Snap2Play: A Mixed-Reality Game based on Scene Identification

Tat-Jun Chin[1], Yilun You[1], Celine Coutrix[2], Joo-Hwee Lim[1],
Jean-Pierre Chevallet[1] and Laurence Nigay[2]

[1] Image Perception, Access and Language Lab
Institute for Infocomm Research, Singapore
{tjchin, ylyou, joohwee, viscjp}@i2r.a-star.edu.sg
[2] Laboratoire d'Informatique de Grenoble
Université Joseph Fourier, B.P. 53, 38041 Grenoble cedex 9 France
{celine.coutrix, laurence.nigay}@imag.fr

Abstract. The ubiquity of camera phones provides a convenient platform to develop immersive mixed-reality games. In this paper we introduce such a game which is loosely based on the popular card game "Memory", where players are asked to match a pair of identical cards among a set of overturned cards by revealing only two cards at a time. In our game, the players are asked to match a "physical card", which is an image of a scene in the real world, to a "digital card", which corresponds to a scene in a virtual world. The objective is to convey a mixed-reality sensation. Cards are matched with a scene identification engine which consists of multiple classifiers trained on previously collected images. We present our comprehensive overall game design, as well as implementation details and results. Additionally, we also describe how we constructed our scene identification engine and its performance.

1 Introduction

The increase in functionality and processing power of mobile phones in conjunction with the massive bandwidth afforded by advanced telecommunication networks allow a plethora of innovative and interesting game applications to be developed for mobile phone users, e.g. [1, 2]. In addition, the existence of high-resolution cameras on many mobile phones opens up another interesting direction for games based on image or video processing techniques.

In this paper we describe an innovative game application which exploits cameras on mobile phones. Our game consists of using the phone camera as an input interaction modality (a way for the user to interact with the game). The aim of our game, apart from the obvious role of providing entertainment, is to convey a feeling of "mixed reality" (i.e. a seamless interchange between a virtual world and the real world) to the user. Indeed our design approach is based on providing similar input modalities for interacting with the virtual world and with the real world. The game is inspired by the popular card game "Memory", where players are asked to match a pair of identical cards among a set of overturned cards by revealing only two cards at a time. In our game, the players are asked to match a "physical card", which represents the real world, to a "digital card", which is a token of the virtual world.

The game begins with the process of collecting a digital card: The player goes to a pre-determined location, and, with the aid of a custom software on the phone which exploits orientation sensing hardware, points the phone camera at a pre-determined direction and orientation (i.e. at a "virtual scene"). This is to simulate a photo taking experience in the virtual world, and it is emphasized that the corresponding real-world scene at which the player is inadvertently guided to aim is inconsequential for the game. Upon "snapping" the virtual scene at the right vantage point, the player receives the digital card which is actually an image of a real-world scene in a separate location.

Drawing from his familiarity of the local geography or guidance from the system, the player proceeds towards the location inferred from the digital card. Upon reaching the correct area, the player attempts to capture the scene with an image (the physical card) which resembles the digital card contents as closely as possible. The physical card is then transmitted to a service provider which employs a scene identification engine to verify the card. If verification is obtained, the player has successfully matched a pair of cards, and he can continue to collect the remaining card pairs (by first receiving directions to the next digital card). In a competitive setting, the player who collects all cards first is the winner.

We call this game "Snap2Play", and the overall success of the game is indicated when the player, apart from being entertained, is immersed during the interchange between the physical and virtual world i.e. he is only mildly aware of the differences between capturing the digital card and the physical card.

The rest of the paper is organized as follows: §2 describes in detail the rules and flow of the game. §3 describes a very important aspect of the game which is how we simulate a photo taking experience on a mobile phone for collecting the digital card. §4 explains the other major aspect of our game which is how our scene identification engine is constructed. §5 reports the progress we have obtained thus far, while a conclusion is drawn in §6.

2 How the Game Works

Snap2Play can be implemented as a single- or multiplayer (co-operative or competitive) game. The player interacts with the game through a mobile phone installed with the Snap2Play application. The overall flow of the game is coordinated through the mobile phone network (i.e. GPRS) by a game server (henceforth, the "game system"). Upon starting, the player is first introduced to the rules and objectives of the game, and is prompted to select his preferred game trail[3] based on the descriptions provided by the application. The player then

[3] The design of the game trail allows the incorporation of various vested interests, e.g. introduction to tourist spots or shopping precincts, promotion of physical fitness.

proceeds to hunt for the first digital card by receiving, from the game system through the phone network, a message with the rough location of the card. The aim is for him to enter the "Primary Search Area (PSA)" of the card; See Fig. 1.

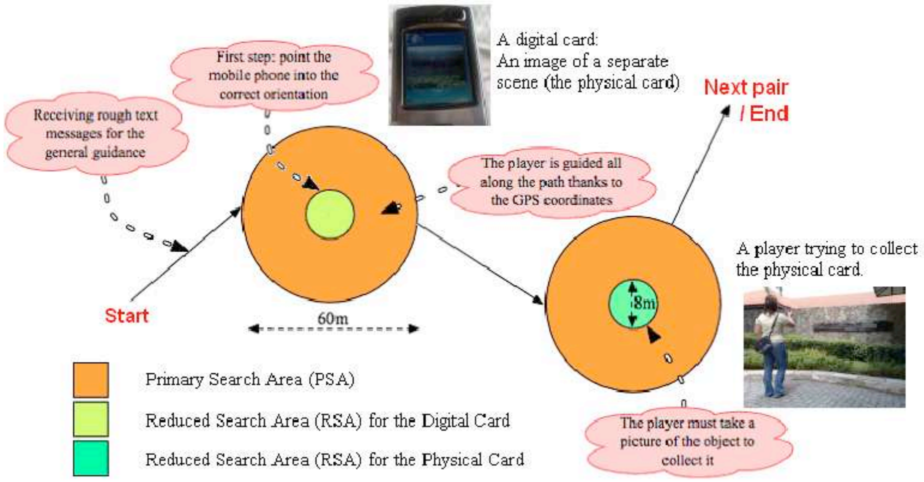

Receiving rough text messages for the general guidance
First step: point the mobile phone into the correct orientation
A digital card: An image of a separate scene (the physical card)
Next pair / End
The player is guided all along the path thanks to the GPS coordinates
A player trying to collect the physical card.
Start
60m
8m
The player must take a picture of the object to collect it
Primary Search Area (PSA)
Reduced Search Area (RSA) for the Digital Card
Reduced Search Area (RSA) for the Physical Card

Fig. 1. The Snap2Play game scenario.

The system will automatically detect when the player enters the PSA, after which a 2D digital compass which points to the direction of the digital card is activated on the phone. Based on the guidance by the compass, the player moves towards the digital card until he eventually reaches a "Reduced Search Area (RSA)". This is the pre-determined physical location, with a certain level of tolerance, where the digital card is embedded at a pre-defined orientation in space. The above steps are achieved through using GPS navigation.

Once the player enters the RSA, a text notification is given to activate the phone camera and to use it to locate and snap the digital card. The player is guided with a software that exploits attached orientation sensors on the phone to aim the camera at a pre-determined vantage point of the digital card. When the correct view-point is established, the player is notified through a superimposed image on the video feed or through tactile feedback, and the player can trigger the camera to collect the digital card. The elaborate manner in which the digital card is obtained is aimed at simulating the experience of capturing an image in a virtual world. §3 describes in detail how this is achieved.

The digital card is actually an image of a scene in a separate location. The player will need to find the location inferred from the digital card based on either his familiarity of the local geography or guidance from the system. When the player reaches the PSA of the physical card, the compass will be invoked to guide the player towards the corresponding RSA; See Fig. 1. The player proceeds by trying to snap an image of the scene (the physical card) to match the contents of the digital card. The physical card is sent (e.g. via MMS) to the system to be verified by a scene identification engine. §4 describes how the scene identification engine is constructed. If the cards are deemed matching, the player can continue to collect the remaining cards (by receiving a message with the rough location of the next digital card) if the game has not finished.

3 Collecting a Digital Card

The goal of this subsystem is to simulate a photo-taking experience in a virtual world. In essence the intention is to guide the player to take aim and acquire the image of a physically non-existent "virtual scene" (the digital card) while simultaneously imparting, as much as possible, the feeling of taking a photo of a real scene. A successful execution renders the player only mildly aware of the differences. The following describes how we propose to achieve this.

3.1 Embedding and Locating a Virtual Scene

Three parameters are required to embed a virtual scene in the physical world: A 2D coordinate (i.e. a GPS location), the direction (relative to the north) to face the scene, and the orientation (relative to the ground) to view the scene. The three different sensing instrument to achieve this, respectively a GPS receiver, a compass and a tri-axis accelerometer, are described in §3.2. When designing the game trail, the three parameters of all digital cards are recorded. Locating a digital card during play is a matter of re-producing its embedding parameters.

Acquiring the first parameter involves the straightforward process of reading from a GPS receiver and digital compass attached to the phone by the Snap2Play application. When locating a digital card during play, the system iteratively monitors the GPS location and heading of the player through the GPRS network and notifies the player when he is sufficiently close to a PSA or RSA (c.f. §2 and Fig. 1). Once the correct GPS coordinate is obtained, the system notifies the Snap2Play application to activate the video feed and begins to monitor the compass and accelerometer reading in order to provide guidance to the pre-determined "vantage point" (orientation) from which to view a virtual scene. Feedback to the player is given visually through the video feed, whereby a variable sized image of the digital card (to signify divergence from the correct orientation) is superimposed. Feedback via video is crucial to compel the player to acquire a digital card as though he is acquiring a physical card. Once the right orientation is obtained, the player triggers the phone to collect the digital card.

3.2 The Hardware

In our implementation, two standalone devices are required apart from the mobile phone. First is the Holux GPSlim 236 Bluetooth GPS receiver which contains a SiRF-Star-III chipset, allowing it to have an accuracy within 3m; See Fig. 2(a). This is digitally connected to the mobile phone via the Bluetooth protocol. Its small size allows the player to carry it conveniently in a pocket. Despite the impressive accuracy, the existence of high-rise buildings and frequently overcast sky in our game trails make GPS reading unrepeatable, hence the inclusion of the PSA and RSA (see §2) to allow some degree of tolerance of positioning error in the game. Note that more advanced phone models with on-board GPS receivers will render such an extra device unnecessary.

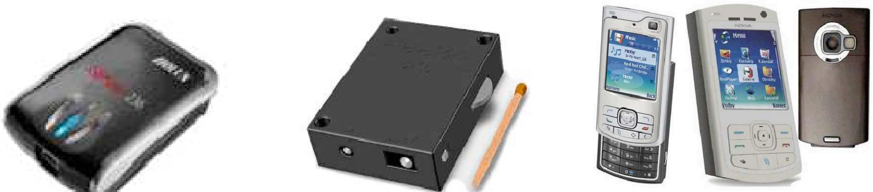

(a) Holux GPSlim 236 (b) SHAKE SK6 (c) Nokia N80

Fig. 2. Some of the hardware in our implementation.

In addition, a compass and a tri-axis accelerometer are also needed for the game. While the compass gives heading relative to the magnetic north, the tri-axis accelerometer provides orientation-sensitive readings. These functions can be conveniently provided by an instrument called the "Sensing Hardware Accessory for Kinaesthetic Expression (SHAKE)" device. The "SK6" model is used in our game. The accelerometer needs to be attached to the phone, and fortunately the small form factor of the SK6 allows this to be easily achieved; see Fig. 2(b). It is also digitally interfaced to the phone via the Bluetooth protocol. Finally, the mobile phone used in our system is the Nokia N80 model which has a 3 Megapixel camera. Fig. 2(c) shows the actual phone used in our system.

4 Collecting a Physical Card

The collection of a physical card requires the determination of, preferably without soliciting manual attention, whether a physical card (an image) corresponds to the scene at which a player should be seeking to match the current digital card he is holding. Many previous work on object and scene category recognition (e.g. [3–6]) provide excellent potential solutions. We take a *discriminative* machine learning approach for this task. Our scene identification engine consists of classifiers trained on previously collected sample images of the scenes. A discriminative approach is favoured here due to its simplicity, in that less parameter selection effort is required, and effectiveness, as is evident in [5, 6].

4.1 Multiple Classifiers for Scene identification

Our scene identification engine consists of N classifiers H_n, where N is the number of physical cards to be collected in a particular route, and $1 \leq n \leq N$. Given a physical card $\mathbf{I}$, the following result is obtained:

$$n^* = \arg\max_n H_n(\mathbf{I}) \ , \tag{1}$$

where $H_n(\mathbf{I})$ evaluates the possibility of $\mathbf{I}$ belonging to the n-th scene. Provided that $H_{n^*}(\mathbf{I})$ is larger than a pre-determined threshold, $\mathbf{I}$ is assigned the label n^*. Otherwise, the system is programmed to decline classification. If n^* matches the digital card that is currently pending, then $\mathbf{I}$ is successfully paired.

Before training H_n, a set of sample images of the N scenes have to be collected. Ideally, the samples should be captured in a manner that includes, as much as possible, the variations expected from images taken by the players as Fig. 3 illustrates. Secondly, the type of feature to be extracted from input images $\mathbf{I}$ on which H_n operates has to be determined. Experimental results from object and scene category classification [3–5] suggest that local features which are invariant to affine transformations are very suited for the task. In particular, the SIFT framework [7] has proven to be effective and robust against distortions caused by rotation, scaling, affine transformations and minor lighting changes. It involves two stages, namely keypoint detection and local descriptor extraction. Each descriptor is typically a 128-dimension feature vector which allows the the keypoint to be compared. Fig. 3 shows the SIFT keypoints of the sample images.

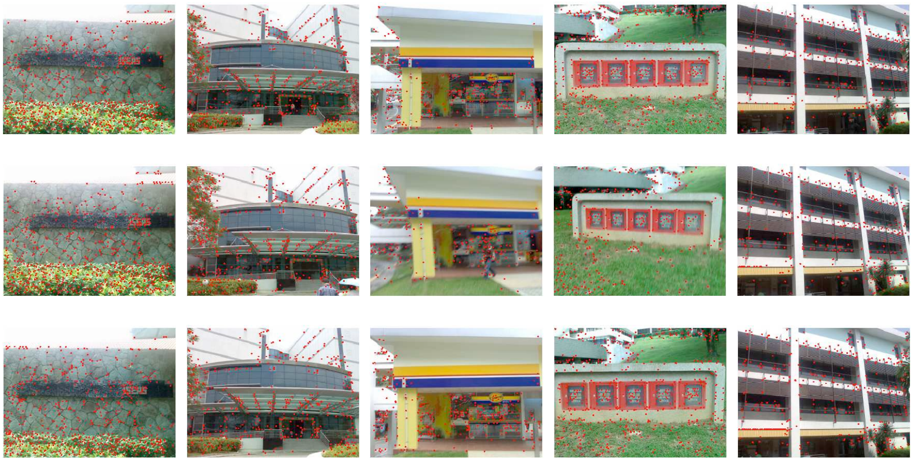


Fig. 3. Sample images of 5 physical cards in the Campus trail. Observe the variations that exist for a particular scene. Points in these images are detected SIFT keypoints.

4.2 Training Classifiers via Boosting

Many possibilities exist to construct H_n given a set of local features of $\mathbf{I}$. We apply the technique of *boosting* to train H_n. A popular boosting algorithm, the AdaBoost procedure, was adapted in [5] for object-class recognition with impressive results, and we apply their method here. Basically boosting constructs the desired classifier H_n by linearly combining T *weak* classifiers:

$$H_n(\mathbf{I}) = \frac{1}{\sum_{t=1}^{T} \alpha_n^t} \sum_{t=1}^{T} \alpha_n^t h_n^t(\mathbf{I}) \ , \tag{2}$$

where h_n^t is the t-th weak classifier of H_n and α_n^t is its corresponding weight ($\alpha_n^t \geq 0$). A weak classifier h_n^t determines whether $\mathbf{I}$ belongs to scene n. Each h_n^t must be able to classify correctly at least only half of the time (hence "weak classifiers"), but when their decisions are aggregated a competent overall classifier H_n can be obtained. For our task, a weak classifier is defined as

$$h_n^t(\mathbf{I}) = \begin{cases} 1 & \text{if } \min d(\mathbf{v}_n^t, \mathbf{v}^m) \leq \theta_n^t \text{ for all } 1 \leq m \leq M \\ 0 & \text{otherwise} \end{cases} \ , \tag{3}$$

where $\mathbf{v}_n^t$ is the defining feature of h_n^t, and $\mathbf{v}^m$ is one of the M SIFT descriptors of $\mathbf{I}$. Function $d(\cdot,\cdot)$ is a pre-determined distance metric (e.g. Euclidean) in the descriptor space, while constant θ_n^t is a closeness criterion for h_n^t. More intuitively, h_n^t is activated (returns '1') if there exists at least one descriptor in $\mathbf{I}$ that is sufficiently similar to $\mathbf{v}_n^t$. In addition, it can be seen that $0 \leq H_n \leq 1$.

For a particular scene, any feature extracted from its sample images can be used to define a weak classifier. The goal of AdaBoost is to select a subset of all available weak classifiers for which the defining features, as far as possible, simultaneously exist in the images of that scene while at the same time do not appear in images of other scenes. The algorithm iteratively chooses weak classifiers depending on how well they perform on sample images with different weightings, which are in turn computed based on how well previously selected weak classifiers perform on these samples. See Table 1. The closeness criterion for a weak classifier is provided by the weak hypothesis finder. See Table 2.

5 Results

5.1 Our system

We implemented Snap2Play as a single-tier Java Platform Micro-Edition (J2ME) client application. The application is designed using the "Interaction Complementarity Assignment, Redundancy and Equivalence (ICARE)" methodology [8], a component-based approach which enables the rapid development of multiple active or passive input modalities such as tactile messages, GPS and orientation sensory readings. Snap2Play requires an interface to capture and perform scene

```
Input: Training images $(\mathbf{I}_1, l_1), \ldots, (\mathbf{I}_K, l_K)$, where $l_k = +1$ if $\mathbf{I}_k$ belongs to class $n$,
and $l_k = -1$ otherwise.
Initialization: Set weights $w_1 = \ldots = w_K = 1$.
for $t = 1, \ldots, T$ do
    If $(\sum_k^K w_k < th_{Ada})$ set $T = t$ and terminate.
    Find best weak hypothesis $h_n^t$ with respect to $w_1, \ldots, w_K$ using the weak
    hypothesis finder (Table 2).
    Compute $\mathcal{E}_n^t = (\sum_{k=1,h_n^t(\mathbf{I}_k)\neq l_k}^K w_k)/(\sum_{k=1}^K w_k)$.
    Compute $\beta_n^t = \sqrt{(1-\mathcal{E}_n^t)/\mathcal{E}_n^t}$ and $\alpha_n^t = \ln \beta_n^t$.
    Update $w_k \leftarrow w_k \cdot (\beta_n^t)^{-l_k \cdot h_n^t(\mathbf{I}_k)}$.
end for
Output: $T$ weak classifiers $h_n^t$ with corresponding weights $\alpha_n^t$.
```

Table 1. The AdaBoost algorithm for scene classification.

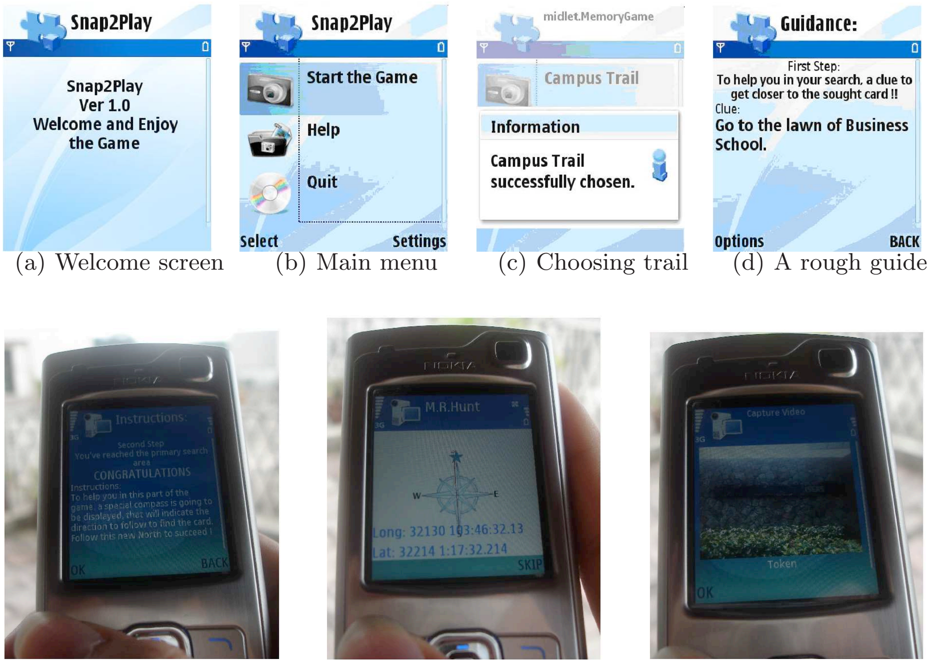
```
Input: Labeled local features $(\mathbf{v}_k^f, l_k)$ and weights $w_k$, where $1 \leq k \leq K$ and $1 \leq f \leq F_k$.
$F_k$ is the total number of local features extracted from the $k$-th image.
1. Define distance metric $d(\cdot,\cdot)$ for local features.
2. For all local features $\mathbf{v}_k^f$ and all images $\mathbf{I}_j$, find the minimal distance between $\mathbf{v}_k^f$
   and local features in $\mathbf{I}_j$:
       $d_{k,f,j} = \arg\min_{1\leq g\leq F_j} d(\mathbf{v}_k^f, \mathbf{v}_j^g)$.
3. Sort the minimal distances as such: For all $(k, f)$, find a permutation
   $\pi_{k,f}(1), \ldots, \pi_{k,f}(K)$ such that
       $d_{k,f,\pi_{k,f}(1)} \leq \ldots \leq d_{k,f,\pi_{k,f}(K)}$.
4. Select the best weak hypothesis as such: For all $v_k^f$, compute the following value
       $\arg\max_s \sum_{j=1}^s w_{\pi_{k,f}(j)} l_{\pi_{k,f}(j)}$
   and select $\mathbf{v}_k^f$ for which the above value is the largest.
5. Compute threshold for best weak hypothesis as
       $\theta = \frac{1}{2}(d_{k,f,\pi_{k,f}(s^*)} + d_{k,f,\pi_{k,f}(s^*+1)})$
   where $s^*$ is the value of $s$ at the maximum in Step 4.
Output: Best weak classifier defined by the selected $\mathbf{v}_k^f$ and its corresponding $\theta$.
```

Table 2. The weak hypothesis finder.

identification on the physical card. We built an interface with a client-server architecture which allows transmission of the image to the game server via GPRS or WiFi. The communication layer of the our interface is developed using Java Platform Standard-Edition (J2SE) and the scene identification engine is built in C++ for performance. Fig. 4 illustrates the actual user interface on the mobile phone of our system.

5.2 Scene Identification

We have implemented our game on a trail which is situated in our campus (henceforth, the "Campus" trail). This trail contains 10 pairs of digital and physical cards. Fig. 3 illustrates 5 of the 10 physical cards, while Fig. 5 depicts



Campus Trail successfully chosen.

First Step:
To help you in your search, a clue to get closer to the sought card !!
Clue:
Go to the lawn of Business School.

Second Step
You've reached the primary search area
CONGRATULATIONS
Instructions:
To help you in this part of the game, a special compass is going to be displayed, that will indicate the direction to follow to find the card. Follow this new North to succeed !

Long: 32130 103:46:32.13
Lat: 32214 1:17:32.214

(a) Welcome screen (b) Main menu (c) Choosing trail (d) A rough guide

(e) Entering a PSA (f) Our compass (g) A digital card

Fig. 4. Sample screen shots of the user interface on the Nokia N80 of our system.

the configuration of the trail. We present scene identification results for the physical cards (scenes) in this trail. A total of 183 images were used for training the scene identification engine, while 300 images captured on a different day were used for testing. As shown in Fig. 3, the images were captured in a manner that aims to include the variations expected during game play. Several scene identification methods are compared. First, given the apparent visual similarity of the images from the same class despite the view-point changes imposed during collection, one is tempted to apply a straightforward Euclidean distance with a nearest neighbour search to classify the testing images. Secondly, we also tried a naive feature matching approach i.e. store all SIFT features from training images and test a query image by matching its features to the feature database. We have also implemented the "discriminative patch selection" method of [6] since it provides an alternative sampling-based approach to boosting [5]. Fig. 5 illustrates the scene identification results in the form of Receiver Operating Characteristic (ROC) curves. It can be seen that despite the vast similarity between the images, trivial approaches like Euclidean distance and naive feature matching do not perform well. Boosting returned the best results, justifying our selection of this method for our game. On the query set it is capable of achieving a true accept rate of 99.33% with a corresponding false accept rate of only 0.67%.

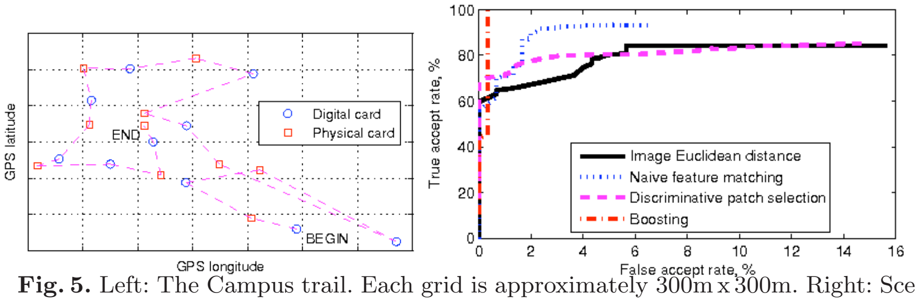


Fig. 5. Left: The Campus trail. Each grid is approximately 300m x 300m. Right: Scene identification results (in the form of ROC curves) of 4 methods on the Campus trail.

6 Conclusion

In this paper we propose a novel mixed-reality game which is implemented on mobile camera phones. The game requires the player to match a physical card in the real world to a digital card in a virtual world. This was achieved by exploiting orientation sensing hardware and a scene identification engine. Experimental results show that boosting SIFT features is an effective technique for constructing the scene identification engine. The next step is to deploy Snap2Play to evaluate the usability of the interaction modalities as well as the experience of the players (in Singapore in September 2007 and in Grenoble-France during autumn).

Acknowledgements

We reserve enormous gratitude to Emelie Annweiler, Wee-Siang Tan, Kelvin Kang-Wei Goh and Clement Yuan-Loong Tung for their tireless effort in implementing the various aspects of our game. This work is partly funded by the ICT-Asia MoSAIC project and by the OpenInterface European FP6 STREP focusing on an open source platform for multimodality (FP6-035182).

References

1. Strachan, S., Williamson, J., Murray-Smith, R.: Show me the way to monte carlo: density-based trajectory navigation. In: Proceedings of ACM SIG CHI. (2007)
2. Ballagas, R.A., Kratz, S.G., Borchers, J., Yu, E., Walz, S.P., Fuhr, C.O., Hovestadt, L., Tann, M.: REXplorer: a mobile, pervasive spell-casting game for tourists. In: CHI '07 Extended Abstracts on Human Factors in Computing Systems. (2007)
3. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Computer Vision and Pattern Recognition. (2003)
4. Li, F.F., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: Computer Vision and Pattern Recognition. (2005)
5. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. Pattern Analysis and Machine Intelligence **28**(3) (2006) 416–431
6. Lim, J.H., Chevallet, J.P., Gao, S.: Scene identification using discriminative patterns. In: International Conference on Pattern Recognition. (2006)
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
8. Bouchet, J., Nigay, L.: ICARE: A component-based approach for the design and development of multimodal interfaces. In: Extended Abstracts of CHI'04. (2004)