Gebze Technical University
Department of Computer Engineering
BIL 565 / BIL 463
(Introduction to) Computer Vision
Spring 2022
Optional HW
15 Jun 2022
No late submissions!

In this work, we will implement the affine structure from motion algorithm as we discussed in the lectures. We have modified this homework from Rob Fergus of NYU.

First, you will create your dataset as follows

- Use your mobile camera to take a 30 second video of an object on a table. You should move your camera in a way that the assumptions of the structure from motion by factorization algorithm is not violated.
- Use your favorite feature detector to detect and match points between video frames.
- You should have more than 100 points tracked on at least 20 frames. There should be enough movement between the frames.
-

After you have your data, you will follow the SFM steps that we have learned in the class. We will only do affine reconstruction, thus the projection matrix of each camera i will have following form:

$$P^i = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} M^i & t^i \\ 0 & 1 \end{pmatrix}$$

where M^i is a 2 by 3 matrix and t^i is a 2 by 1 translation vector.

So given m = 20 views and n = 100 points, having image locations x^i_j , where j = 1, . . . , n, i = 1, . . . , m, we want to determine the affine camera matrices M^i , t^i and 3D points X_j so that we minimize the reconstruction error:

$$\sum_{ij} ||\mathbf{x}_j^i - (M^i\mathbf{X}_j + t^i)||^2$$

We do this in the following stages:

- Compute the translations t^i directly by computing the centroid of points in each image i.
- Center the points in each image by subtracting off the centroid, so that the points have zero mean
- Construct the 2m by n measurement matrix W from the centered data.
- Perform an SVD decomposition of W into UDV^T. For SVD, use OpenCV SVD::compute
- The camera parameters M^i can be obtained from the first three columns of U multiplied by D(1 : 3, 1 : 3), the first three singular values.
- The 3D world point locations are the first three columns of V.

You can verify your answer by projecting 3D points to an image plane using OpenCv function **`projectPoints.`** Use different rotation values (**`rvec`**) for this function to show the reconstructed points.

- Write a report to describe your step by step results.
- Use Python 3D visualization libraries to show your reconstructed 3D object points from different angles.
- You may have to calibrate your camera or you may assume you have identity intrinsic camera.