Incremental Structure from Motion

Ali Jahani Amiri
Instructor: Prof. Ping Tan

Introduction

The purpose of this project is generating a 3D point cloud from multi views of a scene.

First we find the features and then matching them. After that, we find the essential matrix by using 5 point algorithm and RANSAC. Then we decompose epipolar constraint (essential matrix) into rotation and translation. Then we delete the outliers and by projection matrix triangulate the 2D coordinations and finding their 3D coordinates.

This is the initial 3D reconstruction from two images. For the third and next images(image number k) , we need to find their common 2D coordinates in the k-1 image. So we have their 3D and 2D coordinates so now we can find out the projection matrix and reconstruct other new points in image k.

Finding 2D matching points:

Features	Good matches	Sift detector on CPU	Sift descriptor on CPU	Brute force Match
3000	1343	0.798413 second	0.970301 second	0.0988223 second

Matching Filter:

There are two kind of filter that these two are serial.

First filter (descriptor distance): We determine the minimum and maximum distance in the matched descriptors and we will choose only ones who has distance less than 10 times minimum distance.

Second filter (using coordination of paired match): Since our baseline is not so much and also we assume we have no sudden rotation around Z axis of our image coordination (image plane) so we can use tan(θ) of the paired matches to only get the ones who are in similar direction.

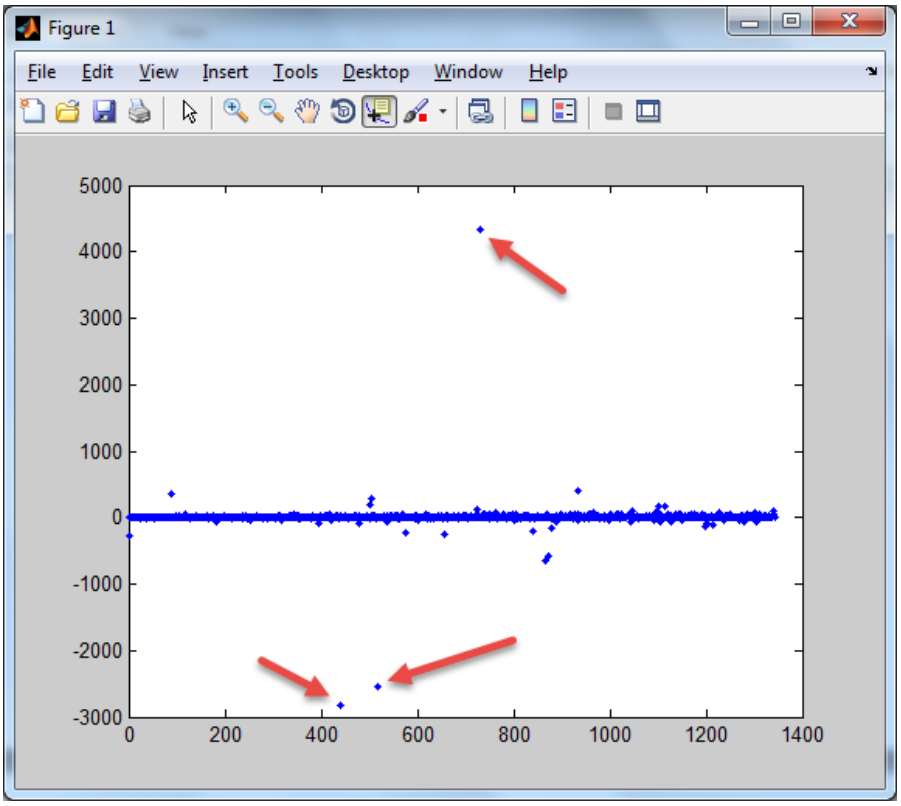
For this one for example we have $p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ and $p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ which are our matches. I added image width to the x2 and then calculate the tan(θ) :

$$\tan(\theta) = \frac{y_2 - y_1}{x_2 - x_1 + \mathrm{Im}\,ageWidth}$$

Adding image width to our x gives us better result.

In the diagram below, Y axis is calculating tan(θ) without adding image width for the data which were filtered with adding image width. As we can see, there are 4

points which seems good match when we added image width, on the other hand, if I don't add image width to x it seems to be very bad match.



The image below is these 3 matches. As we can see these are not so bad matches.

So adding image width to one x and then calculating the tan(θ) will lead us to better results.

Image below is the matched points without and with filter. As you can see there is a good matching between these 2d coordinates.

So now we have our 2d coordinates let's find the motion matrix(rotation and translation)

We need to normalize our x coordinates by using intrinsic matrix:

$$K = \begin{pmatrix} fS_x & S_\theta & O_x \\ 0 & fS_y & O_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1077.9 & 0 & 594.0 \\ 0 & 1077.9 & 393.3 \\ 0 & 0 & 1 \end{pmatrix}$$

So we need to go from homogenous pixel coordinates to homogenies image (plane) coordinates:

$$x_{normalize} = K_{\text{int}rinsic}^{-1} \begin{bmatrix} x'_{pixel} \\ y'_{pixel} \\ 1 \end{bmatrix}$$

Finding Essential Matrix:

$$x'^{T} Ex = 0, E = \begin{bmatrix} 0.0160 & 0.2605 & 0.0372 \\ -0.3866 & 0.0226 & 0.5907 \\ -0.0889 & -0.6506 & -0.0187 \end{bmatrix} \Rightarrow E = U * diag(0.7071, 0.7071, 0) * V$$

$$E_{normalized} = U * diag(1,1,0) * V = \begin{pmatrix} 0.0226 & 0.3684 & 0.0526 \\ -0.5468 & 0.0319 & 0.8354 \\ -0.1257 & -0.9201 & -0.0265 \end{pmatrix}$$

Now we need to find R and t. We have 4 possible solutions. We will choose one who has the majority of points in front of both cameras.

$$\widehat{u} \doteq \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \quad \in \mathbb{R}^{3\times 3}$$

$$R = \begin{pmatrix} 0.975 & -0.088 & 0.1993 \\ 0.086 & 0.996 & 0.0184 \\ -0.200 & -0.0008 & 0.9797 \end{pmatrix}, t = \begin{pmatrix} -0.9278 \\ 0.0466 \\ -0.3699 \end{pmatrix} \Rightarrow \hat{t} = \begin{pmatrix} 0 & 0.3699 & 0.0466 \\ -0.3699 & 0 & 0.9278 \\ -0.0466 & -0.9278 & 0 \end{pmatrix}$$

$$\hat{t} * R = \begin{pmatrix} 0.0226 & 0.3684 & 0.0526 \\ -0.5468 & 0.0319 & 0.8354 \\ -0.1257 & -0.9201 & -0.0265 \end{pmatrix} = E_{normalized}$$

Here is the epipolar lines for image 0000.jpg and 0001.jpg. Because of camera situations it is not so obvious that these lines are intersecting.

So I draw image 0005.jpg and 0000.jpg to show they are intersecting.

We can see they are intersecting at epipoles (where baseline and image plane intersect) and all points in one line, exist on the same colored line in other image.

Source images:

 #0	 #1
 #2	 #3


#4


#5


#6

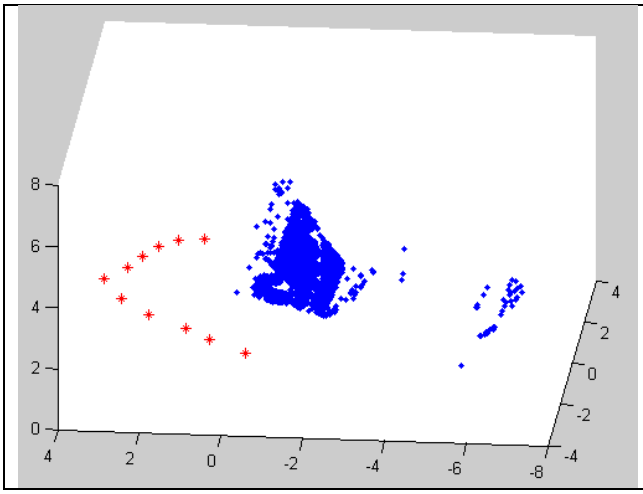

#7

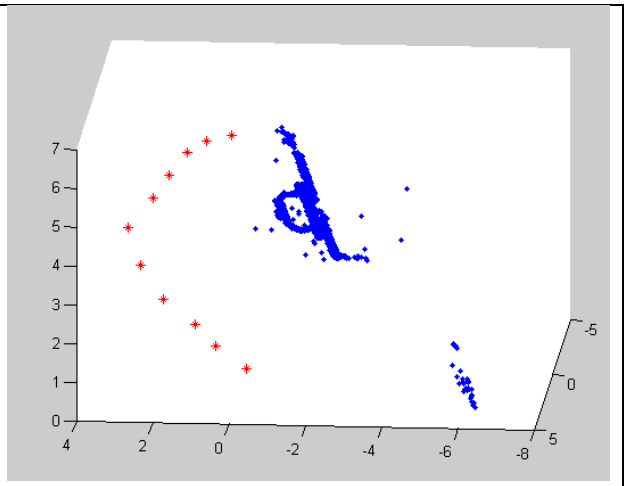

#8


#9

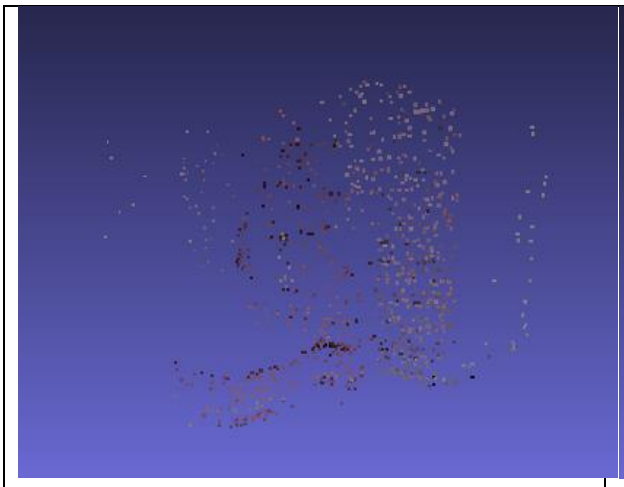

#10

Results:

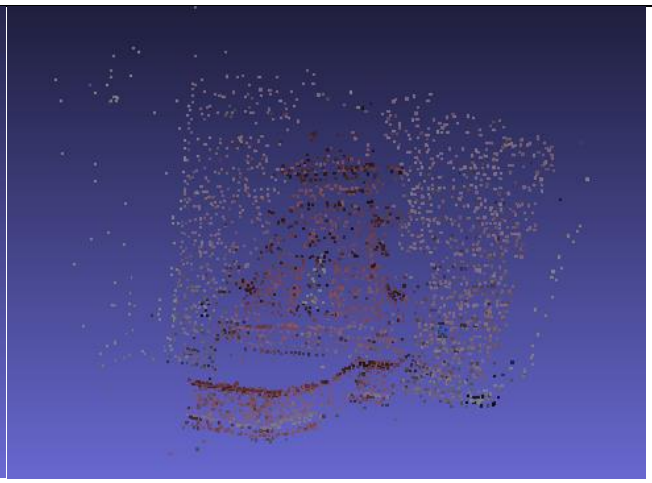




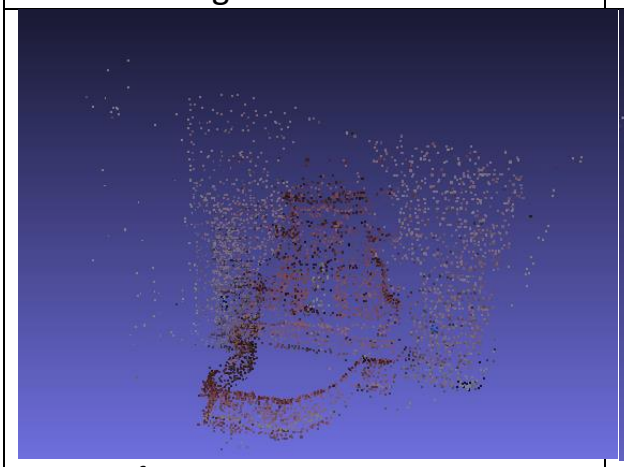
Red dots are camera locations.



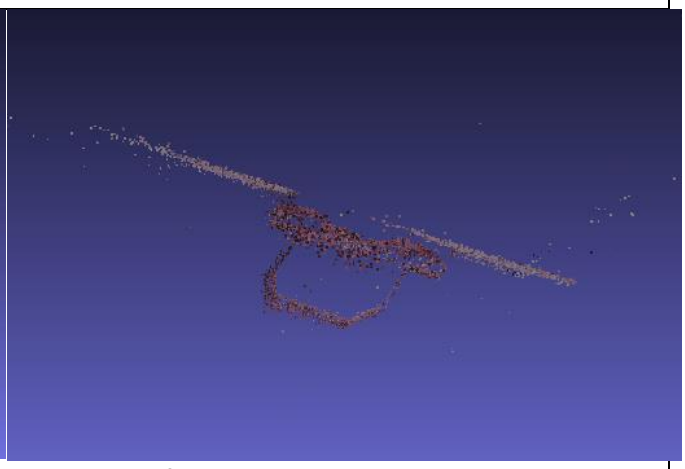
2 image reconstruction



After 6 images reconstruction



After 11 images reconstruction



After 11 images reconstruction