M4 Project
Week 5: Structure from motion

Group members:

José Manuel López, Alex Martin, Marcos V. Conde, Oriol Catalan Yarza

Group ID: 4

1. Introduction

The main goal of the final lab of the module can be separated into two goals:

1. Understanding of the structure-from-motion algorithm pipeline.
2. Apply structure-from-motion to a real set of images of a building.

Structure-from-motion algorithms allow the recovery of the camera matrices for each view to later and calculate a sparse set of 3D points. With enough views of the same scene, reconstruction of the 3D scene should be attainable.

For our test-cases, the structure for the Structure-from-motion algorithm corresponds to the facade of a building and all the elements captured in the scene.

2. Computation of Projective cameras (n=2)

First, we will begin with a projective reconstruction. To perform it, we just have to extract the camera matrices from the fundamental matrix and perform a triangulation to reconstruct the 3D points from the keypoints at both imaged in homogeneous coordinates. To recover the camera matrices, we will assume that the first one, P , is the canonical one and the second camera matrix will have the following form:

P' = [SF | e']

Where S is any skew-symmetric matrix. It is a good choice to have S= [e'x]. We can obtain e', the epipole of the second matrix, as the null-vector of the fundamental matrix transposed as we have that $e'^{T} \cdot F = 0$.

From applying the above, we obtained a reprojection error of 10^-7 and the following plot.

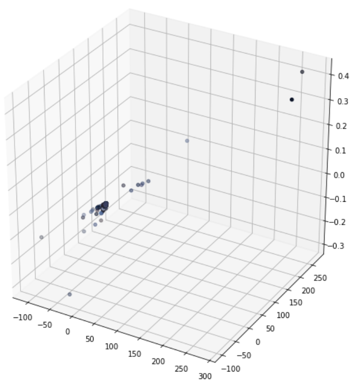

Fig1: projective reconstruction

2.1. Estimate the reprojection error

The reprojection error named above is computed in the following way:

$$error = \underbrace{||PX - \mathbf{x}||}_{\text{Error in Image 1}} + \underbrace{||P'X - \mathbf{x}'||}_{\text{Error in Image 2}}$$

Where P and P' are the camera matrices of the first and second cameras, x and x' are the keypoints in homogeneous coordinates for the first and the second views and X are the 3D triangulated points.

2.2. Affine Rectification

Affine rectification consists in locating the projected plane at infinity. To achieve this, we must assume we already know p (projected plane at infinity). Knowing p we must find a transformation that maps p to the canonical plane at infinity. In short, we want to find H that:

$$H^{-T}{}_{p} = (0,0,0,1)^T$$

This transformation is given by:

$$H_{a\leftarrow p} = \begin{pmatrix} I & \mathbf{0} \\ \mathbf{p}^T & 1 \end{pmatrix}$$

The transformed H is then applied to each point and two cameras with the following equation:

$$P_i' = P_i H_p^{-1} \text{ , } H_p X$$

Before computing p we need first to compute vanishing points (points at infinity). Since we can form a plane with 3 points. Not only do we need to find these vanishing points, in order to find them in 3D space, we need to match them. Knowing 3D lines are parallel in the plane, the intersection of those parallel lines on the 3D space provides a point in the plane at infinity. In other words, the newfound point on the 3D space is the vanishing point for both lines.

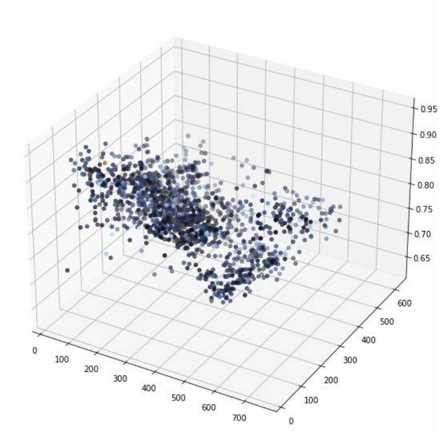

Fig 2: affine reconstruction

In this case, the reprojection error is also a very low number of 10^-7, but we can observe that the points are more or less organized in what seems a tiled plane which may be the facade of the images.

2.3. Metric Verification

The key to metric reconstruction is the identification of the absolute conic Ω_∞. The absolute conic, Ω_∞, is a planar conic lying in the plane at infinity π_∞. Then we will have to apply an affine transformation to the affine reconstruction such that the identified absolute conic is mapped to the absolute conic in the standard Euclidean frame.

The metric reconstruction can be achieved by applying a 3D transformation:

$$H_{e\leftarrow a} = \begin{bmatrix} A^{-1} & 0 \\ 0^T & 1 \end{bmatrix}$$

where A is obtained by Cholesky factorization $AA^T = (MT\ \omega M)^{-1}$ and ω is the image of the absolute conic. To obtain the image of the absolute conic, we can use different constraints, such as:

- **Scene orthogonality:** using two vanishing points v1 and v2 we can have a linear constraint on ω which is $v^T 1 \cdot \omega \cdot v2 = 0$
- **Known internal parameters:** the knowledge about some restrictions on the internal parameters contained in K may be used to constraint or determine the elements of ω.

Using all the constraints, we can build the following matrix that contains all the linear constraints that satisfies A·ω=0.

$$A = \begin{pmatrix} u_1v_1 & u_1v_2 + u_2v_1 & u_1v_3 + u_3v_1 & u_2v_2 & u_2v_3 + u_3v_2 & u_3v_3 \\ u_1z_1 & u_1z_2 + u_2z_1 & u_1z_3 + u_3z_1 & u_2z_2 & u_2z_3 + u_3z_2 & u_3z_3 \\ z_1v_1 & z_1v_2 + z_2v_1 & z_1v_3 + z_3v_1 & z_2v_2 & z_2v_3 + z_3v_2 & z_3v_3 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

The solution ω = (ω11, ω12, ω12, ω22, ω23, ω33) is the null vector of A. After applying the transformation to the affine rectified camera matrices and the reconstructed points we obtained the following plot.

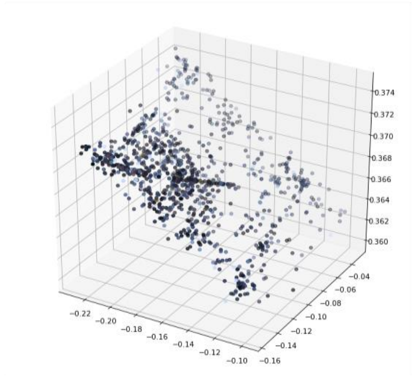

Fig 3: 3D points with the metric reconstruction

In this case the reprojection error increases a lot to number around 10^8 which may be due to an implementation error, but the plot doesn't vary a lot from the affine reconstruction, and we can observe qualitatively that the points are arranged in a more flat shape with a central line that may be the transition between the two buildings of the facade.

3. Bundle Adjustment

After gathering all the data, we need to refine the estimates with a non-linear optimization process, such as the following minimization.

$$\min_{\mathbf{R}^i, \mathbf{X}_p} \sum_{i=1}^{i} \sum_{p=1}^{P} d(\mathbf{R}^i \mathbf{X}_p, \hat{\mathbf{x}}_p^i)^2$$

The distance function used to refine the measurements is the previous defined reprojection error. By updating the values of the 3D points to minimize the difference between the projection of this points to the images and the coordinates obtained with the keypoint detector. To do so, we used the PySba module after adapting the obtained data for the function to perform the bundle adjustment.

After many tries, we couldn't implement correctly the method, which may be caused because of an error in the previous calculations of the metric rectification.

4. Conclusions

This week we tried to solve a very simple structure-from-motion scenario, focusing on the case of just two images. The qualitative assessment of results was a challenge using only as input as two images, it is hard to visualize the computed point cloud and yield conclusions. Judging mainly by quantitative measures, like the re-projection error, our current implementation shows some issues in the metric rectification part, as this error increases dramatically. For the bundle adjustment, we couldn't perform the proposed experiments, but we should have obtained a more clear result and a lower number in the reprojection error.