Lecture 13: (SLAM Part II)

Scribes: Qiyang Qian

1.1 Normalization

1.1.1 Problem with 8-point Algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1' & v_1' & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2' & v_2' & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n' & v_n' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

If different column has different magnitude, say column 1 is around 10000 and column 3 is around 100, higher weight might be placed on certain element of the f and yields poor results.

1.1.2 Normalized 8-point Algorithm

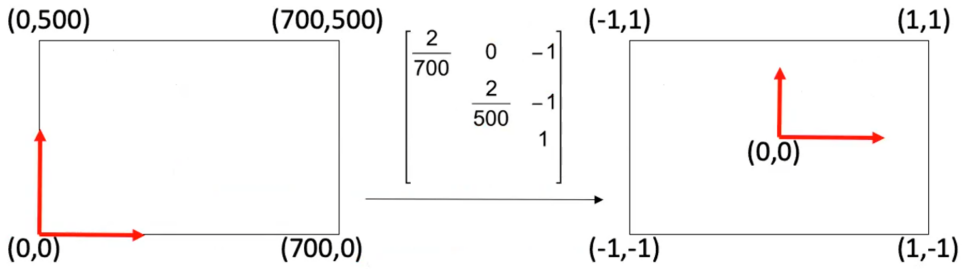

Figure 1.1: Normalization of an image

In this modified algorithm, all the images are normalized to have $(0, 0)$ as central and $(-1, 1)$, $(1, 1)$, $(-1, -1)$, $(1, -1)$ as four corners. Let this recentering and normalizing operation to image be T, then the input is transformed as $\hat{x}_i = Tx_i$, $\hat{x}_i' = Tx_i'$

Then, using the 8-point algorithm on $\hat{x}_i$ and $\hat{x}_i$ so that the normalized fundamental matrix $F = T'^T \hat{F} T$ is obtained as follow:

$$\hat{x}'^T \hat{F} \hat{x} = 0$$
$$(x'^T T'^T) \hat{F} (Tx) = 0$$
$$x'^T (T'^T \hat{F} T) x = 0$$

1.2 Structure from motion

Generally, structure from motion means we want to construct 3D point cloud of a scene from moving cameras. So a structure from motion can be breaking down into (a): Find the structure and (b): Know the motion

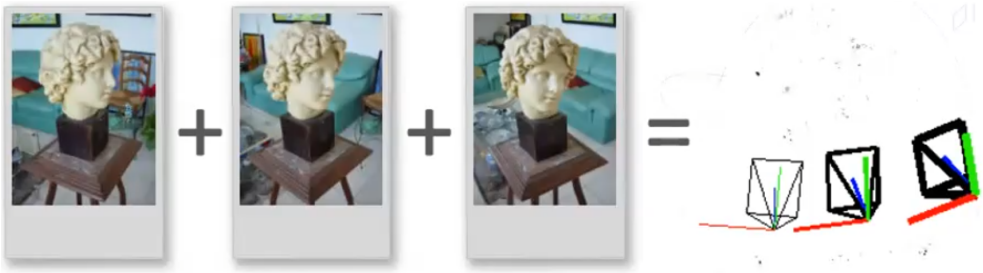

Figure 1.2: Example of Structure From Motion

1.3 Two-view structure from motion

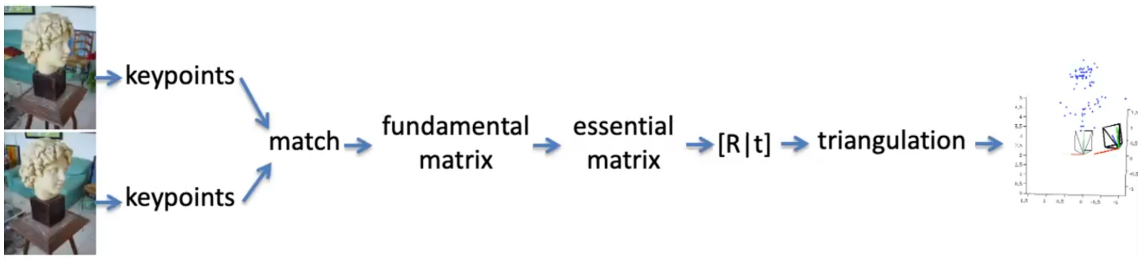

Figure 1.3: Pipeline of Two-view Reconstruction

1.3.1 Keypoints detection

For both images, find several keypoints and use a SIFT feature descriptor for each keypoint detected.

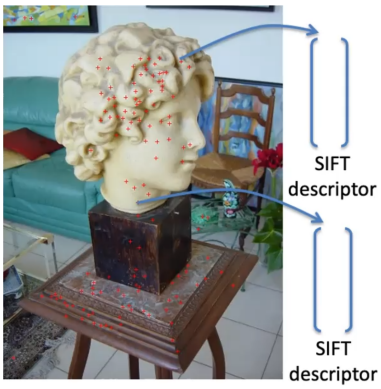

Figure 1.4: keypoint detected for one image

1.3.2 Point match for correspondences

Compare the different descriptors and try to match them.

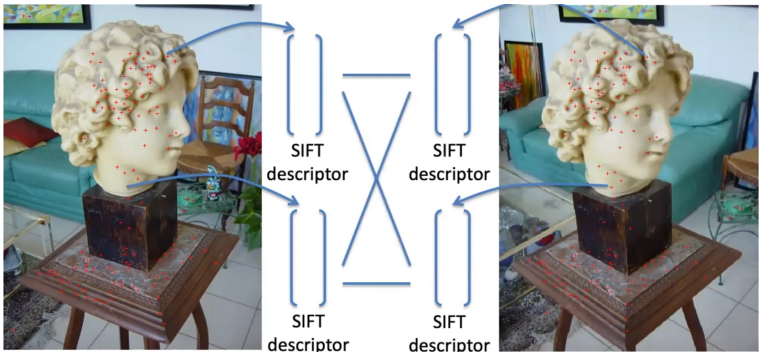

Figure 1.5: keypoints matching

1.3.3 Fundamental matrix

Using matched keypoints x_1 and x_2 on both images to find the fundamental matrix as $x_1^T F x_2 = 0$

1.3.4 RANSAC to estimate fundamental matrix

Iteratively picking 8 or more points to compute F. Suppose in total n F is computed and m pairs of keypoints matched. Find $F_i, i \in [0, n]$ that has most $x_{1i}^T F x_{2i} \approx 0$.

1.3.5 Fundamental matrix → Essential matrix

$$E = K_1^T F K_2$$
$$K = \begin{bmatrix} fm_x & s & P_x \\ & fm_y & P_y \\ & & 1 \end{bmatrix}$$

Where f is focal length, $m_{x,y}$ is camera pixel size, $p_{x,y}$ is principal point coordinate s, s is skew parameter

1.3.6 Essential matrix → Rotation and Translation

Use singular value decomposition to decompose E as:

$$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

Assume the first camera matrix is $P_1 = [I|0]$, then four choices for second camera matrix is:

$$P_2 = [UWV^T| + u_3]$$
$$P_2 = [UWV^T| - u_3]$$
$$P_2 = [UW^TV^T| + u_3]$$
$$P_2 = [UW^TV^T| - u_3]$$
$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where W is simply a rotation matrix by 90° for U and V. Finally, since normally the image points are in front of both camera, the desired solution is one that with maximal number of points in front of both cameras.

1.4 Multi-view structure from motion

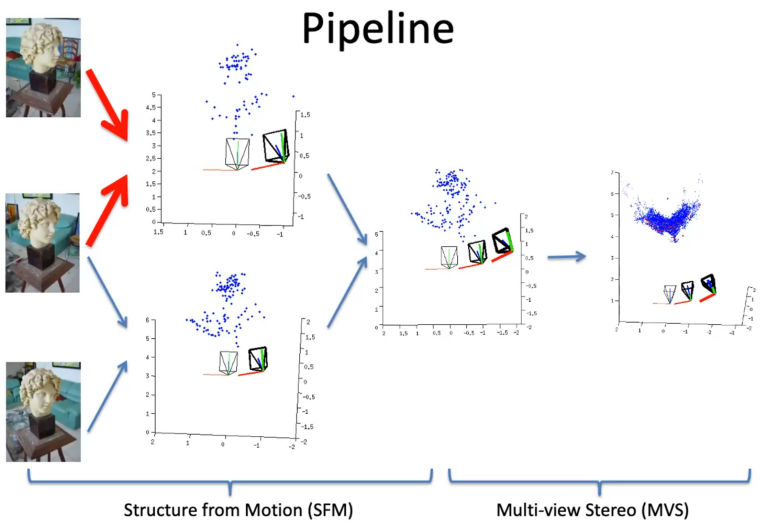

Figure 1.6: Pipeline for Multi-view SFM

1.4.1 Merge Two Point Cloud

Suppose 3 images exists as im_1, im_2, im_3, which means:

$$im_1,\ im_2 \rightarrow [R_1|t_1],\ [R_2|t_2]$$
$$im_2,\ im_3 \rightarrow [R_2|t_2],\ [R_3|t_3]$$

The problem is only one $[R_2|t_2]$ should exist, solution to it is called Bundle Adjustment

1.4.2 Bundle Adjustment

Assumption: All rotation and translations are correct and error are all in point cloud, so only need to jointly optimize rotation, translation and points.

Therefore bundle adjustment deals with minimizing sum of squared reprojection errors as

$$g(X,R,T) = \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}||P(x_i,R_j,t_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}||^2$$

Where w_{ij} indicates if point i is visible in image j, $P(x_i,R_j,t_j)$ is predicted image location and $\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}$ is the observed image location. Hence using a non-linear least squares can solve this optimization problem. In general, this optimization means that the reprojection must be close to the observation. When capturing data for reconstruction, several aspects to take care of, 1) Texture: in a plain wall, there might not be a lot of feature to detect or match. 2) Good lighting. 3) Subject don't move, otherwise the assumption of bundle adjustment fail. 4) No motion blur that might caused by camera fast moving and result in fail to detect and match keypoints. 5) Common overlapping, suppose camera are taking two image with no overlap, then since two images are too different, though keypoints can be detected, it'll be hard to match any keypoint from two images.

FAQ for Bundle Adjustment:

Q: Why do we need to estimate essential matrix?

A: To initialize non-linear optimization

Q: Can we optimize only Rotation and translation or only points?

A: Can, but since error exists in both side, better optimize together

Q: Time efficiency?

A: Can use sparsity to speed up

1.5 Large-scale structure from motion

Using the similar pipeline as multi-view structure from motion and create an image connectivity graph using keypoint matching, interatively find rotation and translation from two images and merge into the whole point cloud to incrementally reconstruct the scene.

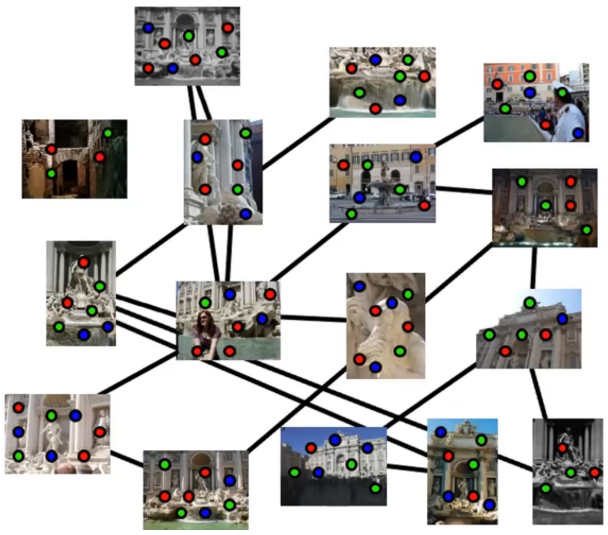

Figure 1.7: Keypoint matching

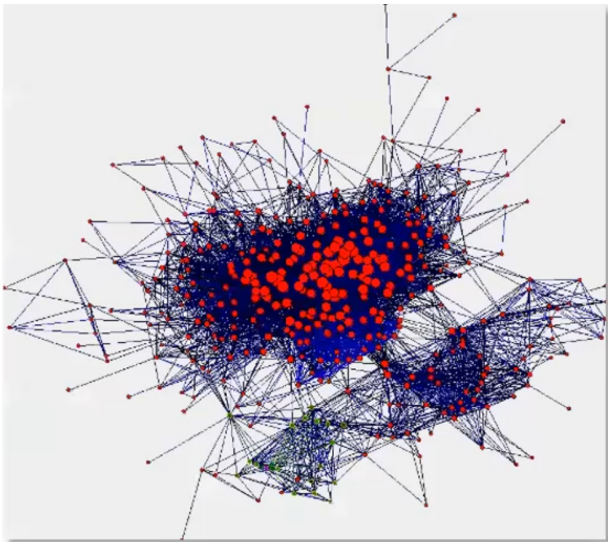

Figure 1.8: Image connectivity graph

Figure 1.9: Incremental structure from motion