16-822: Geometry Based Methods
Assignment 3

Andrew ID: sajaved

Question 1

Answers have been matched to GradeScope question list.

Question 2

Question 2.1

2.1.1

For an affine camera, the projection of the centroid of 3D points is the centroid of their 2d projections in the image.

2.1.2

Let the affine camera matrix be

$$M_a = \begin{bmatrix} A_{2\times3} & b_{2\times1} \\ 0_{1\times3} & 1 \end{bmatrix}$$

Let the two lines in the 3D world be L_1 and L_2 and the projections of the points lying on these lines onto the image using the above matrix as l_1 and l_2. Since the world lines are assumed to be parallel, they meet at a 3D point at infinity $P_\infty = [X, Y, Z, 0]$. Hence, in order to prove that the 2D lines are also parallel, we need to prove that they too are lines meeting at infinity. Projecting P_∞ using M_a, we get:

$$M_a P_\infty = \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix}$$

where $b = [b_1, b_2]$ Hence, the 2D lines in the image which are formed by the projection of the 3D parallel lines also meet at infinity. Therefore they too are parallel.

2.1.3

Consider 4 3D points P_1, P_2, P_3 and P_4, with the first two and the last two lying on two parallel lines in space respectively. Since the lines are parallel, we can write the following equation:

$$P_1 - P_2 = \lambda(P_3 - P_4)$$

Pre-multiplying with the affine camera matrix:

$$M_a P_1 - M_a P_2 = \lambda(M_a P_3 - M_a P_4)$$

$$\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} P_1 - \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} P_2 = \lambda(\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} P_3 - \begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} P_4)$$

Since the affine matrix does a linear transformation, we can write the vectors after solving the above equations as the projections p_1, p_2, p_3 and p_4

$$\begin{bmatrix} AP_1' + b \\ 1 \end{bmatrix} - \begin{bmatrix} AP_2' + b \\ 1 \end{bmatrix} = \lambda(\begin{bmatrix} AP_3' + b \\ 1 \end{bmatrix} - \begin{bmatrix} AP_4' + b \\ 1 \end{bmatrix})$$

$$p_1 - p_2 = \lambda(p_3 - p_4)$$

Hence the ratio of the lengths on the line segment is invariant when mapped using an affine camera.

2.1.4

Let the affine camera matrix be denoted as the following:

$$M_a = \begin{bmatrix} A_{2\times 2} & c_{2\times 1} & b_{2\times 1} \\ 0_{1\times 2} & 0 & 1 \end{bmatrix}$$

The two equations for the projections are:

$$x = A \begin{bmatrix} X \\ Y \end{bmatrix} + cZ + b$$

$$x' = A' \begin{bmatrix} X \\ Y \end{bmatrix} + c'Z + b'$$

Substituting the variables X, Y from one equation into the other, we get:

$$x' = A'A^{-1}(x - cZ - b) + c'Z + b'$$

Re-arranging the above, we get:

$$x' = Tx + Zd + e$$

where $T = A'A^{-1}$, $d = c' - Tc$ and $e = b' - Tb$ These three parameters only depend on the camera matrices. The above equation is of the form $x' = \lambda x + t$ where $Tx + e$ is an offset and d denotes the direction vector associating two parallel lines since it's a constant. Therefore, the epipolar lines and in turn the planes are parallel to each other.

2.1.5

In the previous equation, we can re-write it using the vector $\perp$ which is perpendicular to the direction d as follows:

$$(x' - Tx - e)^T d^{\perp} = 0$$

This is an equation of the form:

$$ax' + by' + cx + dy + e = 0$$

We can factorize this into the form of fundamental matrix equation

$$\mathbf{p}'^T F \mathbf{p} = \begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

Given the structure of the fundamental matrix above, we can see that changing the world coordinates using a 4×4 affine transformation, the camera matrix still stays an affine matrix. This is because the cameras are at infinity and the affine transformation will leave the coordinate system unchanged. Hence affine camera fundamental matrix is invariant to an affine transformation of the world coordinates.

Question 2.2

2.2.1

We derived the inequality relating the no of images (m), no of known parameters (K) and no of fixed parameters (K) for autocalibration with quadtratic equations in class

$$Km + f(m-1) \geq 8$$

Since skew is given as zero and the 4 other parameters of the intrinsics are fixed, we get

$$\begin{aligned} m + 4(m-1) &\geq 8 \\ m &\geq 2.4 \end{aligned}$$

Thus at least 3 images are needed

2.2.2

It can't be solved linearly since it's still a quadratic in the parameters of the intrinsic matrix.

2.2.3

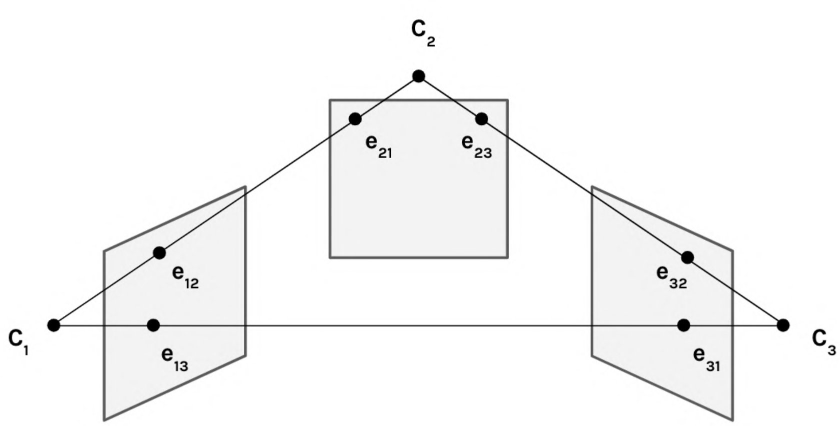
Since the principal points can vary but will be same across all images, we have the following relation:

$$m + 0(m-1) \geq 8$$

Thus at least 8 images are needed

Question 2.3

2.3.1



(a) Trifocal Epipoles

2.3.2

We need to show the following:

$$F_{ij}e_{ik} = e_{jk} \times e_{ji}$$

We know that the projection of the line connecting C_i and C_j is given by the line passing through e_{ji} and e_{jk} ($M_j(C_i \times C_k) = e_{ji} \times e_{jk}$). Furthermore, we also know that the camera center projects C_k projects to images i and k at e_{ik} and e_{jk} respectively. Therefore, the epipolar transfer of point e_{ik} using F_{ij} should satisfy the point e_{jk}. We also know that using F_{ij} for any point in image i will also satisfy the epipolar point e_{ji} in image j. Hence the epipolar transfer of point e_{ik} should satisfy both e_{ji} and e_{jk} and hence the equation $F_{ij}e_{ik} = e_{jk} \times e_{ji}$ is true.

2.3.3

The trifocal setup enforces the following relation on the 3 fundamental matrices:

$$e_{jk}^T F_{ij} e_{ik} = e_{ij}^T F_{ik} e_{kj} = e_{ki}^T F_{jk} e_{ji} = 0$$

In general, 3 independent fundamental matrices have 21 degrees of freedom, but due to the above constraint, we have 3 less degrees of freedom. Thus the F matrices for 3-camera setup are not independent and has 18 degrees of freedom.

Question 3

Question 3.1

The following major steps are executed for SfM and dense matching:

1. Set intrinsics
   (a) Find focal length from the EXIF file, else manually set it.
   (b) Use the focal length to specify the intrinsic matrix
2. Two-view SfM
   (a) Extract SIFT features on two camera views
   (b) Use MST based matching to find correspondences
   (c) Estimate fundamental matrix using matching points
   (d) Extract essential matrix, and in turn the two camera poses
   (e) Triangulate 3D points
   (f) Refine estimates with bundle adjustment over the two views
3. Graph merging
   (a) Sequentially merge the graph data structure storing the motion and structure information after each two-view SfM step
   (b) Refine estimates after each merge with bundle adjustment and outlier pruning
4. Dense matching
   (a) For each pair, Find dense matches using zero mean normalized cross correlation (ZNCC), storing them in a priority queue.
   (b) For all windows, keep deleting old points and adding new matches based on the ZNCC metric in the queue
   (c) Triangulate the dense points

Question 3.2

The principal points are assumed to be at (0,0). Changing the principal points from their correct position results in linear shifts in the projected image points. The (0,0) assumption will get violated if the camera center is not at the origin. This can happen due to multiple reasons like shifting the image coordinate axis or cropping/resizing the images etc.

Question 3.3

The 5 main fields of the SfM graph are:

1. Focal length (f): Double scalar. Stores the focal length for the camera. Used for defining the intrinsic matrix.

2. Camera poses (Mot): $3 \times 4 \times 2$ matrix. Stores the camera pose for each view (extrinsic matrix). Used for specifying the camera motion.

3. 3D points (Str): $3 \times N$ matrix. Stores the 3D points in space. Used for specifying the 3D scene structure.

4. Keypoint matches (ObsIdx): $2 \times N$ matrix. Stores the 2D feature matches between 2 views. Used for finding feature match indices.

5. Keypoint matches (ObsIdx): $2 \times 2N$ matrix. Stores values of the feature matches between 2 views. Used for fetching the keypoint value.

Question 3.4

The results for the dense 3D reconstruction are shown below:

Image 1

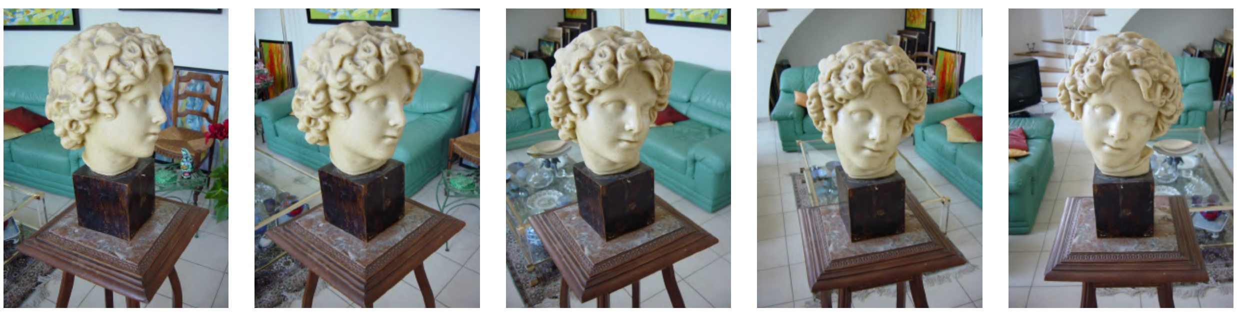


Figure 2: Original Images from multiple views

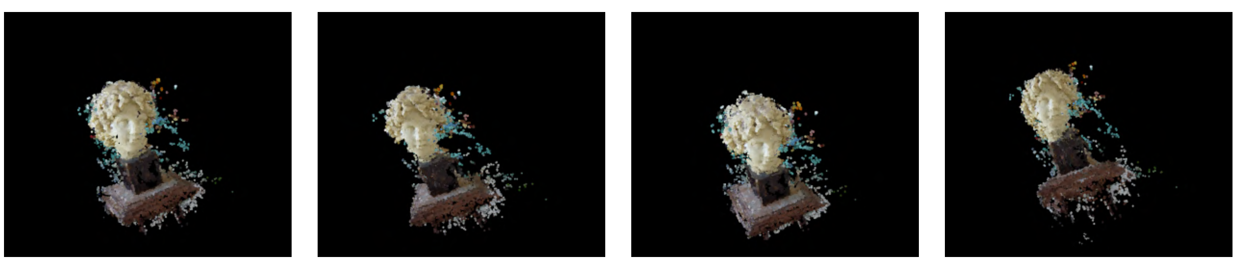


Figure 3: Dense Reconstruction

Image 2

Figure 4: Original Images from multiple views

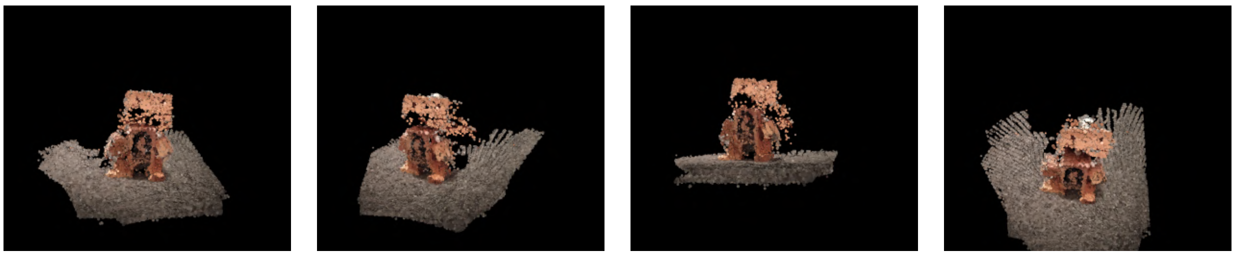


Figure 5: Dense Reconstruction

Image 3

Figure 6: Original Images from multiple views

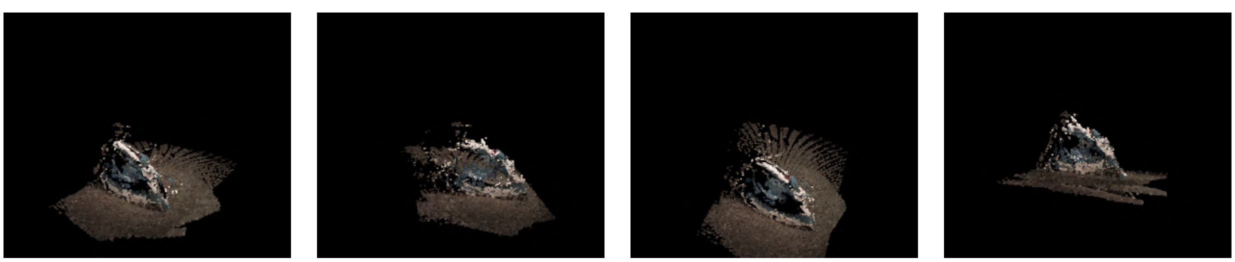


Figure 7: Dense Reconstruction

Online resources used

- Affine epipolar geometry

There are various unstable parts of the pipeline which introduce errors into the final reconstruction, the most important being the finding the SIFT feature correspondences and their matching. A small list of these factors which can also help a non-expert take good images for reconstruction would be:

- Issue: Bad feature matches
  Suggestion:

1. Take images which have texture and structure in the scene
2. Take images which don't have varying illumination across multiple views
3. Take images which don't have shiny or transparent surfaces

- Issue: Bad initial intrinsic estimates
  Suggestion:

  1. Find the camera lens details for the images and update the EXIF file
  2. Run with different initial values of the focal length parameter

- Issue: Bad pose estimation
  Suggestion:

  1. Take images which have some minimal structure in the background instead of a plain background as it can help localize the camera better
  2. Not take camera views which have a huge motion from the other image sequences
  3. Not take images where the objects have moved between sequences

Question 3.5

The reprojection errors for the above sequences are as follows:

Scene	**Total Reprojection Error**
$Head$	60.354
$RIRobo$	42.706
$Iron$	56.499

Question 3.6

The output for the reprojection point visualization is shown below:

Image 1

Figure 8: Reprojection Point Visualization

Image 2

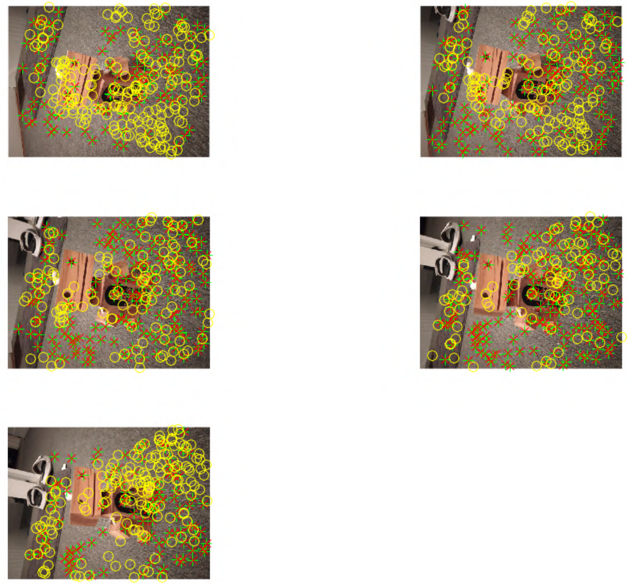


Figure 9: Reprojection Point Visualization

Image 3

Figure 10: Reprojection Point Visualization

Question 3.7

The bundle adjustment objective jointly optimizes for the motion, structure (and also the intrinsics depending on the flags). The function it optimizes is the sum of the reprojection errors for all the view pairs for all the observable matching keypoints. The general form of objective function being optimized with the full bundle adjustment is given by:

$$\min_{\mathbf{X},\mathbf{M}} \sum_{i=1}^{I} \sum_{n=1}^{N} \|\mathbf{x}_{in} - Proj\left(X_n; M_i\right)\|_2$$

where X_n is the n^{th} 3D world point, M_i is the camera matrix from i^{th} view, x_{in} is the observed pixel point in the i^{th} image for the n^{th} feature. I denotes the total no of views and N denotes the number of features. The function $Proj(X_n; M_i)$ does a 3D to 2D projection of the point X_n using the the camera matrix M_i. The camera matrix can also include the parameters of the intrinsic matrix (the focal length in our case).