CMSC 25050 Computer Vision Project Report

Liu Cao & Deqing Fu

Introduction

It's always amazing that living creatures have the ability to measure distance and reconstruct three-dimensional structures though the images they sense from their retina are in fact 2D. Then it brought up the question — is there a way to reconstruct back the 3D structure from a set of images? Then the topic of "Structure from Motion" (SfM) arises. It has its profound influence in nearly every field, such as geoscience where scientists can utilize SfM to reconstruct and analyze terrains, and such as archaeology where archaeologists can use SfM to reconstruct lost ancient objects with their remaining documents of images. The most interesting is a gigantic project that reconstruct Roma using hundreds of thousands of images. Attracted by these huge applications and research projects, we decided to learn and implement a simple *Structure from Motion* model where we can deal with descent datasets.

For this project, in terms of the camera, we use the pinhole camera model. There are two different scenarios about real life situations. One is that we only have a set of images and the intrinsic parameters of the camera and the other is that we have the set of images together with its extrinsic parameters. The latter would be easier for computation where the extrinsic parameter matrix can be written as the rotation matrix augmented with the translation matrix: $[R \mid t]$, where R is the rotation matrix while t is the translation matrix. Let c be the camera center, then we have $t = -Rc$. As rotation matrices are orthogonal matrices, then $R^{-1} = R^T$, thus we can compute the camera center from the extrinsic matrix as $c = -R^T \cdot t$. If it's the first scenario where we only have the intrinsic parameters and the images, then we can apply feature matching first and use RANSAC method to find the *fundamental matrix* from the matched points. Then we can derive *essential matrix* from the fundamental matrix and use the essential matrix and the intrinsic matrix to compute the extrinsic which would return the method of the latter scenario. Once we have the camera centers, we can use triangulation to find the intersection (or say the point whose distance is the minimum to two matched beams) of the beams from their camera locations of the two matched points. Thus, we'll have a set of points in the 3D space, which is the cloud of the structure we want to reconstruct.

In terms of the data, we uses two sets of data for two different scenarios discussed above. The first one is the dinosaur dataset from **[CITE HERE]** which provides us the projection matrix $P = K[R \mid t]$, where K is the intrinsic matrix, R the rotation matrix and t the translation matrix. The second data set is 9 headphone images from different directions using an iPhone 8, where we only have the intrinsic information of the camera.

The results of our project are the reconstructed 3D points cloud from the set of 2D images. **[Result section in Introduction. I think this is only brief results.]**

Method

We firstly consider the scenario where the projection matrix $P = K[R \mid t]$ is given (the dinosaur dataset). Notice that for this kind of problem, $P \in \mathbb{R}^{3\times 4}$, then we can write it as $P = [A \mid d]$. Then we use the following decomposition method: firstly we use the QR algorithm to decompose P to get $A = qr$ where q is orthogonal and r is upper triangular. Then we have

$$R = \begin{cases} q^{-1} & \text{if } \det(q^{-1}) > 0 \\ -q^{-1} & \text{if } \det(q^{-1}) < 0 \end{cases}$$

And we derive the intrinsic matrix as $K = r^{-1}$ and the translation matrix as $t = rb$.

If the extrinsic matrix is not given. Then we **[Find it via Fundamental, Essential and Intrinsic]. Please write this. I'm confused.**.

The methodology flow is

1. Find Extrinsic Matrix $[R \mid t]$ for each image.
2. For images i and $i+1$, find their matched points X_1 and X_2, and find their camera centers $\vec{c}_1$ and $\vec{c}_2$. Then for each matched point pair $(\vec{x}_1, \vec{x}_2)$, we compute the direction vectors $\hat{v}_1 = \dfrac{\vec{c}_1 - \vec{x}_1}{||\vec{c}_1 - \vec{x}_1||}$ and $\hat{v}_2 = \dfrac{\vec{c}_2 - \vec{x}_2}{||\vec{c}_2 - \vec{x}_2||}$. Then we have two beams, whose parametrization are $\vec{b_1}(t) = \vec{x}_1 + \hat{v}_1 \cdot t$ and $\vec{b_2}(t) = \vec{x}_2 + \hat{v}_2 \cdot t$. Then we want to find a 3D point $\vec{p}$ whose distance to $\vec{b_1}$ and $\vec{b_2}$ achieves minimum. Then we can use the formula from the book (suppose the vectors are represented as column vectors) :

$$\vec{p} = \left(\sum_{i=1}^{2} (I - \hat{v}_i \hat{v}_i^T) \right)^{-1} \cdot \left(\sum_{i=1}^{2} (I - \hat{v}_i \hat{v}_i^T) \vec{c}_i \right)$$

Thus we derive a set of $\vec{p}$'s, which is the cloud of points of our reconstruction.

Experimental Results

Discussion

Reference