

# SnapToTell: Ubiquitous Information Access from Camera

## A Picture-Driven Tourist Information Directory Service

Joo-Hwee Lim  
Institute for Infocomm  
Research  
21 Heng Mui Keng Terrace,  
Singapore 119613  
jooohwee@i2r.a-  
star.edu.sg

Jean-Pierre Chevallet  
IPAL-I2R Laboratory  
21 Heng Mui Keng Terrace,  
Singapore 119613  
viscjp@i2r.a-star.edu.sg

Siheem Nouarah Merah  
IPAL-I2R Laboratory  
21 Heng Mui Keng Terrace,  
Singapore 119613  
stumerahn@i2r.a-  
star.edu.sg

### ABSTRACT

With the proliferation of camera phones, many novel applications and services will emerge. In this paper, we present the SnapToTell system, which provides information directory service to tourists based on pictures taken by the camera phones and location information. We discuss key issues that motivate the design of the system and describe the system architecture. Next we present preliminary experimental results on scene recognition based on a realistic data set of scenes and locations in Singapore. Last but not least, we also discuss directions to be taken in the near future.

### Keywords

Mobile Information Retrieval, Picture-Driven Information Directory, Scene Recognition

## 1. INTRODUCTION

Imagine you are at a tourist spot looking at a beautiful lake or interesting monument. Instead of searching through your travel guide books to learn more about the scene, you snap a picture of the scene using your camera phone and send it to a service provider via Multimedia Messaging Service (MMS). Almost instantaneously, you receive an audio clip (MMS) or a text message (SMS) that provides you more information about the scene. You can continue to enjoy the scene while your fingers carry out this information retrieval task.

In a similar fashion, when you are in a museum and are eager to find out more about a master piece, you can snap a picture of the master piece (subject to the rules of the museum for taking pictures), send it to a service provider, and get the narration about the master piece. This kind of picture-driven information access scenario, known as *SnapToTell* (or

Snap2Tell) in this paper, is illustrated in Fig. 1.

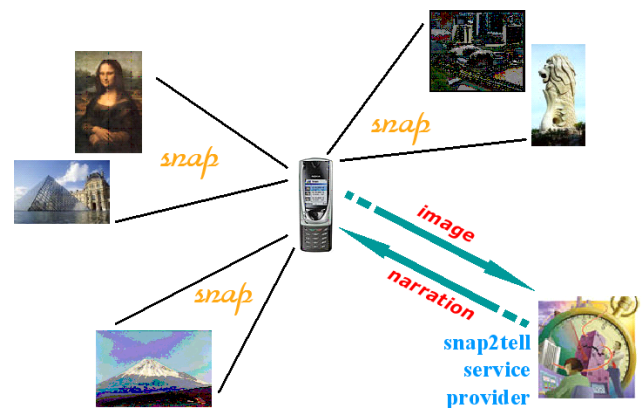


Figure 1: An application scenario of SnapToTell picture-driven information access

As the saying goes “A picture is worth thousand words”, a tourist can forget about the hassle of looking up scene description in a travel guide that distracts him/her from enjoying the scene or recalling the right name for the scene (assuming he/she knows what the scene is) to access a text-driven information directory. Moreover the charging of the on-demand service is more fine-grained and hence can tailor to the need of each tourist. Service providers can charge a fee for using this fun, easy-to-use and convenient picture-driven information directory, independent of the MMS charges.

Camera phones shipments, which already outnumber digital cameras worldwide, are expected to reach 298 million in 2007, according to a forecast released by IDC ([www.idc.com](http://www.idc.com)) in October 2003. In another market forecast, by 2008, 366 million of the 680 million (i.e. 53.8%) mobile phones sold will have cameras inside. Moreover, the quality of camera phones is constantly improving with recent models supporting 1 to 2 Mega pixels, optical zoom, built-in flash, etc. Hence it is likely that camera phones will dominate the lower end of the digital camera markets.

At the same time, the traditional handheld market will continue to consolidate and smart phones, devices that com-

bine the features of a mobile phone and handheld organizer, are expected to grow and dilute the handheld market (www.news.com). In fact, Sony is scaling back its CLIE handheld line and exit the U.S. and European markets this year.

In a nutshell, the authors feel that camera mobile phones will become pervasive personal devices beyond the role of traditional voice communication. With the built-in cameras and other improving features (e.g. more computing power, memory etc), many new applications and services would be made possible in the near future. In particular, innovative service such as the SnapToTell scenario described in this paper would greatly enhance the visual communication experience of the consumers.

From the technological point of view, obtaining location-based information is already possible with the GPS devices or the GSM cellular network infra-structure. However, knowing the location of a mobile phone user is not sufficient to determine what he or she is interested in (or looking at). The location-based information certainly helps to refine the user's context, but fails to capture his or her intention.

Using an ubiquitous camera phone to take a picture of a scene or an object of interest and use it as a query provides a very intuitive way to specify the information need even when the name of the scene or object is unknown to the user, which is crucial for human computer interaction on small devices. In return, the user will receive some audio or text information about the scene or object in real time. Sending the query as a text message is also possible provided the user is able to articulate the scene and the description is unambiguous. Speaking the text query into the mobile phone can relieve the burden of typing on small devices. However, voice recognition in a possibly outdoor noisy environment is still a technical challenge.

In this paper, we focus on using the SnapToTell framework for tourist information service i.e. a visual directory using real images as the input to look up relevant information about a scene. The SnapToTell framework can also be applied to other applications. For example, in the area of education, zoological students on a field trip can snap a picture of an insect to record what has been observed and to obtain more information about the insect almost instantaneously.

More generally, computer mobility has changed computer application and user needs and behavior. On the computing side, we have shift from the computer-centric paradigm (with one computer and many users) to the distributed computing paradigm (with one computer for each user). We are now moving into the ubiquitous computing paradigm in which the user is at the center of the scene and is connected with many computers. Moreover, the computers have shrunk in size into every day wearable objects and even tend to disappear into the surrounding.

Ubiquitous wearable devices also change how the user interacts with his or her computers. Using a small device like a mobile phone, or a handheld, forces software designers to revisit user interface design. Wireless communication also enables a real ubiquitous access to information. The mo-

bility of the user implies that the context, such as location information, should be taken into account to better service the user's information demand.

The application scenario proposed here is similar to the one in [5]. As described in [5], the client device used is a PDA system connected to internet through WLAN. It supposes that this wireless access point is installed in the area in which the system is going to work. The system includes an iPAQ 3870, a NexiCam PDA camera, an orientation sensor, and a GPS receiver. The position detection is ensured by a GPS attached to the PDA. However, the direction and tilt sensor is connected to the PDA via a laptop computer due to technical difficulty.

In our case, we have chosen a camera mobile phone which is a lighter and more ubiquitous device. The camera is integrated into the telecommunication device and localization is provided by the telecommunication operator. We believe that a camera phone is a better choice for communication than a PDA. Moreover, we plan to use the location features of the phone usually achieved by triangulation: receivers at separate locations intercept the signals from a mobile phone. Having measurements from three or more receivers can then determine the position of the object that emits the signal. The PDA system [5] requires a GPS device, orientation sensor, and WLAN connection. We think this solution is not realistic.

In the PDA prototype [5], the image taken from the connected camera together with GPS and orientation data are sent to a server. The server then runs the 3DMax program to generate a reference image from the same position and angle in a 3D model built in advance based on the GPS and orientation data. The matching is performed using detected line features. Only one building model has been constructed and tested in the paper though color segmentation has been explored for future experimentation.

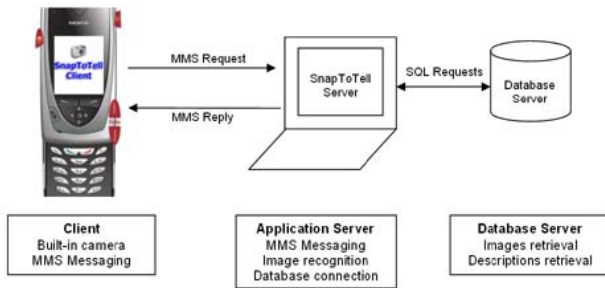
In SnapToTell, our approach of scene recognition is different. Instead of unnatural matching between a real image and a synthesized image from 3D model, our server will match the query image with different images of a scene, taken using different angles and positions. We think that 3D model construction is costly and not applicable to all kinds of scenes. We would exploit advanced invariant local features and matching in computer vision to accommodate variations in viewpoints, illuminations, scales, positions as well as to deal with problems of clutter and occlusion.

In essence, our scene indexing and matching strategy assumes that a set of images of the same object or scene have in common some characteristic recurrent local features that are discriminative enough to correctly detect the object among other possible ones in a given area. The location information about where the picture was taken reduces the search space and tends to simplify the problem as can be seen in our experiments.

## 2. SNAPTOTELL ARCHITECTURE

The SnapToTell framework is realized as a typical three-tier client/server architecture as depicted in Fig. 2. The client is a mobile phone with built-in camera that supports MMS,

such as the Nokia 7650 model used in our development and test. With the camera phone, a user can launch the SnapToTell application to send a request in the form of MMS to the application server. The MMS request is a picture of a real scene or object that information is sought.



**Figure 2: SnapToTell: a three-tier client/server architecture**

Upon receiving the MMS routed from the Multimedia Messaging Service Center (MMSC), the application server obtains the location information from the mobile network operator via triangulation based on a group of base stations [7]. If the mobile phone is equipped with a GPS receiver, the location information can also be sent from the client to the SnapToTell server. Although GPS is potentially the most accurate method to obtain location information, it has drawbacks such as higher cost, power consumption, warming-up, satellite visibility in urban area [1]. Hence we do not foresee GPS-equipped phones to become common in the near future.

With the location identified, the Snap2Tell server sends a SQL query to the database to retrieve the image meta-data for the scenes related to the location. The image meta-data of the query image is extracted and compared with image meta-data of the scenes by image matching algorithm. If the best matching score is above a certain threshold, scene descriptions of this best matched image is extracted from the scene database. Otherwise, a no match situation has occurred. In either case, the reply is formatted into an MMS message and sent to the mobile phone of the user who initiated the query via MMSC.

## 2.1 Client: Nokia 7650

The Nokia 7650 mobile phone uses the Nokia Series 60 platform (powered by Symbian OS v6.1), and is one of the earliest all-in-one device that combines mobile phone, digital camera and PDA. Nokia provides an emulator for each series that makes development and debugging more productive before deploying the application on the real device. Fig. 3 displays a sequence of screen shots for a running SnapToTell client which is written in C++ programming language.

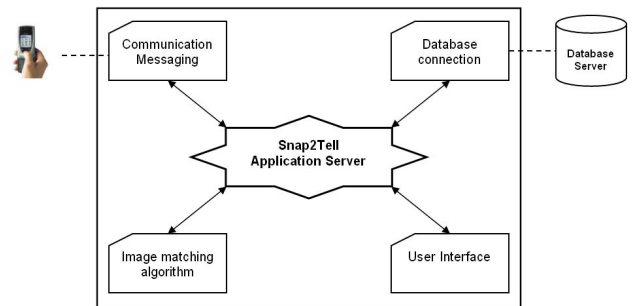
Following a top-down, left-to-right order, the first three screen shots shows the invocation of the SnapToTell application on Nokia 7650 phone. After the SnapToTell application is active, the user can start the camera to take a picture or open an existing image stored on the phone to used as the query as shown in the fourth screen shot. In this illustration, the user has chosen to select a stored image “SupremeCourt-16.jpg” as query (fifth screen shot on the second row) which

is displayed in the sixth screen shot. Note that if the user has decided to take a picture instead, the video camera will be turned on to allow the user see what the camera is focused at.

Once the user has selected or created a query image, he or she can scroll to the “Get Description” option to initiate a query. As described above, the query will be sent as a MMS to the SnapToTell application server (described below). Once a MMS reply is received from the SnapToTell application server, the user can play the MMS. As illustrated in the last screen shot in Fig. 3, the description is shown as text or/and audio.

## 2.2 SnapToTell Application Server

The SnapToTell application server is the functional core of the system and is developed in Java. As shown by the schematic diagram of Fig. 4, it has four components.



**Figure 4: Key components in SnapToTell application server**

The *Communication/Messaging* component handles the communication with the client device through MMS. As we shall see below, this component of the current prototype talks to the Nokia 7650 phone using Bluetooth interface. This short distance wireless communication protocol is used only for free testing, or for local indoor low cost connection with a distance internet server. We set up a generic communication layer that encapsulates the underlying protocol for ease of porting. The *Database Connection* component deals with all database access via SQL queries. The *Image Matching* component is responsible for the matching between a query image and a database image based on their image meta-data. It also needs to decide the best matches and handle no-match cases. Finally, SnapToTell server has a *User Interface* for administrator to configure the application server such as selection of database server to connect to, etc.

Once the application server has been configured properly, the administrator launches the server which is then in a listening mode, waiting for a client request to process.

## 2.3 Scene Database

Microsoft Access is used as the database server in the current prototype. Using Singapore as a test bed, we have divided the map into zones. A zone includes several locations, each of which may contain a number of scenes. A scene is characterized by images taken from different viewpoints, distances, and possibly lighting conditions. For example, in



Figure 3: Sample SnapToTell screens on Nokia 7650

the “South” zone of Singapore, we have the location “Sentosa”, where we admire the “Merlion” scene described using several images. Besides a location ID and image examples, a scene is associated with a text description, an audio description, and is assigned a category (e.g. “Monument”, “Art” etc).

Fig. 5 shows relationships among zone, location, scene, and category with examples. For Location 11: Chinatown in Zone 4, two scenes “Chinatown” and “Thian Hock Keng Temple” are shown. Three scenes labeled as “Indian National Monument”, “Supreme Court”, and “Sir Raffles Statue” are located in Location 14 of Zone 5.

In our current database, we have 6 zones, 15 locations, 78 scenes, and 390 images. On average, there are 2.5 locations per zone, 5.2 scenes per location, and 5 image examples to describe a scene.

## 2.4 Limitation and Enhancement

The current communication mechanism between the Nokia 7650 phone and the SnapToTell application server is based on the Bluetooth interface for two reasons. First, using local wireless connection during development and testing allows us to focus on the core algorithm issues without incurring necessary MMS charges. Meanwhile, a GSM modem that supports MMS is still not available in the market. Once it becomes available, the computer running the SnapToTell server would be able to receive MMS messages much like a mobile phone. That would be one step forward in our development plan.

We are not able to obtain realistic location information using the current development tools and the Nokia 7650 phone. The location information is currently simulated by a location ID sent by the SnapToTell client. For portability and

flexibility in the near future, we plan to rewrite the client with J2ME (Java 2, Micro Edition) and its optional package of Location API (JSR 179) [1]. JSR 179 requires the CLDC (Connected Limited Device Configuration) version 1.1 that supports floating-point numbers for representing coordinates and other measurements. We also require MMAPI (Mobile Media API) 1.1 to interface with the camera and WMA (Wireless Messaging API) 2.0 to send and receive MMS. As these specifications are relatively new, we have to acquire new mobile devices that support them when they become readily available. More details can be found at [developers.sun.com](http://developers.sun.com).

## 3. SCENE RECOGNITION

### 3.1 Empirical Study

Using the 390 images that we have collected for 78 scenes and organized into 15 locations, we have conducted an empirical study on scene recognition. For this initial study, we have adopted color histograms [9] to characterize and index the images. They are known to be invariant to translation and rotation about the viewing axis and change only slowly under change of angle of view, change of scale, and occlusion [9]. Hence we think that this a good starting point to see how effective they are for invariant scene recognition.

We have experimented with both global and local color histograms. There is a trade-off between content symmetry and spatial specificity. If we want images of similar semantics with different spatial arrangement (e.g. mirror images) to be treated as similar, we can have histograms of larger blocks (i.e. the extreme case will be a single block that covers the entire image, similar to the effect of a global histogram). However, spatial locations are sometimes important for discriminating more localized objects. Then local histograms will provide good sensitivity to spatial specificity. Furthermore, we can attach different weights to the blocks

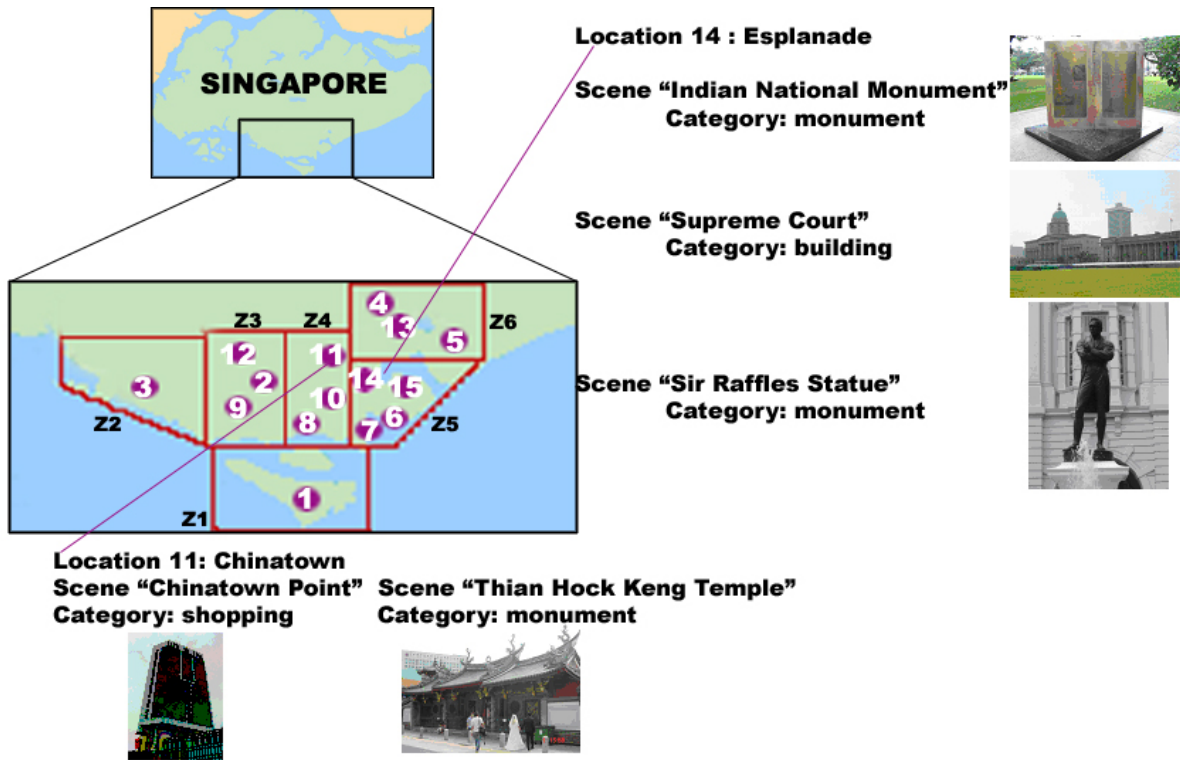


Figure 5: Locations and scenes in the database

to emphasize the focus of attention (e.g. center). That is, the similarity  $\lambda$  between a query  $q$  (with  $m$  local blocks  $Z_j$ ) and an image  $x$  (with  $m$  local blocks  $X_j$ ) is defined as:

$$\lambda(q, x) = \frac{\sum_j \omega_j \cdot \lambda(Z_j, X_j)}{\sum_k \omega_k}, \quad (1)$$

where  $\omega_j$  are weights, and  $\lambda(Z_j, X_j)$  is the similarity between two image blocks defined as

$$\lambda(Z_j, X_j) = 1 - \frac{1}{2} \sum_i |H_i(Z_j) - H_i(X_j)|. \quad (2)$$

Note that this similarity measure is equivalent to histogram intersection [9] between histograms  $H_i(Z_j)$  and  $H_i(X_j)$ .

We use color histograms in the Hue, Saturation, and Value (HSV) color space as it is found to be perceptually more uniform than the standard RGB color space [8]. The number of bins of a color histogram is  $b^3$  where  $b$  is the number of equal intervals in each of the H, S, and V dimensions. We also partition an image in identical blocks in both X-Y dimensions (i.e.  $K \times K$  grid). When two images are compared, we only compare the two local histograms of the corresponding blocks (Eq. (2)) with equal weights (eq. (1)). We have tested a maximum of  $K = 6$  in each dimension. That is, images are split into 36 blocks, and 36 histograms are computed in this case. Note that  $K = 1$  refers to global color histogram.

In our empirical study, we want to investigate the effect of location for reducing the search space and performance improvement. All experiments are performed using the 390

image test collection. As our data set is limited, we have adopted the leave-one-out methodology for evaluation. In all tests, each image of the test collection is considered as a query and is removed from the collection. This image is tested for histogram similarity matching against the rest of the collection: we compute histogram similarity between this query image and all other images, and we sort the similarities in descending order.

Nb of bins	Number of blocks			
	1 × 1	2 × 2	4 × 4	6 × 6
2	35.8	45.3	55.3	57.1
3	55.6	57.9	62	62
4	68.7	67.1	67.4	69.2
5	71.7	68.2	70	68.4
6	74.6	70.7	72.5	69.4
7	75.3	74.1	73	70.2
8	73.8	73	73.8	72.3
9	77.1	74.6	72.3	71
10	76.9	<b>75.8</b>	73.8	71.2
11	<b>78.9</b>	75.6	<b>75.8</b>	<b>73.3</b>
12	78.7	75.6	75.1	70.2
13	78.7	77.1	75.1	72.8

Figure 6: Scene precision for the closest image

In Fig. 6, we list the percentage of exact match. An exact match arises when the most similar image to the query image belongs to the same scene. In that case, the system has recognized the correct scene using one query image. When the number of histogram bins increases, the discrim-

ination power increases and hence the quality of the results increases too. At some point, more histogram bins may result in mismatch of the bins when slight change in the color distribution can cause shifts of pixel counts in adjacent bins. Without using any location information, the best precisions are mainly found at  $11^3$  (i.e. 1331) bins. Among these, global histograms have achieved the best precision value of 78.9%. Indeed, we shall see, global histograms seem to be the most effective for our 390 test collection in other experiments reported below when sufficient numbers of bins are deployed.

Nb of bins	Number of blocks			
	$1 \times 1$	$2 \times 2$	$4 \times 4$	$6 \times 6$
2	53	61.5	67.4	69.7
3	72.3	71.5	74.3	73
4	76.1	76.4	77.4	77.9
5	80.5	78.2	77.9	76.9
6	80.5	78.9	80	77.1
7	81.5	82	82	77.1
8	80.2	81.5	81	79.2
9	82.5	81.7	80.7	80
10	83.5	83.3	82.5	80.7
11	<b>84.6</b>	84.1	<b>83.8</b>	<b>81.5</b>
12	83.8	83	82.3	79.4
13	<b>84.6</b>	<b>84.3</b>	82.8	80.5

Figure 7: Scene precision using zone

Figure 7 shows the precisions for same exact match but the use of zone information. That is, we select the best matching image from the images that share the same zone as the query image. Clearly, we notice the enhancement over the previous figures. We also notice that the best results are moving toward more detailed histograms. As before, partitioning the images into smaller blocks for matching seems not very useful.

Nb of bins	Number of blocks			
	$1 \times 1$	$2 \times 2$	$4 \times 4$	$6 \times 6$
2	66.9	70.5	74.6	75.3
3	80.5	78.4	82.5	80.2
4	81.7	82.3	84.1	84.1
5	86.6	83.3	83.8	83.5
6	86.1	84.6	86.1	83.8
7	86.6	86.1	86.9	84.3
8	86.4	86.9	86.6	85.1
9	87.1	87.1	86.9	86.6
10	87.6	87.1	87.6	86.4
11	<b>89.2</b>	<b>88.4</b>	<b>88.4</b>	<b>87.1</b>
12	87.9	<b>88.4</b>	<b>88.4</b>	86.4
13	<b>89.2</b>	<b>88.4</b>	87.4	85.8

Figure 8: Scene precision using location

We have tested the use of more precise context information, which is the location constraint as shown in Fig. 8. Again we notice further enhancement in the results, and the distribution of the best results is similar to the previous table. Hence we conclude that global color histograms are most effective for our 390 real scene data set as long as enough

feature dimensions are adopted. The most likely explanation is that global color histograms are most invariant to translation and rotation about the viewing axis and change only slowly under change of angle of view, change of scale, and occlusion [9] for the purpose of scene recognition in our experiments.

Nb of bins	Number of blocks			
	$1 \times 1$	$2 \times 2$	$4 \times 4$	$6 \times 6$
2	13.1	16.9	17	17.4
3	21.4	20.5	20.7	19.9
4	24.4	23.1	22.6	22.3
5	26.4	24.9	23.9	23.3
6	28.6	25.7	25.4	24.8
7	<b>29</b>	26.6	26	25.1
8	28.4	26.8	26.9	26.3
9	28.7	<b>27.8</b>	27.2	26.4
10	28.8	27.2	26.5	25.8
11	28.8	27.7	<b>27.3</b>	26.2
12	28.1	26.5	26.3	25.2
13	28.7	27.6	27.1	<b>26.7</b>

Figure 9: Scene precision at full recall

Finally, we examine the results more globally. In Fig. 9, we have computed the precision at 100% of recall. It is in fact the ratio of images in the correct scene, on the total of images retrieved when all images of this scene are retrieved. The distribution of best performance is not the same as the previous tables, and the figures are much lower. In fact, the distribution of the results depends on the content of actual images. For some scenes, the image set is very homogeneous, and the whole set is retrieved at the top of the list. As for other scenes, the set is very heterogeneous because the pictures are taken differently with varying distances or view angles, hence these images are computed with very low matching values. Clearly in these situations, the color histogram approach is not discriminative enough.

### 3.2 Discriminative Local Features

Invariant object/scene recognition is an open problem in computer vision. Recently there have been a lot of interest in local invariant features [4, 6, 10]. In essence, local features are extracted from both a test image and a model image. They are characterized by invariant descriptors (e.g. generalized color moments [6]) and compared to generate a matching score for decision making. The feature extraction process and description are expected to be viewpoint and illumination invariant. Compared to global methods (e.g. global color histograms), local features are more tolerance to clutter and occlusion, thus removing the need for prior segmentation which is not robust for complex real scene images.

The authors have also explored local semantic features for image indexing and matching lately [3]. In particular, we are interested in discovering local semantic features that are recurrent within an image class and discriminative across classes of images [2]. It is beyond the scope of this paper to discuss the limitations of current local invariant feature approaches and possible enhancement for scene recognition. We are working on new local features and shall report the

results in the near future.

#### 4. CONCLUSION

In this paper, we have presented a framework with an experimental system for mobile and ubiquitous multimedia information retrieval. Our approach deals with real situation and real access device in order to measure the feasibility of such a system. It turns out that we have stretched the limit of currently available wearable technology, but we are convinced that ubiquitous computing is going to have rapid development in the very near future and we will soon be able to demonstrate the full system for real applications. We are also convinced that mobile information access, in particular context-aware image-based information access, will be a hot research and development topic.

The preliminary results we obtained using our small database in this paper has shown us that simple matching, based on color histograms, combined with localization information, is powerful enough to solve the image matching part. We are going to scale up the image database to measure the stability of our matching approach. In particular, we will study the effect of weather condition (i.e. varying illumination and noise for image capture), rejection criteria (i.e. when the query does not belong to any scene in the database), more powerful discriminative local features, ways to automate the construction of the scene database etc.

#### 5. REFERENCES

- [1] J2ME and Location-Based Services. <http://developers.sun.com/techttopics/mobility/apis/articles/location>.
- [2] J.H. Lim and J.S. Jin. Semantics discovery for image indexing. In Tomas Pajdla & Jiri Matas (Eds.), *Proc. of European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004*. Springer-Verlag, Germany, LNCS 3021, pp. 270–281, 2004.
- [3] J.H. Lim and J.S. Jin. Combining intra-image and inter-class semantics for consumer image retrieval. *Pattern Recognition* (accepted).
- [4] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision* (submitted).
- [5] W. Mai, G. Dodds, and C. Tweed. A PDA-based system for recognizing buildings from user-supplied images. In F. Crestani et al. (Eds.), *Mobile and Ubiquitous Information Access Workshop 2003*, LNCS 2594, pp. 143–157, 2004.
- [6] F. Mindru, Ti. Tuytelaars, L. Van Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94: 3–27, 2004.
- [7] Mobile Pipeline. <http://www.mobilepipeline.com>.
- [8] G. Paschos. Perceptually uniform color spaces for color texture analysis: an empirical evaluation. *IEEE Trans. on Image Processing*, 10(6): 932–937, 2001.
- [9] M.J. Swain and D.N. Ballard. Color indexing. *Intl. J. Computer Vision*, 7(1): 11–32, 1991.
- [10] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. *IEEE Computer Vision and Pattern Recognition*, vol. II, pp. 272–277, 2003.