1 Introduction

In this project, we are going to apply augmented reality (AR) and computer vision algorithms for a Building Design Exhibition.

Currently, when an architect designs a building and wants to show it to the client, what they usually do is just show them some 2D pictures rendered by graphic engine, and some may also apply 3D computer graphics techniques to get better demonstration effect.

However, these are all virtual scenes. It could be hard for client to imagine what it will really look like on the site and what factors will affect the building and the environment, for there is no real scene accompany with the virtual building. Some architects do realize these and they render the virtual building with a picture of the real site as the background, so as to give clients a sense of reality and immersion.

Inspired by this, our project introduces AR into the field. Using computer vision and augmented reality approaches, we should be able to show client the virtual building in proper position and orientation when they point the mobile phone (DROID) to the site.

2 Technical Details

Since our goal is to perform outdoor augmented reality, meaning we have no prior knowledge about the environment, the widely used method nowadays, which is based on some plane markers with predefined size and also patterns to determine camera’s orientation and position, is not applicable in our application. So it requires us to look into some state-of-the-art marker-less based approaches.

2.1 Parallel Tracking and Mapping (PTAM)

PTAM [1] will be our project’s basis, for its amazing performance on personal computer, its open-source code and a released but unopen-source iphone application. The main idea of PTAM is that it wants to take advantage of existing visual simultaneously location and mapping (SLAM) achievements. However, the key step in visual-SLAM, Bundle Adjustment, is so highly computational expensive that if it is applied frame by frame, the real-time performance of the system will be impossible temporarily even in personal computer, not to mention mobile phone. So the innovation of PTAM is that it split the SLAM into two different parts, the tracking part and the mapping part. The former is usually not that highly computational expensive. So one can do the tracking in each frame in one thread, while in another thread, the system will occasionally do the bundle adjustment to improve mapping quality. In this way, PTAM can achieve real-time performance without losing registration quality.

In our application, since the platform changed to mobile phone, we want to make some changes of the algorithm. First, taking a further step from PTAM’s split of tracking and mapping into different thread, we want to put them into two different platforms. That is to say, the tracker will be run on
the DROID, while the mapper will be run on the server side, i.e. the personal computer. Of course, lots of configurations need to be explored, especially for the data communication part. What should be sent to server, only features detected or along with image? What should be sent back to DROID, fully reconstructed point cloud of just a sparse part of it? We need to figure those out during the project.

Also, as the author of PTAM pointed out [2], the modern mobile phone usually comes with different sensors, providing rough but quick information of the phone’s pose, how can we utilize those information is another topic we want to explore.

![Figure 1: Proposed schema of our project

2.2 Phony-SIFT

In the PTAM, the most important component is the tracker, which will detect and then match feature points between two frames. Fast detector [3] is applied in PTAM. However, we want to try other different detector and descriptor for this task. Phony-SIFT [4] [5] is one of our target. It takes advantages of SIFT, which is too slow to achieve real-time, and FERNS, a fast but high memory consuming feature descriptor. After combining these two state-of-the-art techniques, the Phony-SIFT turns to fit mobile phone quite well. Is it applicable to replace FAST detector with Phony-SIFT so as to improve the tracker’s matching performance? In the last two stages of our project, we want to do some related experiments.

3 Milestones Achieved

Sep. 15 - Sep. 30 Literature Review.

Oct. 01 - Oct. 15 Understand the architecture of android system, and get familiar with it’s development. Understand PTAM’s algorithm logic.

Oct. 16 - Oct. 30 Implement basic applications of the system: sensor, android-opencv, android-opengl.

4 Milestones Remained

Nov. 01 - Nov. 15 Understand PTAM’s source code, split it into DROID part and server part, finish major functions of the system.

Nov. 16 - Nov. 30 Try to improve PTAM with sensor data and other tracking methods, like Phony-SIFT, line based methods [6].

Dec. 01 - Dec. 10 Experiments, summary of the project, presentation and report.
References


