Mobile object retrieval in server-based image databases

D. Manger, F. Pagel and H. Widak
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation,
Fraunhoferstr. 1, 76131 Karlsruhe, Germany

ABSTRACT

The increasing number of mobile phones equipped with powerful cameras leads to huge collections of user-generated images. To utilize the information of the images on site, image retrieval systems are becoming more and more popular to search for similar objects in an own image database. As the computational performance and the memory capacity of mobile devices are constantly increasing, this search can often be performed on the device itself. This is feasible, for example, if the images are represented with global image features or if the search is done using EXIF or textual metadata. However, for larger image databases, if multiple users are meant to contribute to a growing image database or if powerful content-based image retrieval methods with local features are required, a server-based image retrieval back-end is needed. In this work, we present a content-based image retrieval system with a client server architecture working with local features. On the server side, the scalability to large image databases is addressed with the popular bag-of-word model with state-of-the-art extensions. The client end of the system focuses on a lightweight user interface presenting the most similar images of the database highlighting the visual information which is common with the query image. Additionally, new images can be added to the database making it a powerful and interactive tool for mobile content-based image retrieval.

Keywords: mobile object retrieval, image database

1. INTRODUCTION

In the last decade, the wide distribution of smart phones equipped with cameras has opened the doors to new ways interaction of users with various information systems. The interaction with mobile devices using buttons has already changed to interfaces based on touch screens and other sensors such as acceleration and position sensors. The availability of camera images of the user environment is further leading to even more complex ways of interaction. A vast number of apps are using the camera image for augmented reality (AR) techniques in order to offer up-to-date information to the user by integrating information into the current image. At the same time, today’s mobile devices provide virtually permanent access to the internet which enables the interaction with other users and their environments. One application which makes use of all these capabilities is mobile object retrieval where the target is to enable the user to determine information about the object in the camera image. Applying content-based image retrieval (CBIR) methods on a server, the task is to query a database of known images or objects with the user’s camera image. By using a common database on a server, the images of other users can contribute to the actuality of the database.

In this article, we present such a framework for mobile object detection with both the client side at the user and the server side for the object retrieval on a variable database. This paper is organized as follows: section 2 deals with related work. In section 3, we briefly outline the image retrieval methods used for the presented system, followed by the description of the client part in section 4 and the client-server interaction in section 5. One example of the usage is presented in section 5 followed by a brief conclusion.

2. OBJECT RETRIEVAL

The aim of content-based object retrieval systems is to compare images with respect to their content. To this end, local image regions are compared using local features. Local features like the popular Scale-Invariant Feature Transform (SIFT) [2] are used in many different topics of computer vision. They typically detect repeatable salient regions in an
image and subsequently encode their local image appearance in a descriptor. Given the two sets of descriptors of two images, similar regions in both images can be searched by determining descriptors which are similar in descriptor space, which is for SIFT usually 128 dimensional. Typically, distances are calculated by L2 norm and a threshold is applied on the distance or on the ratio of closest to second closest distance. The similarity of two images is then often calculated as the number of matching features.

For matching sets of descriptors, various heuristic algorithms have been proposed which can lead to an impressive speedup while sacrificing not too much of the descriptors discrimination [3],[4]. Nevertheless, in large-scale CBIR systems with thousands or millions of images, a pair-wise image comparison of the query image with every image of the database becomes infeasible. Besides, the memory consumption of the image features and their processing during one query prohibit a direct matching of descriptors sets. To solve this, the bag-of-words (BOW) representation has been proposed [9], which quantizes the features by assigning every feature to one element of a set of feature representatives called visual words. Thus, the image matching can be performed with text retrieval methods analyzing the common visual words of images. The set of visual words termed codebook or visual vocabulary is commonly obtained by clustering an independent set of features. Using large codebooks, the representation of an image becomes a very sparse vector indicating the occurring visual words. This sparsity can be exploited by inverted files which store for every visual word a list of references to the images containing at least one feature corresponding to that visual word. Figure 1 summarizes the basic components of an image retrieval system.

While enabling the construction of fast and efficient systems, the quantization of features also comes with the drawback of loss of information which leads to a reduced accuracy of the overall system. We use two popular extensions to circumvent the loss of accuracy namely Hamming Embedding (HE) [1] and Weak Geometry Consistency (WGC) [1]. Both techniques have shown a significant improvement of performance in large scale image retrieval. In previous experiments, we could confirm this for the performance in tattoo image retrieval [9] with a database of up to 330,000 images.

As rare visual words are assumed to be more discriminative, the similarity of two images given the two BOW vectors is commonly calculated using the tf-idf scheme [9]. It weights the BOW vectors according to both the local frequency (within the image) and the global frequency (within the entire database). In all experiments in this paper, we use the similarity function of [8] which is the cosine angle between the weighted BOW vectors which equals the L2 normalized dot product of the vectors. See [1] for details.

Figure 1. Basic setup of a content-based image retrieval system which quantizes features into visual words to create an inverted file.
To further increase the performance, we make use of a subsequent re-ranking step, which performs a matching based on the original features. The images are re-ranked according to the number of direct matching features with the query image.

3. MOBILE CLIENT APPLICATION

The mobile client of the system is realized through an Android App. The Android API functionality is used to access the camera for capturing a number of images. For every image, the resolution is set in order for the longest side to be 800 pixels at maximum and all available sensor data is attached as meta data (e.g. position, acceleration and compass sensor data). The user can select the images to be used as query images and start the search by transmitting the images to the server (see fig. 3). As soon as the server has processed the query image, it notifies the client of completion of the search which then in turn can download the result list of the most similar images.

4. CLIENT-SERVER ARCHITECTURE

The client server model is a standard-concept for sharing tasks in networks. Tasks are distributed via server on different computers and can be requested if required by multiple clients to solve their own tasks. The tasks provided can be standard tasks (e-mail, e-mail reception, web access, etc.) or specific tasks of a software or a program. A task is called service in a client server model.

A client server system is a software (application system) which uses the client server model for their tasks and functions or, in other words, the software has been developed to use the client server model. The system consists at least of two parts, a server and a client component, which usually run on different computers.

The smartphone, tablet PC or any other device running android establishes a socket connection to the server and sends a request containing either a query image or an image to be added to the database. After sending the request the socket connection is closed. Thus, the application is not blocked and the user can continue working with the mobile device as usual. After the server has provided the results of the request, a notification message is sent to the mobile device (client) using Google Cloud Messaging for Android. Google Cloud Messaging for Android is a free service which allows sending small data packets from a server to an Android powered device. This service is used by the mobile device to
indicate new results on the server. If the mobile device is receiving a message, the message is displayed to the user by a notification. The results can then in turn be received over a socket connection from the server (see fig. 2). The advantage of this is that the user can decide when to download the data from the server. In case of a bad connection or if the mobile device is turned off, the user receives the message when the device is ready to receive.

Figure 3: Client App for mobile object retrieval (from left to right): main menu; capturing images with a region of interest; selecting the images to transfer to the server; displaying the results – query images together with the nine most similar images of the database.

[Query image: yogilino / pixellio.de]

5. EXAMPLE

Mobile object retrieval has plenty of domains of application. Possible scenarios range from field workers dealing with a gradually growing database of spare parts which are supposed to be recognized by means of image retrieval all the way to investigative work where new insights lead to new images which in turn should be available to other investigators in a timely manner. Due to license issues of the respective images, we present in this word an example of querying an image database of some 5,000 private holiday images. Figure 3 shows the process and the results of a query image of an object which was taken at different conditions of time, illumination, weather and angle of view. Please note that the capability of the local feature-based image algorithms to provide certain invariances to these effects is vital for real-world object retrieval.

6. CONCLUSION

In this article, we presented a mobile object retrieval system looking both at the image processing algorithms, the mobile client, the server backend side and the communication. The proposed approach allows a flexible setup of the subsystems which result in a variety of applications where users on site can contribute to and consult a common image database. For the object retrieval part, we provide a free demonstration software which can be downloaded and installed on windows machines to create and query one’s own local image database. See fig. 4 for an example. Further work might deal with the integration of the user location into the database and the searching functionality which can lead to increased search speed as only nearby images have to be compared if immobile objects such as buildings are meant to be found.

1 http://s.fhg.de/objectretrieval
REFERENCES


