Virtual Telescope: A Mobile Augmented Reality System

EE368 Digital Image Processing Final Project Report 2012

Qiyuan Tian
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
qytian@stanford.edu

Qingyi Meng
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
qingyim@stanford.edu

Abstract—We present a mobile image matching application for Android mobile phone called Virtual Telescope to assist users in viewing distant buildings. The application automatically augments photo's field of view by retrieving replacement images and satellite images from the Internet. The application adopts content based image retrieval method and allow user two choices of feature descriptor, namely Scale-invariant feature transform (SIFT) [1] and Affine-SIFT (ASIFT) [2]. The proposed application can also work as a general image retrieval system.

Keywords—content based image retrieval; SIFT; ASIFT; Android application

I. INTRODUCTION

Many mobile devices today come equipped with high-quality displays, high-resolution cameras, and hardware-accelerated graphics. With access to broadband connection and location-aware GPS capability, these mobile devices enable a new class of augmented reality applications for people to more conveniently browse inside or around a point of interest.

Sometimes people wish to more closely and clearly view a specific structure, especially a building or landmark, when they have a panoramic view of a large area, but suffer from the problem of great distance. A telescope is helpful in this case, and many well known tour spots, like the Oriental Pearl Television Tower in Shanghai, provide high-performance telescopes for tourists. However, most telescopes are fixed, bulky, usage specific, and inconvenient to carry.

Mobile phones with digital cameras can serve as convenient tools in this scenario. Specifically, users can use their phone cameras to continuously zoom in to those distant objects. Because an average camera phone available today has very limited optical power, leading to a zooming effect that is largely dissatisfying for examining the details within an image, the concept of augmented reality needs to be introduced.

We propose a mobile-compatible augmented reality software application "Virtual Telescope" as shown in Figure 1 to provide a zoomed-in version of the given image based on the requests of the user. Best of all, the resulting image is no longer limited by the optics and hardware of the camera. The organization of the report is as follows. In the next section, we briefly review previous works on content based image retrieval and mobile augmented reality. In Section III, we describe the system in details. Section IV shows the experimental results and section V concludes the report and discusses the future work.

II. PREVIOUS WORKS

A. Content Based Image Retrieval

Our proposed system retrieves images based on robust local descriptors [1], a method often known as content based image retrieval (CBIR) [2]. Good candidate descriptors include SIFT by Lowe et al. [1], GLOH by Mikolajczyk and Schmid [3], and SURF by Bay et al. [4]. These descriptors have been shown to be robust against variations in illumination, perspective, and scale.

B. Mobile Augmented Reality

There are also similar works in the context of mobile augmented reality. Takacs et al. [5] match images against a large database of location-tagged images using modified SURF and return the name and location of the building. They have considered aspects like geographic information and feature compression to achieve high matching accuracy and low network latency. Greene [6], on the other hand, does not perform any image analysis. Instead it uses an external GPS for
localization and an inertial sensor to provide orientation. PhoneGuide [7] employs a neural network trained to recognize normalized color features and is used as a museum guide. Other works include [8], [9] and [10]. To the best of our knowledge, there is currently no other proposal for a system similar to our virtual telescope.

III. SYSTEM DESCRIPTION

A. System Overview

Figure 2 provides an overview of the proposed system, which is divided into two major components: a mobile device and a server, forming a Server-Client Model. The two components communicate over a wireless network.

On the client side, the application allows users to choose the features for different tasks, namely SIFT for normal image retrieval and ASIFT for viewing distant buildings, after which users can zoom in and zoom out the phone camera and capture an image of the viewfinder. In order to decrease the size of transmission, the application provides users an interactive way to manually crop out the region of interest, i.e. region containing the building they wish to view in details. It also down-samples the cropped query image if it is of large size. Due to the limited computational power of the experimental phone, the application simply sends the query image to the sever, which will accomplish rest work.

The server maintains a dataset of a small number of images with their satellite images, two for each of them. The server has already computed and stored the descriptors for all images in the database. The moment the query image arrives, the server starts extracting corresponding features according to users' choice, either SIFT or ASIFT and then matches the extracted features against those in the database. If an image is identified in the database, the server return the matched image along with the satellite images, such as indoor scene and night appearance, which will be received and displayed by the phone. In this way, instead of displaying the actual scenes captured by the phone camera, the application augments the viewfinder with replacement images retrieved from the Internet, taken at a corresponding distance from the object, to change the field of view accordingly.

B. Client

On the android phone, we develop a java program containing an activity class named VirtualTelescope. It calls the Android system camera to take photos. In this class we declare three buttons associating with different functions, namely SIFT choice, ASIFT choice and photo capture. There are several key functions implemented. onActivityResult decides what to do: if a photo is taken, it prepares the photo for the user to crop by calling doCropPhoto and getCropImageIntent; if a cropped region is returned, the function delivers it to processImage function to set up query connection. Once called, processImage converts the image from bitmap to jpg format, connects the server, calls corresponding PHP file and uploads the image. After obtaining the result image, processImage returns to onActivityResult to display the result image.

C. Sever

1) SIFT feature

Lowe (2004) presented SIFT for extracting distinctive invariant features from images that can be invariant to image scale and rotation. Then it was widely used in image mosaic, recognition, retrieval and etc. Despite of its good performance, the algorithm is computationally extensive. However, in our experiment, we find that the bottleneck for real time interaction is uploading and downloading process, which occupy a time period at least 20 seconds based on the Internet. However, the running time for extracting and matching SIFT features is about 1 second, which is almost negligible. Although there are faster methods like SURF, we choose SIFT as the target descriptor for the retrieval process.

There are four steps to extract SIFT features: scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor. On the first stage, a difference-of-Gaussian function is used to identify potential interest points, which are invariant to scale and orientation. DOG is used instead of Gaussian to improve the computation speed [1]. Keypoint localization step rejects the low contrast points and eliminates the edge response. Hessian matrix is used to compute the principal curvatures and eliminates the keypoints that have a ratio between the principal curvatures greater than the ratio. An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment. Figure 3 shows the image with a random selection of 50 features visualized. We have also computed the SURF features, which is shown by the right image of Figure 4 with 150 strongest points.

2) Feature Matching

We use the pipeline shown in Figure 5 to accomplish the matching process. A k-dimensional tree (k-d tree) is a space-partitioning data structure for organizing points in a k-dimensional space. k-d trees are a useful data structure for accelerating searches involving a multidimensional search key, like nearest neighbor searches for descriptors in our case. FLANN is a library for performing fast approximate nearest
neighbor searches in high dimensional spaces. It contains a collection of algorithms we found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset. This enables fast medium and large scale nearest neighbor queries among high dimensional data points.

The method to reject ambiguous match was proposed by Lowe. A descriptor D1 is matched to a descriptor D2 only if the distance d(D1,D2) multiplied by a threshold is not greater than the distance of D1 to all other descriptors. Figure 5 shows the matched results for the given image pair. There are 1124, 1072 SIFT features for the left and right image in Figure 6. Before RANSAC, 1124 points are matched while there are only 11 post-RANSAC matches, which are obviously wrong. The experiment has demonstrated that SIFT does not work in this senario. The reason is that SIFT is not fully affine and can not deal with the situation that the perspectives of the two images vary a lot. In order to solve this problem, we adopts another algorithm called Affine-SIFT.

3) Affine-SIFT

Affine invariant extension of SIFT (ASIFT) is fully affine invariant. It simulates three parameters and normalizes the rest. The scale and the changes of the camera axis orientation are the three simulated parameters. The other three, rotation and translation, are normalized. More specifically, ASIFT simulates the two camera axis parameters, and then applies SIFT which simulates the scale and normalizes the rotation and the translation as shown in Figure 7. The matching result from ASIFT algorithm is shown in Figure 8. 48 matches are found and more than 40 are right. However, the computational efficiency of ASIFT algorithm is very low. It takes about 30 seconds to compute the descriptors and match them for an image pair.

IV. EXPERIMENTAL RESULTS

A. System Setup

In our experiment, we use Motorola MOTA855 as the client, and a Lenovo Laptop Y470 2.3 GHz 4 GB RAM wish WAMP software as the server. For the normal image retrieval function using SIFT features, the database contains 8 images of Stanford buildings shown in Figure 1. For ASIFT features, the database only contains 2 images in order to decrease the running time for the demo.
B. Results and Discussion

A video is recorded to demonstrate the effectivity of the system. We have tried using in the real world, however there is no access to network at all Stanford landmarks. Instead, we take photos of printed images.

In the experiment, the round trip time is about 20 seconds for the SIFT feature retrieval and about 45 seconds for ASIFT features. Uploading and downloading process take most of the time.

In the future, this technology can be incorporated into products like Google Image Search, Google Goggles, and Project Glass. It allows both professionals and the general public to virtually explore places of interest both spatially and temporally without being physically present at the location. For students, the technology leads to a more meaningful educational experience. For journalists, the technology enables a more efficient information-gathering process. For the general public, it is simply an eye-opening exploration. This technology would widen our perspectives of the world and help us better understand different countries, people, and cultures.

V. Conclusion and Future Work

In this report, we have presented an augmented reality system Virtual Telescope and demonstrated its effectivity in very limited situation. There are two main problems to make it into practice. The first is that the query image is of very low resolution, which makes the retrieval impossible in some extreme case. Besides, the perspective of the query image and search images in the database could be very different and this makes rather difficult. Even thought some algorithm can solve this problem to some extent, they are not computationally optimal. The future work focus on solving the two problems and improving the user interface. Possible solution for low resolution problem may be using multiple zooming in and retrieval. This method can have some intermediate steps instead of translating from two extreme cases. Besides, geographic information can be added to assist retrieval. All the images should be tagged with their capture location.

ACKNOWLEDGMENT

The authors would like to thank Professor Bernd Girod, teaching assistants David Chen and mentor Derek Pang for all the advice, instruction, and inspiration they provided throughout the Digital Image Processing class.

Images and figures used in this report courtesy of corresponding authors.

All work of this project is equally shared by Qiyuan Tian and Qingyi Meng.

REFERENCES


Images and figures used in this report courtesy of corresponding authors.

All work of this project is equally shared by Qiyuan Tian and Qingyi Meng.

REFERENCES