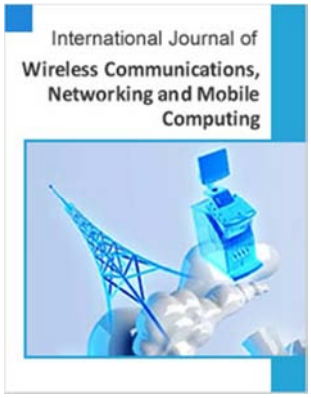
International Journal of Wireless Communications, Networking and Mobile Computing
2017; 4(1): 1-15
http://www.aascit.org/journal/wcnmc
ISSN: 2381-1137 (Print); ISSN: 2381-1145 (Online)

American Association for Science and Technology



Keywords
Photographic Treasure Hunt,
Instance Recognition,
Computer Vision,
Image Content Analysis,
Mobile CBIR

Received: March 28, 2017
Accepted: April 18, 2017
Published: June9, 2017

GeoPhotoHunt: A Mobile Application for Playing Photo Treasure Hunts and Suggesting Tourist Tours

Alessandra Lumini

Department of Computer Science and Engineering, University of Bologna, Cesena, Italy

Email address
alessandra.lumini@unibo.it

Citation
Alessandra Lumini. GeoPhotoHunt: A Mobile Application for Playing Photo Treasure Hunts and Suggesting Tourist Tours. *International Journal of Wireless Communications, Networking and Mobile Computing*. Vol. 4, No. 1, 2017, pp. 1-15.

Abstract
This paper presents a novel mobile application for the implementation on Android mobile devices of a photo treasure hunt. GeoPhotoHunt (GPH) provides services for creating/playing a photographic treasure hunt as a guide for city visiting or for recreational purposes. The application uses computer vision algorithms executed onboard to quantify similarity between the picture captured by the phone camera and the correct image, without requiring a remote server and an internet connection: this is a main requirement for tourists who often pay a lot of money for international data roaming. Both creation and playing phases are performed on the mobile device in order to facilitate the use on site. The image recognition module on the mobile device manages the feature extraction and the calculation of similarity to the stored objects. In this paper we report a rather broad analysis of existing works in the domains of location-based pervasive games applications and image recognition using limited resources. Moreover, in order to select the best image similarity method, we perform a wide comparison of descriptors used to represent images and measures used to evaluate their similarity. Our experiments carried out on 7 image datasets prove that the system is efficient and robust in comparing images with different characteristics.

1. Introduction

A photo treasure hunt is an adventure game where the players have to find locations or places using a sequence of clues and provide a picture to verify the acquisition of the treasure.

A photo treasure hunt played with the support of a digital device is a pervasive game [1], where the game experience is extended into the real world thanks to mobile technologies and may change on the basis of a player's position and actions. As a treasure hunt is location-dependent, it can also be classified as a location-based game [2], where players move around the territory and retrieve clues by visiting certain places in order to fulfil the game's requirements.

During the past years, location-aware and pervasive games have obtained growing success, and several new outdoor games have been released. Geocaching [3], for example, is an outdoor treasure-hunting game in which the players use GPS to seek containers (called "geocaches") previously hidden somewhere in the world.

Geocatching[1], Opencatching,[2] etc.). A similar and older game is Letterboxing, an outdoor game which combines elements of orienteering, art, and problem-solving. Treasure trails are a variation on the theme of a treasure hunt, in which participants follow a set of directions and discover clues to help to address a riddle.

Another class of mobile applications that exploit map navigation are tourism applications, designed to offer a guided tour of a city. Planning a travel itinerary is a hard task for tourists, who usually benefit from different sources of information, such as maps, travel guides, travel sites, and blogs. Several applications have been recently designed to help tourists in planning a complete city tour that best covers major points of interest in the time planned for travel (i.e. VISITO Tuscany [4], WhaiWhai, Picture Geo Hunt).

In this paper a novel application for the development of a photo treasure hunt game for mobile devices is presented, which is substantially different from all existing application.

Differently from all the above applications, where the clues that guide the search in general consist of the GPS coordinates of the place where the tag to be discovered is hidden and the achievement of the goal is simply reaching the designated place (without any supervised control), GPH is a photographic treasure hunt that requires users to take a picture to complete a task. As urban games are becoming more and more popular in Italy and worldwide, we design an application to manage an urban game played by taking photos of remarkable places. In such a game the presence of the player in the given location is not enough: the player has to take a photo of a given landmark to prove that he/she has solved the clue. GPH can be used to design city tours where a tourist is invited to walk along a path (possibly optimized by the designer) in order to discover the main points of interest in a city.

GPH implements on mobile devices a photographic treasure hunt and makes available the following services:

a) Creation and implementation of a photographic treasure hunt consisting of a sequence of textual or visual clues to which the player is called to answer in order to solve the present clue and access the next clue. Each clues is based on a riddle, a question, or some simple wordplay to be solved in order to reach a place on a map. The solution is in the form of the geographic coordinates of the place to be reached and a picture of the subject to be photographed. The solution can be complemented by tourist information to make the visit more interesting.
b) Playing the game using a mobile device (possibly not connected to the internet). During the photo treasure hunt, the player uses his/her mobile device to answer the clues providing solutions in the form of geographic coordinates detected by the GPS sensor and pictures taken by the device camera. The system verifies the accuracy of the answer to go on with the next clue and provides aids/suggestions if required. The game can be played in offline mode since all the information related to a hunt is loaded into the device's memory.

The proposed application is based on content-based image similarity and uses computer vision algorithms executed onboard with the aim of quantifying image similarity between the picture captured by the phone camera and the correct image, without requiring a remote server and an internet connection. Please note that this is a main requirement for tourists who often pay a lot of money for international data roaming.

Computer vision is the science that aims at understanding the content of images or other complex data acquired by means of different kinds of devices and sensors. Computer vision is a research area that has been widely studied the past several decades, but now the growing diffusion of mobile devices with their high processing capabilities and powerful sensors like cameras, GPS, and compasses has renewed interest in this field; because of their growing ubiquity and capabilities, smartphones and tablets have turned out to be an ideal platform for computer vision applications [5]. In this work, a new project that combines mobile devices and computer vision algorithms is proposed that exploits the high computational power of modern devices to perform image processing onboard: the image recognition module on the mobile device manages the image acquisition, the feature extraction process, and the calculation of similarity measures to the stored objects (which are saved in the form of a set of descriptors).

In this paper we also report several experiments carried out in order to select the best method for evaluating image similarity. In the literature, there are several descriptors used to represent images and several similarity measures used to evaluate their similarity. Our experiments are aimed at comparing descriptors and measures for the instance recognition task, i.e. the task of assessing whether two images depict the same subject. This question must be answered without a training phase because the target images are not known a priori. To accomplish this goal, we perform experiments according to an ad hoc testing protocol that is quite different from those used in landmark recognition or object recognition. Our experiments, performed on seven datasets, prove that the image recognition system implemented in our application is efficient and robust in comparing images with different characteristics.

The arrangement of this paper is as follows. In section 2 related work is provided. In section 3 the architecture of the proposed system is described. In section 4 the implementation of the computer vision component is detailed, and in section 5 experiments on several benchmark datasets are discussed. In section 6 conclusions are drawn and future research directions are suggested.

2. Related Works

In the literature there are two classes of works related to

1 http://www.geocaching.com/
2 http://www.opencaching.com/

this application: *(i)* studies aimed at planning complex routes that exploit the potential of mobile devices to increase the quality of the end-user expertise in the context of educational or recreational activities and *(ii)* studies aimed at implementing some computer vision activities directly on mobile devices.

2.1. Location-aware Applications

Due to their small size and high portability, mobile devices are well suited to be used to explore museums, zoos, or historic heritage places and to guide visitors in their tours [6]. The continuously evolving technology brings new generations of mobile devices year by year, making them more powerful and more feature-rich: GPS sensors and high-resolution cameras increase the potential to utilize modern smartphones in outdoor activities, like location based games [7] or treasure hunt games [8].

Mobilogue ("MOBIle LOcation GUidancE") [7] is a framework for authoring and running mobile scenario-guided trips based on locations that contain information about the location, optionally a quiz and multimedia data. CityTreasure [8] is a client-server game designed to help players to visit a city focusing their attention on a specific touristic issue (e.g., the cultural heritage field). It is based on textual clues and supported by SMS mobile technologies.

Treasure-HIT [9] is a mobile application that implements a treasure hunt game. The creation tool allows users to prepare the hunts, which are then shared through a repository and can be run on mobile phones by means of GPS based activities and tasks and feedbacks from the players. Although this project is much related to GPH, it does not use photo clues and does not provide a content-based image similarity module to perform image recognition.

Another application for game-based city tours is presented in [10] where a framework for authoring and sharing complex tours is proposed. These tours allow tourists to explore new interesting sites in an exciting way: moving from station to station, users are requested to answer short questions or learn interesting details.

Snap2Play [11] is a mobile game inspired by the popular card game "Memory" where players are asked to match a pair of identical cards: a given image of a scene in the real world proposed by the system to a query image proposed by the user. Cards are matched with a scene identification engine which is located on an external server.

As tourism applications are concerned, the following works are somewhat related to GPH:

a) VISITO Tuscany [4] is an application which offers an interactive and customized advanced tour guide service to visit the cities of art in Tuscany. Specifically, the application focuses on offering personalized tours based on user interests, detailed information describing monuments related to user profiles, and information sharing in social networks. Unfortunately, the application requires an internet connection to work; therefore, it is useless for foreign tourists who usually have no 3G connection.
b) WhaiWhai[3] is an online game that allows the player to visit a city around the world in an unconventional way: by solving puzzles and discovering original stories. It is a collection of interactive guidebooks written to move the tourist away from the beaten tourist track through a collection of short stories about fascinating places. A mobile application is available which allows users to play in loco as well as in the comfort of their own homes.
c) Picture Geo Hunt[4] is a mobile GPS treasure hunt game, whose goal is to find the location of a picture using hints such as the direction or the distance to the picture. The pictures themselves are taken from the Panoramio database.

A crucial issue in most existing systems is the awareness of the user's location. While outdoor systems like Treasure-HIT can rely on GPS to detect the exact location, many projects aimed at including indoor activities uses alternative techniques such as WiFi [12], object recognition [13] [6] [14], RFID tags [15], and QR codes [7]. The advantage of GPH is the possibility of working in absence of GPS signals and an internet connection, thanks to the image recognition technology.

2.2. Computer Vision Applications

As the computational power of the smartphone and tablet has been rapidly increasing the last few years, several researchers have focused their attention on the design of computer vision systems for mobile devices, where the presence of built-in cameras and network connectivity make it increasingly attractive for users to take pictures of objects and then obtain relevant information about the object photographed. The efficiency, robustness and distinctiveness of image descriptors are critical to the user experience and to achieve real-time performance on a mobile device; however, existing descriptors and classification approaches are often too computationally expensive to perform on smartphones and tablets and are not sufficiently scalable and distinctive to gain high recognition accuracy in a wide range of applications.

For these reasons, most proposed applications for mobile devices focus on a single problem (i.e. landmark classification, object recognition [16], or face recognition) and use the device only for image acquisition, utilizing existing server-side engines to perform the search. A system based on images and text is proposed in [17], where images are first searched on the web, then text that is extracted from webpages is used to query a text search engine. In [18] a detection system for images of buildings acquired by a phone camera is proposed: the search engine is server-side and is based on the i-SIFT descriptor [18]. The first applications performing image recognition onboard are related to face recognition [19] and landmark recognition [20] [21]. In [22] an efficient coding of SURF descriptors suitable for mobile devices is proposed, and a comparative study of lossy coding schemes operating at low bitrate is carried out. In [23] a landmark recognition approach is proposed which performs

3 http://www.whaiwhai.com/

4 https://play.google.com/store/apps/details?id=pk.games.PictoGeoHunt

comparisons on the client, but requires a connection to the server in order to download appropriate models on the basis of GPS coordinates. An outdoors augmented reality system for mobile phones is proposed in [24] where images acquired by the camera on a mobile phone are matched on board against a database of location-tagged images using a robust image retrieval algorithm based on SURF descriptors [25]. In this application the connection to the server is required in order to continuously update the database of surrounding objects to reflect changes in the environment (according to proximity to the user). Recently a mobile client for the Lucene CBIR has been proposed [26] that uses GPS coordinates and a server-side image recognition module to recognize images acquired by the users. The trend in this field is to use new or existing descriptors capable to ensuring high efficiency, robustness, and distinctive power. For example, in [27] a binary descriptor called Learning-based Local Difference Binary (LLDB) is proposed that directly computes a binary string for an image patch using simple intensity and gradient difference tests.

The image recognition module in GPH does not require a search on a large database since it is designed to perform a simple instance recognition, i.e. to evaluate whether the acquired image and the reference image depict the same subject.

3. The Proposed System

3.1. Game Rules

The goal of the hunt game is to complete all the clues of a photo treasure hunt in the shortest time and with the highest score. The hunt is defined by its author as a list of clues that should be solved sequentially or in a random order: this is an author's choice and depends on the type of game. In the first case the author decides the path and in the second case the user can decide the shortest path. A clue is characterized by a question and is identified by a location, a photo, and a short description. A clue is considered solved if the user is in the correct location (evaluated by computing the distance of his/her GPS coordinates and the clue's location) and whether he/she has taken a photo of the searched object (evaluated by computing the similarity between the input photo and the stored image). In order to take into consideration GPS inaccuracy and variations in point of view, illumination and position in taking the photo, the two conditions above are evaluated using a low threshold. A score is given to each clue in consideration of the time taken to its solution, of the number of tips required, and of the quality of the answer (position and photo).

A game is created by inserting a question, a location, a photo, and possibly some tips for each clues. A game starts when the player opens the first clue and ends when the player has solved all the clues. After concluding a game, the player can upload his score on the server where a leaderboard is maintained. Differently from geocaching games, which also require identifying and visiting a number of targets, GPH does not provide GPS coordinates of the target position, which is one of the thing to be discovered to complete the clues. Rather the client system only provides the direction and the distance to help find the unknown location.

3.2. System Architecture

The system is designed according to a client-server architecture, where the client is a mobile device and the server is a physical machine that stores data about existing games and highest scores. Almost all the work is done on the client, both for the creation phase and the playing phase. A graphical schema of the software architecture is reported in Figure 1. Both the game authoring and playing are performed on the client: a local database is used to store the data of each game and the local CBIR module performs feature extraction and similarity calculation for the photo images. The server stores downloadable games and high scores in its database. The registration and connection to the server is needed only to interact with the community of players (upload/download new games, insert/view scores).

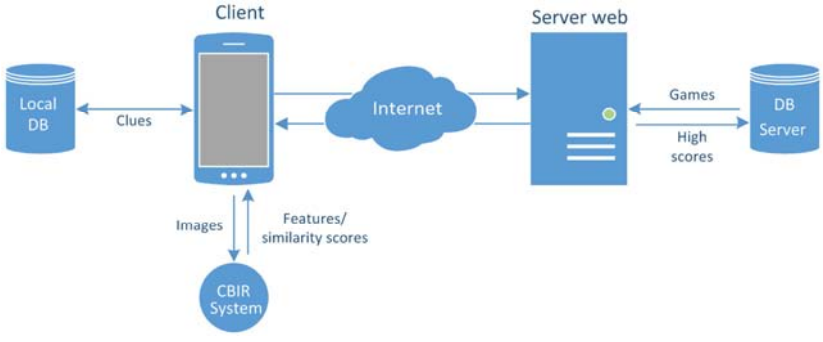


Figure 1. *Software architecture.*

3.3. Client Application

The client application, designed for Android platforms, is the core of the system and performs the following tasks:

a) Game authoring: it allows the user to create a photo treasure hunt by inserting a list of clues. Each clue is defined by a position (acquired by GPS or selected on a map), a question, a photo and, optionally, some tips. The CBIR module inside the client performs feature extraction from the photo and the visual descriptors are stored with other data in the local DB (Figure 1). An

UML class diagram of the hunt is illustrated in Figure 2.

b) Game play: it allows the user to open a clue, to read the question, to use the compass to drive the search, to ask for tips and to give an answer in the form of a photo. The playing phase is designed to work without requiring internet connection. The system evaluates the answer according to: *(i)* distance between the clue and user position, *(ii)* orientation of the user to the target, *(iii)* similarity between the captured image and the stored image. The image similarity is evaluated as a 1:1 matching by the internal CBIR system.

The class diagram in Figure 2 shows the structure of a treasure hunt: the main information such as length, difficulty, and name resides in the main class (class *PhotoTresureHunt*) while the clues (class *Clue*) are connected to one treasure hunt. Each clue is provided with one to three tips (class *Tip*). The features (class *Features*) are part of the class Clue and contain the descriptors extracted from the image provided as a solution (the image is not stored unless it is given as a tip).

A local database (SQLLite) is used to save the main information of each treasure hunt on the device. The client application has the following hardware requirements: GPS, gyroscope, magnetometer (to detect user position) and camera (to take a photo); the internet connection is required to view maps in the authoring phase and for the connection to the server.

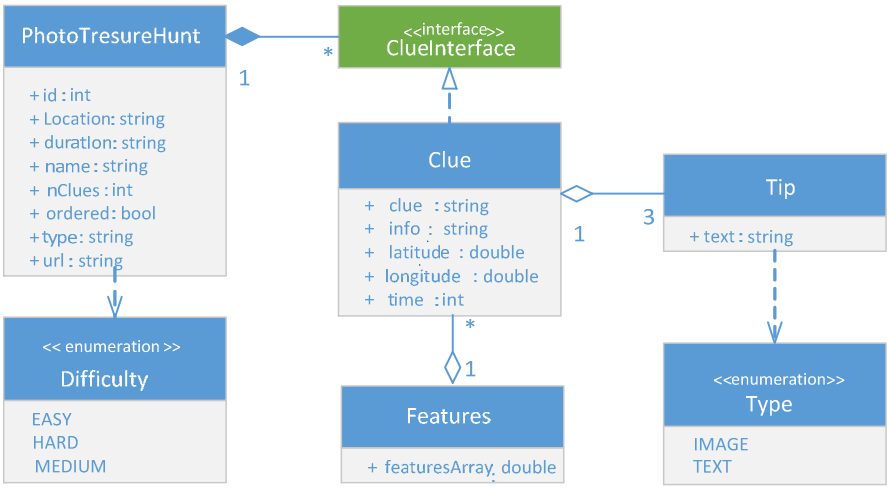


Figure 2. *UML class diagram of the treasure hunt client application.*

In Figure 3 and 4 some screenshots of the client application are reported.

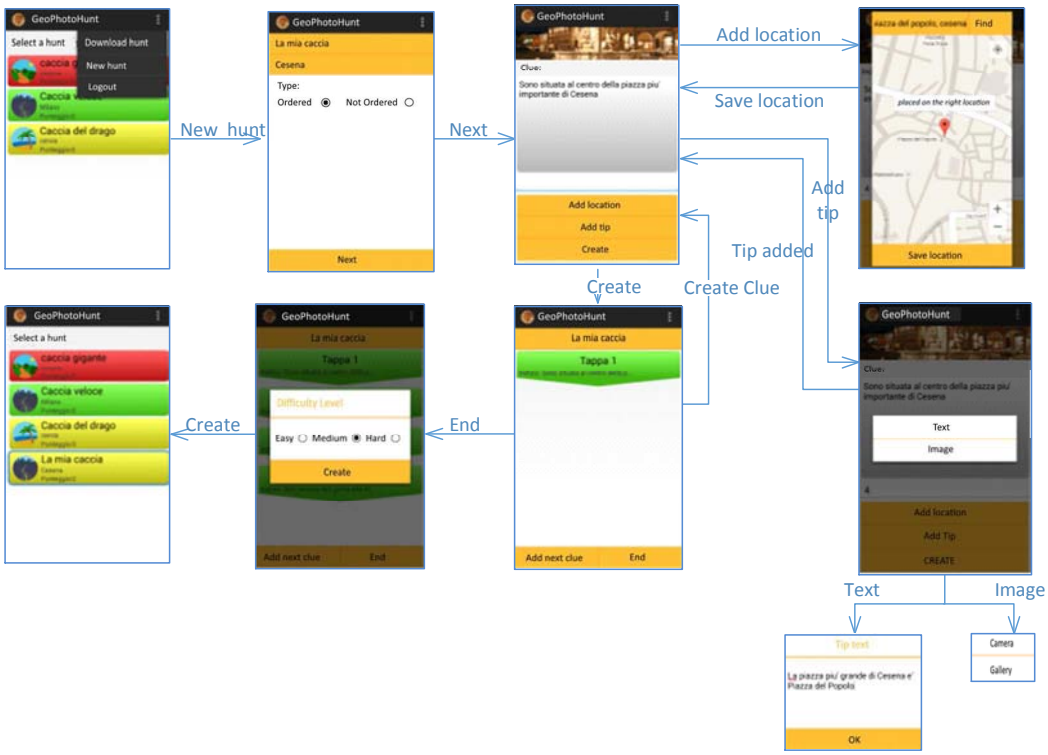


Figure 3. *Screenshots of the client application for the game authoring phase.*

Figure 4. *Screenshots of the client application for the game playing phase.*

3.4. Server Application

The server application supports the management of an online community of players in order to share hunts and scores. The user registration on the server is not mandatory to use the client and authoring/playing games. In particular the server offers the services of:

a) Registration and login
b) Cloud storage: a user can share a hunt by uploading the hunt to the server. Each hunt is assigned a unique "id", stored in its own directory and its main data are added to the server DB. Other registered users are allowed to search and download stored hunts.
c) Scoreboard: when a player completes a treasure hunt he\she can choose to share his\her score with other players.

The server is composed by a DBMS MySQL and a Web Server; the communication between client and server is performed according to the HTTP protocol to send requests about existing hunts and scoreboards. The upload/download functionality is implemented using PHP scripts and the JSON protocol for the exchange of data (Figure 5). Each treasure hunt is composed of a file with the extension.gph containing a serialization of the data detailed in Figure 2 and a folder where the image tips (at low resolution) are stored: all these files are compressed in a single zip file.

```
{
    "id":"15",
    "name":"Caccia gigante",
    "nClues":"5",
    "ordered":"1",
    "type":"outdoor",
    "location":"Cesena",
    "duration":"5",
    "url":"uploads\08152014125251ncsc017kqk.zip",
    "difficulty":"HARD",
    "username":"Alessandra Lumini",
    "longitude":"12.3121584579349",
    "latitude":"44.1939817211012"
}
```

Figure 5. *JSON format for data exchange between client and server.*

The PHP script that provides the service of uploading treasure hunts has the following functions: when a new hunt is required to be uploaded, the script first checks the user authentication and ensures the hunt is not already in the database. Then the JSON data received via POST is saved in the DB and the ".zip" file is loaded and saved in a new directory. Finally the server informs the client about the outcome of the operation.

4. Computer Vision Component

The computer vision module consists of:

a) Data acquisition: performed by the device camera;
b) Feature extraction: characterization of an image by descriptors, possibly with high discriminating capability and robustness to object variations.
c) Matching: evaluation of the similarity among descriptors of the acquired image and the reference one in order to recognize that two images, as are represented, are the same.

Typical descriptors used for assessing image similarity are based on color, shape, texture, and spatial layout to represent the image by multi-dimensional feature vectors. The matching step involves the definition of similarity/distance functions between the feature vectors of two images, which can be used to assess similarity, to rank a database of images according to their similarities to the query image or to perform a classification task.

Early descriptors, used in the first computer vision systems as QBIC [28] and Nextra [29] were related to global features based on image color, texture, and shape.

Color is one of the most important properties identified by human vision; therefore, color descriptors have a great importance in the literature. A taxonomy [30] of color based approaches divides the proposed methods into: *(i)* global approaches, which consider the image color information globally, without encoding information about the spatial distribution of colors (e.g. the global color histogram [31] and the cumulative global color histogram [32], *(ii)* fixed-size region approaches, which extract color information locally from regular cells of fixed size, at the cost of generating larger feature vectors (e.g. local color histogram [31]); *(iii)* segmentation-based approaches, which are similar to the fixed-size region approaches except for the fact that they divide an image into regions that may differ in size and quantity via a clustering algorithm (e.g. color-based clustering [33]), thereby introducing extra complexity to the extraction process.

Texture is an important property that characterizes the spatial arrangement of colors or intensities in an image. Texture can be analyzed considering the value of attributes as roughness, contrast, directionality, regularity, and coarseness, and can be evaluated in a neighborhood of pixels. A taxonomy of texture-based approaches divides the proposed methods into: *(i)* statistical analysis, based on the extraction of statistical measures from co-occurrence matrices (e.g. Haralick [34]); *(ii)* geometrical methods which analyze textures by "texture elements" or primitives (e.g. [35]); *(iii)* model-based methods which rely on the construction of image models (i.e. a dark spot, an horizontal transition, corners, and lines) used to describe a texture (e.g. local binary patterns [36]); and *(iv)* signal processing methods, which use filter responses to characterize textures (e.g. Gabor Filters [37]).

The shape of objects is also often used for image comparison. Most methods for representing shape can be divided into two groups: *(i)* contour-based descriptors, which only employ shape boundary information obtained by various signatures, Fourier descriptors or wavelet descriptors (e.g. histogram of oriented gradients [38]); *(ii)* region based descriptors, which make use of all the pixel information across the shape region, for example, in terms of simple geometrical parameters, such as area or compactness (e.g. Zernike moments descriptors [39]).

Around 2000 newer descriptors were proposed based on feature extraction at keypoints. The idea of this approach is to detect salient image regions (keypoints) such that they are detectable despite changes of viewpoint, scale, illumination and to use a compact descriptor to capture the most important and distinctive information around the keypoints (e.g SIFT [40], ORB [41]).

More recently a large consensus has also emerged for Bag of Features (BoF) [42], a method derived from information retrieval and based on powerful scale-invariant descriptors that are used as a vocabulary to match identical regions between images. The descriptor is a vector that counts the occurrence of a vocabulary of local image features. The disadvantage of this approach is that it requires a training to determine the vocabulary.

It is known that image descriptors and related similarity functions are application dependent; therefore, conducting comparative evaluation of image descriptors considering a typical environment of use is very important in designing a computer vision application. In this work we tested several descriptors to find the most suitable for this problem, which is an instance recognition problem, i.e. we want to determine whether the query image represents the same subject of a given stored image.

4.1. Descriptors Tested in This Work

The state-of-art descriptors/methods evaluated in this work are the following:

1) Histograms of color in RGB and HSV color spaces [31]: Global and Local (from 3×3 grid). Color histograms are extracted from RGB or HSV color spaces in order to describe the number of pixels in each range of colors (or bin) independently. Different matching functions are used (see experimental section).
2) Color Coherence Vector (CCV) descriptor [43]: CCV is a color-based method similar to the color histogram. CCV also considers some spatial features of the image by separating coherent pixels from incoherent pixels in the histogram.
3) Histogram of oriented gradients (HoG) [38]: HoG is a shape descriptor which characterizes images by the distribution of local intensity gradients or edge directions. In this work an 18 bin histogram is extracted from local regions (5×5 grid). The cosine similarity is usually adopted as a matching function.
4) Perceptual hashing [44]: perceptual image hashing maps an image to a fixed length binary string based on the image's appearance to the human eye; in a-hash, the hash function is obtained from the average of the

colors, while p-hash uses a discrete cosine transform (DCT) to reduce the frequencies. The Hamming distance is used for the matching.

5) Local Binary Patterns (LBP) [45]: the LBP operator is an efficient method for describing texture in 2D. It is computed at each pixel of an image by considering the differences between grey-level values of a small circular neighborhood (with radius R pixels); the final descriptor is a histogram of binary codes. In this work the standard parameters (R=1; P=8) are used to extract the code.
6) Scale Invariant Feature Transform (SIFT) [40]: SIFT is a method for extracting distinctive features from images that can be invariant to image scale and rotation. The SIFT approach consists of a feature detector and a feature descriptor: the detector extracts a number of keypoints invariant to (some) variations of the illumination, viewpoint, and other transformations, and the descriptor extracts a feature vector from the region around a keypoint. Thus, differently from the methods above, the SIFT descriptor is not a vector of fixed size but a variable-length list of vectors. The matching procedure is a 1:1 matching of such vectors to obtain a global similarity score.
7) Speeded Up Robust Features (SURF) [25]: SURF algorithms is a keypoint based descriptor inspired by SIFT but several times faster and more robust than SIFT.

4.2. Similarity Measures Tested in This Work

After approximating images via feature representations, the main task is to find an appropriate function to determine similarities among data objects. In the literature [46] several similarity measures applicable to different classes of descriptors have been proposed, resulting in different effectiveness and efficiency. In this work we tested the following measures:

Two of the most used distance function for histograms are

a) Chi-Square Distance (χ^2):

$$D(h_1, h_2) = \frac{1}{2}\sum_k \frac{h_1(k)-h_2(k)}{h_1(k)+h_2(k)} \quad (1)$$

b) Bhattacharyya Distance (Bha):

$$D(h_1, h_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{h}_1 \cdot \bar{h}_2 \cdot n^2}} \sum_k \sqrt{h_1(k) \cdot h_2(k)}} \quad (2)$$

where h_1 and h_2 are the two input histograms of n buckets, h(k) is the value of the k^{th} bucket and $\bar{h}$ is the mean value of the histogram h. Both distances are normalized in the range [0..1] and transformed to a similarity measure by S=1-D.

For HoG descriptors the cosine similarity is used, which is a measure of the cosine of the angle between two vectors. Please note that this is already a similarity function; therefore, it does not need normalization.

c) Cosine Similarity (Cos):

$$S(h_1, h_2) = \frac{\sum_k h_1(k) \cdot h_2(k)}{\|h_1\| \|h_2\|} \quad (3)$$

Where $\|h\|$ is the norm of the histogram h. The cosine similarity is already defined in the range [0..1].

The perceptual hashing approaches use the Hamming distance (Ham). The Hamming distance between two strings of bits (binary integers) is the number of corresponding bit positions that differ. This can be found by using the XOR operator. Hamming distance is normalized by the length of the string and transformed to a similarity measure by S=1-D.

Both SIFT and SURF features require a matching procedure in order to compute similarity between two images, since both descriptors are not a vector of fixed size but a variable-length list of vectors. In this work the original matching procedures proposed in [40] and [25], respectively, have been used to obtain a list of paired keypoints ($\text{match}(f_1, f_2)$). Then the similarity is evaluated by considering both the number of the matched keypoints ($\#\text{match}(f_1, f_2)$) and the distance between the matched keypoints ($\sum \text{match}(f_1, f_2)$) to calculate the similarity measure among two sets of paired feature vectors f_1 and f_2:

d) Percentage of matching features (PMF):

$$D(f_1, f_2) = \frac{\#\text{match}(f_1, f_2)}{\min(\#f_1, \#f_2)} \quad (4)$$

e) Weighted Average Matching (WAM):

$$D(f_1, f_2) = \frac{\sum \text{match}(f_1, f_2)}{\#\text{match}(f_1, f_2)^2} \quad (5)$$

5. Experimental Results

In order to select the best method to evaluate image similarity, several experiments were conducted on the descriptors and similarity measures described above (implemented in MATLAB) and tested on several datasets. The experimental evaluation of the proposed system has been conducted on seven datasets: five datasets of landmarks, a dataset of objects taken from different points of view and a dataset of images appositely collected for the photo treasure hunt task. For the selection of the method to be implemented in the computer vision component, the retrieval accuracy, the computational cost, and the descriptor length are considered as performance indicators due to the strict requirements in response time and memory capacity. Since our purpose is to select a general approach without a dataset-dependent normalization, we do not use any kind of optimized or trained descriptor/similarity measure. A final experiment has been conducted on a mobile device to evaluate the effectiveness of the selected method.

5.1. Datasets

In Table 1 a summary of each dataset is reported. Due to computational issues all images having a size larger than 640 pixels have been resized, maintaining the aspect ratio, to 640 pixels.

The following 7 datasets have been used:

a) Mobile Phone Imagery Graz[5] [18]: this is a dataset of buildings collected in the city of Graz, Austria. It contains 80 images (640x480 pixels) of 20 buildings, with four views for each building.
b) Caltech Building Dataset[6] [47]: this dataset contains 250 high resolution images (2048x1536 pixels) of 50 buildings around the Caltech campus. Each building was photographed from different angles and distances.
c) Zurich Building Dataset[7] [48]: this dataset contains 1005 images (480x640 pixels) from 201 Zurich city buildings, with five images for each building taken from different points of view. Differences among the views include the angle at which the picture is taken, with relatively small scaling effects and occlusions. The Zurich Building Dataset contains a standardized query set consisting of 115 images, which were not used in this work since we do not need training data.
d) Italian Landmarks Dataset[8] [5]: this dataset contains 1435 images (of different sizes) from 41 famous Italian monuments. For each monument there are 35 images with different points of view, scaling, or occlusions.
e) Sheffield Buildings[9] [49]: this dataset contains over 3000 low-resolution images of 40 different buildings in Sheffield. For each building there are typically between 70 and 120 photos taken from different points of view and under various lighting conditions. These images are intended to represent the quality and variety of images obtained by hand-held mobile devices for a range of weather and time-of-day conditions.
f) Zurich 53 Object Dataset[10]: this small dataset contains 265 images (320x240 pixels) representing 53 objects, with five images for each object taken from different points of view. Differences among the views include the angle at which the picture is taken and relatively small scaling effects.
g) Cesenatico Dataset[11] [18]: This is a self-collected dataset of images taken in the city of Cesenatico, Italy. It contains 300 images (4096x2304, 2592x1944, 3264x2448 pixels) of 20 landmarks, with a variable number of points of view for each landmark. This dataset has been collected ad-hoc to simulate the acquisition task performed by game players, therefore images have been acquired using three different mobile phones in unconstrained conditions and during two sessions (one sunny and one cloudy); it is characterized by very large intra-class variations. A few samples from this dataset are shown in Figure 6 (three images for each landmark).






5 Available at: http://dib.joanneum.at/cape/MPG-20/
6 Available at: http://vision.caltech.edu/malaa/datasets/caltech-buildings/
7 Available at: http://www.vision.ee.ethz.ch/showroom/zubud/
8 Available at: http://bias.csr.unibo.it/lumini/download/dataset/ItaLa.zip
9 Available at: http://eeepro.shef.ac.uk/building/dataset.rar
10 Available at: http://www.vision.ee.ethz.ch/showroom/zubud/Obj_DB.tar.gz
11 Available at: http://bias.csr.unibo.it/lumini/download/dataset/Cesenatico.zip

Figure 6. *Samples from Cesenatico Dataset.*

5.2. Testing Protocols and Performance Indicators

In order to replicate the instance recognition problem required by the GeoPhotoHunt project, where the internal CBIR system has the task of evaluating the similarity between the captured image and the stored one, the testing protocol used for experimental evaluation in the above labeled datasets is a non-trained 1:1 matching. Each CBIR method (intended as descriptor and similarity measure) is evaluated measuring its accuracy in a two-class classification problem which consists in assessing whether two images do or do not belong to the same class (i.e. represent the same subject). Each image is matched against all the other images of the dataset.

The two distribution of scores obtained from pairs of images from the same class (named "genuine") and from pairs of images from different classes (named "impostor") are good graphical indicators to show how the CBIR method "separates" the two classes. Another standard performance indicator is the area under the ROC curve (AUC) [50], which is used to compare different classification algorithms independently of operating points, priors, and costs since it does not require a fixed classification threshold. The ROC curve is a two-dimensional measure of classification performance that plots the probability of classifying correctly the genuine samples against the rate of incorrectly classifying impostor samples. The AUC is a scalar measure to evaluate performance which can be interpreted as the probability that the classifier will assign a higher score to a randomly picked genuine couple rather than to a randomly picked impostor couple. AUC is included in [0, 1] range and should be maximized: 1 is the correct prediction, 0.5 is a random prediction, and 0 is an inverse prediction.

Other important factors for an implementation on devices are the memory occupancy of the descriptor and the computation time for feature extraction and matching.

Finally in order to compare the CBIR methods across the seven dataset the average rank is reported, which measures the relative performance of each method in the different classification problems.

5.3. Method Selection Experimental Results

In Table 2 the experimental results obtained using the descriptors listed in section 4.1 and different distance measures (section 4.2) on the seven datasets are reported. The following indicators are considered: AUC (in each dataset), Rank averaged in the seven datasets, size of the descriptor (fixed size for almost all, except SIFT and SURF, where the average number of keypoints is reported), and execution time in seconds for a single feature extraction and matching (evaluated on the Graz dataset, i.e. for images at 640×480 resolution). The experiments have been carried out using non-optimized MATLAB code on a Window 7 Pro 64bit machine (PC with Intel Xeon CPU E5-1620 v2 @ 3.70GHz, 64GB RAM).

Table 1. *Summary of characteristics of the seven datasets.*

Dataset	Short name	N° images	N° classes	Images/class	Resolution
Mobile Phone Imagery Graz	GRA	80	20	4	640×480
Caltech Buildings	CAL	250	50	5	2048×1536
Zurich Building Dataset	ZUB	1005	201	5	640×480
Italian Landmarks Dataset	ITA	1400	40	35	Variable
Sheffield Buildings	SHB	4178	40	35-334	160×120
Zurich 53 Object Dataset	ZOD	265	53	5	320×240
Cesenatico Dataset	CES	300	20	9-24	Variable

Table 2. *Experimental results on the seven datasets.*

Method		**Dataset (AUC)**							**Rank**	**Size**	**Time**	
Descriptor	Distance	GRA	CAL	ZUB	ITA	SHB	ZOD	CES	avg	avg(*)	FE	M
Global	χ^2	0.851	0.877	0.942	0.598	0.758	0.835	0.686	13.0	24	0.009	0.000
RGB Histogram	Bha	0.850	0.879	0.942	0.594	0.748	0.838	0.684	13.1			
Global	χ^2	0.835	0.932	0.968	0.629	0.810	0.841	0.676	9.3	24	0.015	0.000
HSV Histogram	Bha	0.834	0.934	0.967	0.625	0.810	0.842	0.675	9.4			
Local	χ^2	0.886	0.913	0.943	0.632	0.777	0.803	0.732	10.1	216	0.048	0.000
RGB Histogram	Bha	0.880	0.915	0.942	0.624	0.767	0.803	0.724	11.4			
Local	χ^2	0.879	0.953	0.964	0.662	0.816	0.833	0.753	6.3	216	0.070	0.000
HSV Histogram	Bha	0.876	0.954	0.961	0.655	0.812	0.829	0.745	7.3			
CCV	χ^2	0.878	0.878	0.953	0.616	0.744	0.848	0.689	11.4	54	0.249	0.000
	Bha	0.893	0.871	0.950	0.616	0.746	0.853	0.680	11.3			
HOG Histogram	Cos	0.738	0.890	0.895	0.728	0.805	0.760	0.809	11.3	450	0.017	0.000
a-hash	Ham	0.781	0.829	0.863	0.603	0.637	0.713	0.660	17.6	1	0.005	0.000
p-hash	Ham	0.688	0.712	0.797	0.576	0.588	0.691	0.608	19.0	1	0.005	0.000
LBP	χ^2	0.794	0.889	0.890	0.655	0.798	0.769	0.726	12.2	256	0.032	0.000
	Bha	0.792	0.889	0.890	0.655	0.797	0.768	0.725	13.1			
SIFT	PMF	0.999	0.953	0.979	0.771	0.905	0.825	0.840	4.1	$1522^{(*)}\times128$	0.552	0.260
	WAM	1.000	0.959	0.981	0.786	0.920	0.838	0.869	2.1			
SURF	PMF	0.995	0.929	0.972	0.758	0.815	0.806	0.800	6.0	$899^{(*)}\times64$	0.115	0.015
	WAM	0.997	0.960	0.983	0.784	0.844	0.899	0.852	1.9			

From the experimental results reported in Table 2 the following conclusions can be drawn:

a) The descriptors based on keypoints (SIFT, SURF) provide the best performance among the tested features in all the classification problems, and the WAM similarity measure is the most suited for these descriptors. Unfortunately, they have the big drawback of requiring high storage space and high computational costs. For this reason they are usually combined with a bag-of-feature approach [42]; however, in this work we did not test bag-of-feature approach since it requires a training phase that is not available for this instance recognition problem.

b) Perceptual hash representations (a-hash, p-hash) have the great advantage of a compact representation and fast matching, but they are more suited to such tasks as "find duplicates in a collection of images" than to determining image similarity.

c) In our experiments both texture and shape descriptors (HOG Histogram and LBP) are outcome by color descriptors in almost all the datasets.

d) Color descriptors perform quite well in almost all dataset, with a low computational time and a compact representation. By comparing the results it is evident that Local HSV Histogram gives the best trade-off between performance and computational requirements; therefore, this is the approach selected to be implemented in the computer vision module of the GeoPhotoHunt project.

In Figure 7 the Genuine-Impostor distributions are reported for the Zurich Building Dataset (which is the dataset containing the larger number of classes). The curves related to other datasets are quite similar and have not been reported for the sake of space.

The ROC curve of the best performing approaches (at least one descriptor for each class is included, coupled to its best similarity measure) are compared in Figure 8: the orange and blue curves denoting global and local HSV histograms are overcome only by SIFT and SURF.

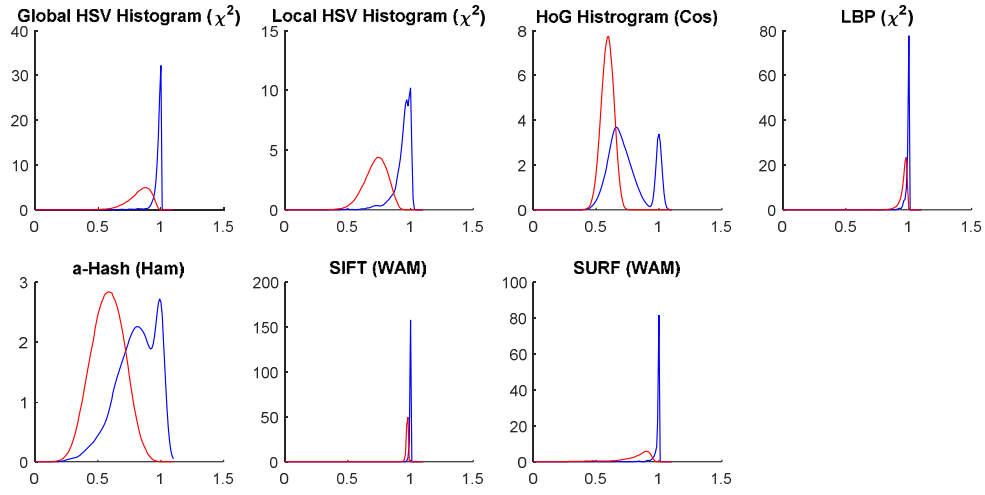


Figure 7. *Genuine (red)-Impostor (blue) curves in the Zurich Building Dataset.*

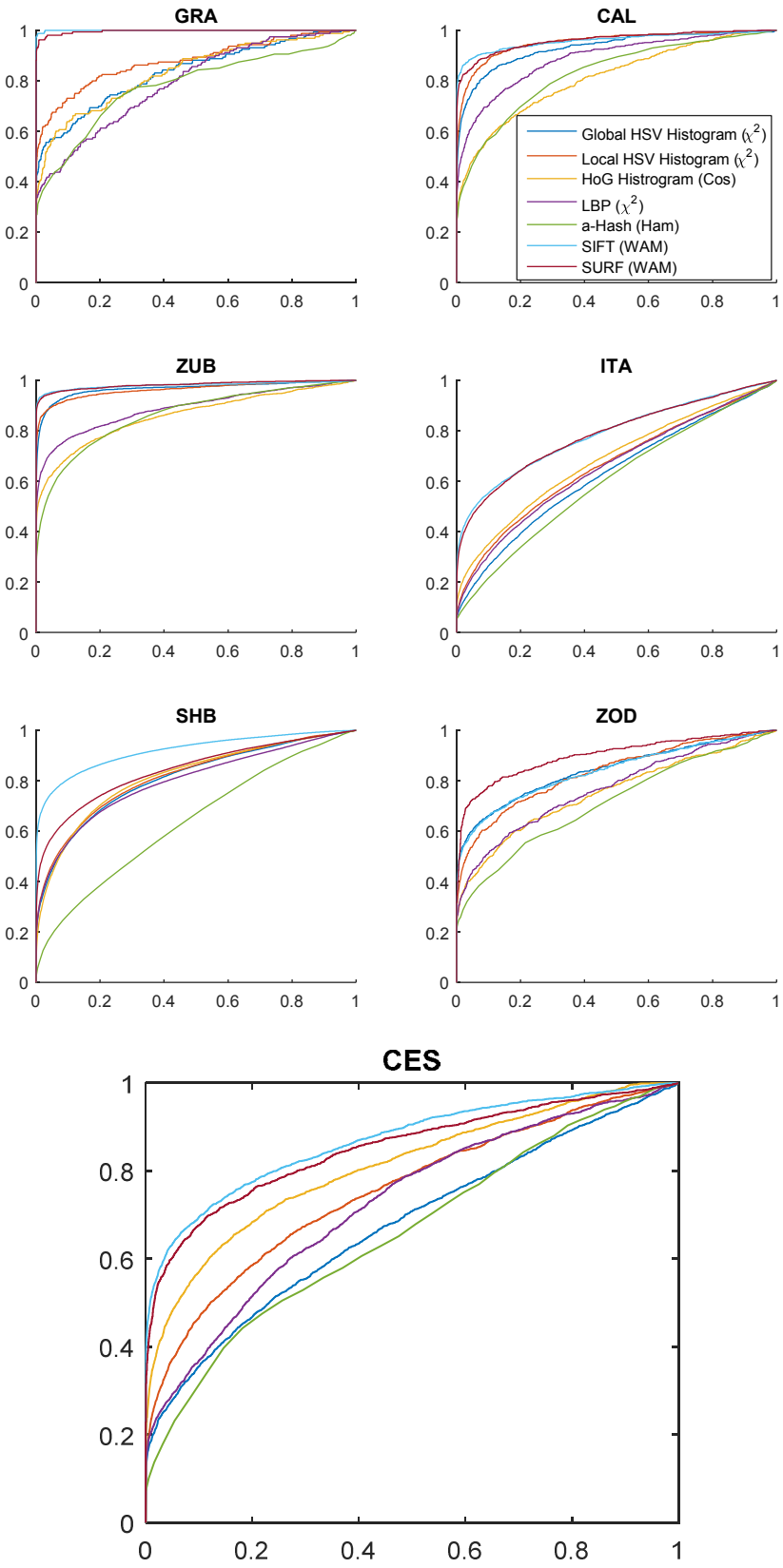


Figure 8. *ROC curve of best approaches in the seven datasets.*

5.4. Experiments on a Mobile Device

The previous experiments were aimed at selecting the most suited method for this image similarity problem, in this experiment we evaluate the effectiveness of the selected approach, i.e. local HSV histograms coupled to the Chi-

Square Distance. The experiment has been carried out using the Cesenatico Dataset which has been dedicatedly collected for this problem. One image per class (20 images in total) has been selected as stored (reference) images for a hunt composed of 20 clues, the remaining 280 images are used as testing images and compared to all the stored samples. This is an unbalanced classification problem, since for each reference image there are about 14 positive samples and 266 negative samples. In such a case accuracy is not a good performance indicator, since a high accuracy only reflects the underlying class distribution (i.e. accuracy paradox). A good performance indicator for this similarity problem is sensitivity, which measures the proportion of positives samples that are correctly identified. For this application we are more interested in that few correct samples are discarded (i.e. maximize true positives) than minimizing the number of false positives since it is not a real problem if a clue is solved by a wrong image. Since sensitivity depends on the classification threshold, both sensitivity and specificity (i.e. the proportion of negatives samples that are correctly identified as such) are reported as a function of the classification threshold in Table 3. According to the result in Table 3 we selected 0.65 as threshold for our application.

Table 3. *Experimental results on the Cesenatico dataset.*

Threshold	0.50	0.55	0.60	0.65	0.70	0.75
Sensitivity	1.00	1.00	0.95	0.85	0.68	0.47
Specificity	0.02	0.06	0.19	0.41	0.67	0.86

From the analysis of errors we observed that about the 80% of classification errors are related to only 5 classes (25% of classes). In Figure 9 some samples of correct and wrong classification are reported for two reference images (the first image in the row is the reference one, the second and third are true positive samples and the last two are false negative samples). It is clear that most error derives from changes in point of views rather the weather conditions.

Figure 9. *Error analysis on the Cesenatico Dataset: with blue border the reference images, with green border two true positive samples, with red border two false negative samples.*

The experiments have been carried out on a LG Nexus 5 (smartphone with CPU Quad-Core 2.26GHz, 2GB RAM): the feature extraction and similarity calculation process takes a computational time lower than 1 sec on the device.

6. Conclusions

In this paper a novel application for the development of a photo treasure hunt game in mobile devices is proposed. Thanks to quick development of mobile devices, a large number of people already own smartphones. The GeoPhotoHunt project provides an attractive way to visit cities by means of a photographic treasure hunt using images captured by the mobile camera of the smartphone. GPH makes available services for the creation and the implementation of a photographic treasure hunt and for playing the game using a mobile device possibly not connected to internet. The possibility of playing “offline” is very attractive for foreign visitors in order to avoid the payment of high roaming costs.

The application uses computer vision algorithms executed onboard (thus not requiring an external server to perform visual matching) with the aim of evaluating the degree of goodness of a photo given as a solution to a clue. In order to select the best method to evaluate image similarity, several experiments on seven different datasets have been performed which allowed a selection of a method with the best trade-off between computational cost and performance.

We performed experiments on 7 different datasets to compare descriptors and similarity measures for the instance recognition task, i.e. the task of assessing whether two images depict the same subject. This question must be answered without a training phase because the target images are not known a priori, therefore our testing protocol is based on 1:1 comparisons in a two-class classification problem which consists in assessing whether two images represent the same subject. Each image is matched against all the other images of the dataset. Our experiments prove that the Local HSV Histogram performs quite well in almost all datasets, with a low computational time and a compact representation, therefore it is well suited to be implemented in the computer vision module of the GeoPhotoHunt project. Another

experiment conducted on a mobile device confirms that the image recognition system implemented in GPH is efficient and robust in comparing images with different characteristics.

In conclusion, even if our results confirm that finding a single image similarity measure that performs well in any application/dataset is yet an open problem, the proposed solution prove to grant adequate performance for this application.

As a future work the effectiveness and efficiency of the system under more case studies will be tested and analyzed.

Acknowledgment

The author would like to thank Giammarco Tosi and Andrea Zagnoli for their contribution in developing the Android application and Prof. Sheryl Brahnam for proofreading. GPH has been developed at the Smart City Lab[12,] DISI, University of Bologna.

References

[1] S. Benford, C. Magerkurth, and P. Ljungstrand, "Bridging the physical and digital in pervasive gaming," *Communications of the ACM*, vol. 48, no. 3. p. 54, 2005.

[2] D. Nicklas, C. Pfisterer, and B. Mitschang, "Towards location-based games," in *Proceedings of the International Conference on Applications and Development of Computer Games in the 21st Century: ADCOG*, 2001, vol. 21, pp. 61–67.

[3] R. M. Webb, "Recreational geocaching: the Southeast Queensland experience," 2001.

[4] G. Amato, F. Falchi, and F. Rabitti, "Landmark recognition in VISITO Tuscany," in *Communications in Computer and Information Science*, 2012, vol. 247 CCIS, pp. 1–13.

[5] C. Casanova, A. Franco, A. Lumini, and D. Maio, "SmartVisionApp: A framework for computer vision applications on mobile devices," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5884–5894, Nov. 2013.

[6] G. Vavoula, M. Sharples, P. Rudman, J. Meek, and P. Lonsdale, "Myartspace: Design and evaluation of support for learning with multimedia phones between classrooms and museums," *Comput. Educ.*, vol. 53, no. 2, pp. 286–299, 2009.

[7] A. Giemza, N. Malzahn, and H. U. Hoppe, "Mobilogue: Creating and Conducting Mobile Learning Scenarios in Informal Settings," in *The 21st International Conference on Computers in Education (ICCE 2013)*, 2013.

[8] L. Botturi, A. Inversini, and A. Di Maria, "The City Treasure: Mobile Games for Learning Cultural Heritage," in *Museums and the Web 2010: Proceedings*, 2009.

[9] D. Kohen-Vacs, M. Ronen, and S. Cohen, "Mobile Treasure Hunt Games for Outdoor Learning," *Bull. IEEE Tech. Comm. Learn. Technol.*, vol. 14, no. 4, pp. 24–26, 2012.

[10] D. Grüntjens, S. Groß, D. Arndt, and S. Müller, "Fast authoring for mobile gamebased city tours," in *Procedia Computer Science*, 2013, vol. 25, pp. 41–51.

12 http://smartcity.csr.unibo.it/

[11] T. J. Chin, Y. You, C. Coutrix, J. H. Lim, J. P. Chevallet, and L. Nigay, "Mobile phone-based mixed reality: The Snap2Play game," *Vis. Comput.*, vol. 25, no. 1, pp. 25–37, 2009.

[12] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *Proc. IEEE INFOCOM 2000. Conf. Comput. Commun. Ninet. Annu. Jt. Conf. IEEE Comput. Commun. Soc. (Cat. No.00CH37064)*, vol. 2, 2000.

[13] P. Föckler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber, "PhoneGuide: museum guidance supported by on-device object recognition on mobile phones," *Proc. 4th Int. Conf. Mob. ubiquitous Multimed.*, vol. 6, pp. 3–10, 2005.

[14] B. Ruf, E. Kokiopoulou, and M. Detyniecki, "Mobile museum guide based on fast SIFT recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 5811 LNCS, pp. 170–183.

[15] A. Mody, M. Akram, K. Rony, S. A. Muhammad, and R. Kamoua, "Enhancing user experience at museums using smart phones with RFID," in *2009 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2009*, 2009.

[16] V. Tyagi, A. S. Pandya, A. Agarwal, and B. Alhalabi, "Validation of object recognition framework on Android mobile platform," in *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, 2011, pp. 313–316.

[17] T. Yeh, K. Tollmar, and T. Darrell, "Searching the Web with mobile images for location recognition," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004.

[18] G. Fritz, C. Seifert, M. Kumar, and L. Paletta, "Building Detection from Mobile Imagery Using Informative SIFT Descriptors," in *Scandinavian Conference on Image Analysis*, 2005, pp. 629–638.

[19] M. F. Arriaga-Gomez, I. de Mendizabal-Vazquez, R. Ros-Gomez, and C. Sanchez-Avila, "A comparative survey on supervised classifiers for face recognition," in *Security Technology (ICCST), 2014 International Carnahan Conference on*, 2014, pp. 1–6.

[20] K. H. Yap, T. Chen, Z. Li, and K. Wu, "A comparative study of mobile-based landmark recognition techniques," in *IEEE Intelligent Systems*, 2010, vol. 25, no. 1, pp. 48–57.

[21] P. Bhattacharya and M. Gavrilova, "A survey of landmark recognition using the bag-of-words framework," in *Intelligent computer graphics 2012*, Springer, 2013, pp. 243–263.

[22] A. Redondi, M. Cesana, and M. Tagliasacchi, "Low bitrate coding schemes for local image descriptors.," in *MMSP*, 2012, pp. 124–129.

[23] V. Hedau, S. N. Sinha, C. L. Zitnick, and R. Szeliski, "A Memory Efficient Discriminative Approach for Location Aided Recognition.," in *ECCV Workshops (1)*, 2012, vol. 7583, pp. 187–197.

[24] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*, 2008, pp. 427–434.

[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[26] K. Lin, J. Li, W. Huang, and C. Fang, "Smartphone landmark image retrieval based on Lucene and GPS," in *Computer Science Education (ICCSE), 2014 9th International Conference on*, 2014, pp. 368–371.

[27] X. Yang and K. T. Cheng, "Learning optimized local difference binaries for scalable augmented reality on mobile devices," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 6, pp. 852–865, 2014.

[28] C. W. Niblack, "QBIC project: querying images by content, using color, texture, and shape," *Proc. SPIE*, vol. 1908, no. 1, pp. 173–187, 1993.

[29] W. Y. Ma, Y. Deng, and B. S. Manjunath, "Tools for texture/color based search of images," *Hum. Vis. Electron. Imaging II*, vol. 3016, pp. 496–507, 1997.

[30] O. A. B. Penatti, E. Valle, and R. da S. Torres, "Comparative study of global color and texture descriptors for web image retrieval," *J. Vis. Commun. Image Represent.*, vol. 23, no. 2, pp. 359–380, 2012.

[31] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[32] M. Stricker and M. Orengo, "Similarity of color images," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1995, vol. 2420, pp. 381–392.

[33] R. O. Stehling, M. A. Nascimento, and A. X. Falcao, "An adaptive and efficient clustering-based approach forcontent-based image retrieval in image databases," *Proc. 2001 Int. Database Eng. Appl. Symp.*, 2001.

[34] R. M. Haralick, "STATISTICAL AND STRUCTURAL APPROACHES TO TEXTURE.," *Proc IEEE*, vol. 67, no. 5. pp. 786–804, 1979.

[35] R. Nevatia, *Machine Perception*. Prentice-Hall, 1982.

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[37] P. Wu, B. S. Manjunath, S. Newsam, and H. D. Shin, "Texture descriptor for browsing and similarity retrieval," *Signal Process. Image Commun.*, vol. 16, no. 1, pp. 33–43, 2000.

[38] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. CVPR05*, vol. 1, no. 3, pp. 886–893, 2004.

[39] a Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 489–497, 1990.

[40] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.

[41] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[42] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[43] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia (MULTIMEDIA '96)*, 1996, pp. 65–73.

[44] X. Niu and Y. Jiao, "An overview of perceptual hashing," *Acta Electron. Sin.*, vol. 36, no. 7, pp. 1405–1411, 2008.

[45] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *\TPAMI*, vol. 24, no. 7, pp. 971–987, 2002.

[46] C. Beecks, M. Seran Uysal, and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," in *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*, 2010, pp. 1552–1557.

[47] M. Aly, P. Welinder, M. Munich, and P. Perona, "Towards automated large scale discovery of image families," in *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, 2009, pp. 9–16.

[48] H. S. H. Shao, T. Svoboda, V. Ferrari, T. Tuytelaars, and L. Van Gool, "Fast indexing for image retrieval based on local appearance with re-ranking," *Proc. 2003 Int. Conf. Image Process. (Cat. No.03CH37429)*, vol. 3, 2003.

[49] J. Li and N. M. Allinson, "Subspace learning-based dimensionality reduction in building recognition," *Neurocomputing*, vol. 73, no. 1–3, pp. 324–330, Dec. 2009.

[50] C. X. Ling, J. Huang, and H. Zhang, "AUC: A better measure than accuracy in comparing learning algorithms," *Adv. Artif. Intell.*, pp. 329–341, 2003.