16811 MATH FUNDAMENTALS FOR ROBOTICS

FINAL PROJECT REPORT

Real-time Virtual Television in Augmented Reality

Kai Yu (kaiy1)

Zhongxu Wang (zhongxuw)

Ruoyuan Zhao (ruoyuanz)

Qiqi Xiao (qiqix)

December 13, 2017

1 ABSTRACT

Augmented Reality(AR) is a group of applications that uses computer vision to analyze the real world and overlays virtual objects onto it. When wearing the AR device(i.e. an AR glass like Hololens), you don't need a real television or a screen connected to your PC. You can simply have a virtual television on any surface in the world, with its size adjustable, content specifiable, and able to be paused automatically when you want to stop for a while. Besides, you don't need to buy any super high-resolution television any more: just project a virtual television on a really large wall and watch it! In this project, our group implemented an augmented reality application from scratch. It projects a real-time virtual television with live video content onto a user-specified surface.

2 Introduction

In this project, we built the basic algorithm of this augmented reality application. And we also use some pre-recorded video for demonstration in stead of a real AR device, which can be harder to obtain and operate. A video shoots the scene that contain a surface, and our program can produce a new video stream that has a virtual television projected onto the surface, with video pre-stored video content playing in it. For a real AR device, this video can be replaced by the new video generated by the AR light engine.

Basically, there are two parts: perform 3D reconstruction inside the system based on 2D camera only, and then project a virtual television onto the scene.

The 3D reconstruction is based on simultaneous localization and mapping, which is as known as SLAM. In this project, we adopt the PTAM-ORB-SLAM framework. PTAM represents a multi-threaded key frame based on SLAM. And ORB features are used for key point detection and matching. Besides, simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it. In our project, we extract the key idea of SLAM to estimate the 3D key points.

The key step of 3D reconstruction is the triangulation. Generally, the configuration contains two sensors. The projection centers of the sensors and the point on the object's surface can define a triangle. Within this triangle, the distance between the sensors must be known. Determining the angles between the sensors, the intersection point and the 2D coordinate can be calculated.

Also, the 3D estimate process may not be accurate enough, thus we apply another key step: Bundle Adjustment. It can optimize the rough solution given by triangulation. However, bundle adjustment is solving a nonlinear least square problem, thus takes much longer time. The PTAM framework is designed to mitigate this problem. It creates a new thread for bundle adjustment, enabling the system to estimate extrinsic metrics in real-time.

In this project, we combine the above knowledge and build an AR system based on these knowledge. In section 3, we introduced some basic background knowledge, including ORB, SLAM, Bundle Adjustment and Canny algorithm. In section 4, some implementation details of this project is elaborated. And in section 5, we summarize our project and propose some further possible improvement.

3 Background Knowledge

3.1 ORB

find points

3.2 SLAM

estimate 3d points

3.3 Bundle Adjustment

Given a set of images depicting a number of 3D points from different viewpoints, bundle adjustment can be defined as the problem of simultaneously refining the 3D coordinates along with the parameters of the relative motion, and the optical characteristics of the cameras. It refines a visual reconstruction to produce jointly optimal 3D points and parameters. When we assume the intrinsic matrices of different frames stay still. Bundle Adjustment can be simplified to just optimize the 3D points and the the extrinsic matrix for each frame.

It means to find the set of parameters that most accurately predict the locations of the observed points in the available images. So bundle adjustment is usually achieved by minimizing the re-projection error between the observed 2D points and the predicted 2D points, using nonlinear least-squares algorithms.

Assume that n 3D points are seen in m views and let x_{ij} be the projection of the ith point on image j. Let v_{ij} denotes the binary variable that equals 1 if point i is visible in image j and 0 otherwise. The re-projection error can be specified as:

$$\min_{a_j,b_i} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} d(\mathbf{Q}(\mathbf{a_j}, \mathbf{b_i}), x_{ij})^2$$

where vector $\mathbf{a_j}$ expresses the extrinsic parameters and each 3D point is parameterized by a vector $\mathbf{b_i}$. And $\mathbf{Q}(\mathbf{a_j}, \mathbf{b_i})$ is the predicted projection of point i on image j, while the $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between x and y.

In order to minimize sum of the errors, non-linear least squares algorithm is applied. The fitting is a maximum likelihood estimation of the fitted parameters. There are four most popular techniques for non linear least square optimization: Gradient Descent Mehod, Newton-Rhapson Method, Gauss-Newton Method, Levenberg-Marquardt Method.

3.4 Canny Edge Detector

The canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect edges in images. Canny edge detection is a technique to extract useful structural information from different vision objects and also reduces useless information. Among the edge detection methods so far, Canny edge detection algorithm is one of the most reliable and popular detection algorithms.

The process of Canny edge detection algorithm is elaborated as follows.

1. Apply Gaussian filter to smooth the image so that remove the noise;
2. Find the intensity gradients of the image;
3. Apply non-maximum suppression to get rid of spurious response to edge detection;
4. Apply double threshold to determine potential edges;
5. Finalize the edges by suppressing some weak edges that are not connected to strong edges.

4 Implementation Details

4.1 Pipeline

Our pipeline contains several steps:

1. Record video and intrinsic matrix of camera;
2. Extract key points based on orb feature extractor;

3. Match key points using KNN method;
4. Pick up key frames according to distances;
5. Reconstruct 3D cloud points using triangulation;
6. Use bundle adjustment to optimize projection matrix and 3D points;
7. Project virtual television to the surface.

4.2 ORB

find points

4.3 SLAM

estimate 3d points

4.4 BA

4.5 CANNY

find edges, cross product

4.6 PROJECTION TO THE PLANE

5 DISCUSSION