Project III: Structure-from-Motion in 2D

Patrícia Marques
Chalmers University of Technology
Instituto Superior Técnico
patrciam@chalmers.se

Abstract

We implement a Structure-from-Motion pipeline in 2D for four datasets, each with six images from different viewpoints of the same scene. To estimate the homography between two images, it is used a RANSAC-based method with the DLT algorithm on four features correspondences. With access to the ground truth homographies, a comparison is made between the ground truth map and the estimated one. For all datasets, the average residual is less than 10 and the images are effectively stitched together in a planar surface.

1. Mandatory Part

1.1. Method

First, we run `prepare_images.m` which is responsible for setting up the image data. For each folder 'bark', 'wall', 'graf' and 'boat', it iterates through each of the six images within the folder. Each image is stored alongside its SIFT features and descriptors. The true homographies from the first image into each other are also stored in a structure.

Algorithm 1 2D Structure-from-Motion

```
Input: Set of 6 images $\{I_1, I_2, \ldots, I_6\}$
Output: Sparse map
Extract SIFT features and descriptors
Match SIFT features between pairs of images
Estimate homographies $H_{ij}$ (Algorithm 2)
Select a reference image $I_{\text{ref}}$
Compute global homographies $H_i$ with respect to $I_{\text{ref}}$
Map SIFT features into the global coordinate system
Visualize the resulting 2D point cloud
```

Algorithm 1 describes how the Structure-from-Motion pipeline in 2D is implemented. Images in the same folder represent the same scene from different viewpoints. Feature matching is perfomed for each pair of images. There exists a 2D projective transformation that maps these corresponding points, then, the images are related by a homography.

To estimate the homographies between images, we perform a RANSAC-based method, where we fit a homography transformation matrix to a subset of four correspondences out of the matching features. These fitting is based on the Normalized Direct Linear Transformation (DLT) algorithm presented in section 4 of the book [1].

Algorithm 2 Estimate homographies H_{ij} using RANSAC

```
Input: Matching SIFT features between $I_i$ and $I_j$
Output: Homography $H_{ij}$ from $I_i$ to $I_j$ and inliers
while probability of missing correct model $< 0.05$ do
    Subset of 4 correspondences/pairs of SIFT features
    Estimate $H_{ij}$ using the DLT algorithm
    Estimate support of the model based on threshold
    if more inliers than previous best model then
        Update the best model
        Update number and ratio of inliers
    end if
end while
```

1.1.1 Global Homographies

Next, our goal is to select a reference image, providing the global coordinate system for the sparse map. To do so, we consider the homography H_{ij} from I_i to I_j with the largest number of inliers, and select image I_j as the global reference. The selected H_{ij} is the global homography H_i. We consider images I_i, I_j as already processed at this point.

After establishing the reference image, we calculate the global matrices for the remaining images. Among the unprocessed images I_i and the already processed I_j, we select the homography H_{ij} with the highest number of inliers. The global homography matrix H_i is then computed as $H_j H_{ij}$. We mark image I_i as processed and continue iterating.

Given the matrices $H_{1j_{\text{truth}}}$, the ground truth global homographies $H_{i\text{truth}}$ are computed as $H_{1\text{ref}\text{truth}} H_{1j}{}^{-1}_{\text{truth}}$.

1.1.2 Sparse Map of the Scene

For each folder, using the ground truth and the estimated global homographies separately, we map the SIFT features found in each image into the global coordinate system. To perform the projective transformation, the 2D points are homogenized and, after applying the H_i, de-homogenized.

1.2. Experimental Evaluation

For all datasets, the average residual is below 10, in terms of distance in pixels, showing a high precision for the estimated sparse map. Such results were possible by choosing appropriate thresholds, varying between 3 and 50.

Notice how the 'wall' and 'graf' datasets correspond to higher thresholds used in the RANSAC-based method for homography estimation. This disparity is due to the difficulty in terms of time complexity in finding optimal models, in both the 'wall' and 'graf' datasets, when being more restrictive on the calculated errors. Even with a higher threshold, the latter datasets are the ones that have the pair of images that correspond that require most interactions to estimate a homography between them.

Indeed, the 'graf' and 'wall' show changes in viewpoint, while 'bark' and 'boat' show zoom and rotation changes, which are less complex geometric transformations between images. When the image suffers a less significant perspective transformation, without changes in viewpoint, RANSAC is quicker finding a good homography estimate.

Dataset	Threshold	Iterations	Avg Residual
bark	3	3 - 241	4.11
wall	50	2 - 22 545	5.54
graf	50	6 - 310 773	5.65
boat	3	8 - 3 524	7.76

Table 1: For each dataset, inlier threshold used in the RANSAC algorithm, the range of maximum number of iterations and the average residual between corresponding points in the estimated and ground truth maps. Note that the maximum number of RANSAC iterations differs according to the images being used for the homography estimation.

The RANSAC algorithm encountered the most difficulty, i.e, more interactions were required, for the homography estimation between images 1 and 6, for both the 'bark' and the 'wall' datasets, and between images 2 and 6, for both the 'graf' and 'boat' datasets. Such observations make sense, given that such pairs of images suffer a quite complex transformation between one another.

The reconstructed 2D sparse maps, both the estimated and the one based on the ground truth homographies, can be seen in Figures 1, 3, 5 and 7. Visually, the estimated reconstructions are quite comparable to the ground truth ones.

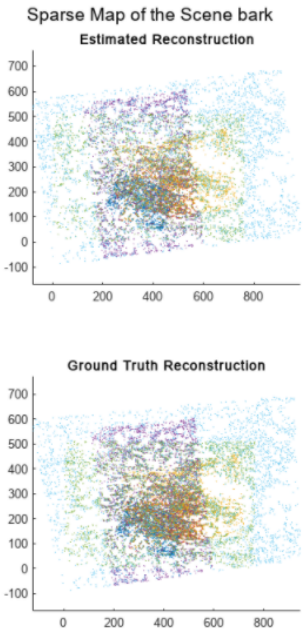

Figure 1: Reconstruction 2D sparse map of dataset 'bark'. Image 5 provides the global coordinate system. Each color corresponds to SIFT features from the same image.

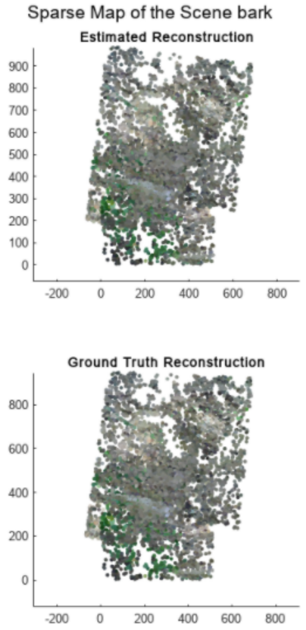

Figure 2: Reconstruction 2D sparse map of dataset 'bark'. Image 5 provides the global coordinate system.

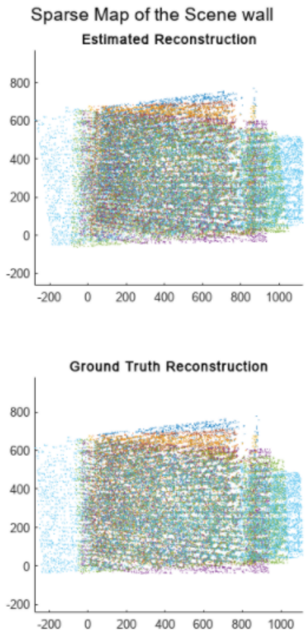

Figure 3: Reconstruction 2D sparse map of dataset ’wall’. Image 3 provides the global coordinate system. Each color corresponds to SIFT features from the same image.

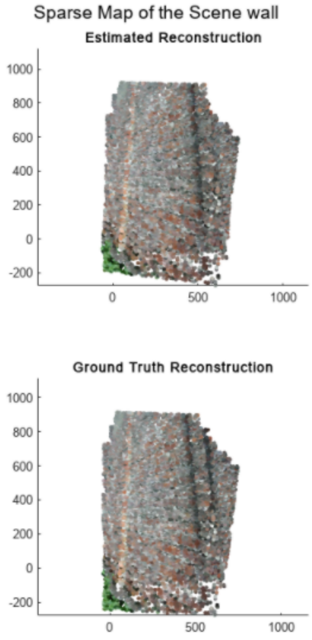

Figure 4: Reconstruction 2D sparse map of dataset ’wall’. Image 3 provides the global coordinate system.

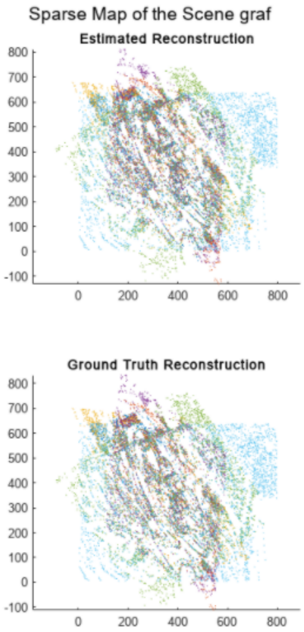

Figure 5: Reconstruction 2D sparse map of dataset ’graf’. Image 6 provides the global coordinate system. Each color corresponds to SIFT features from the same image.

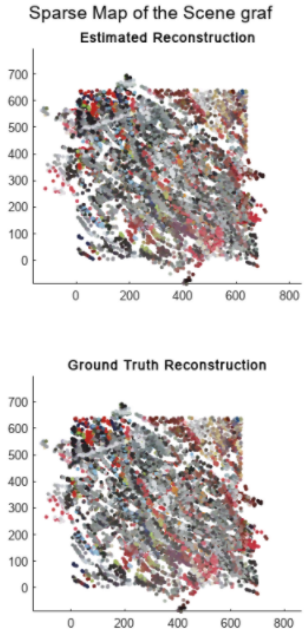

Figure 6: Reconstruction 2D sparse map of dataset ’graf’. Image 6 provides the global coordinate system.

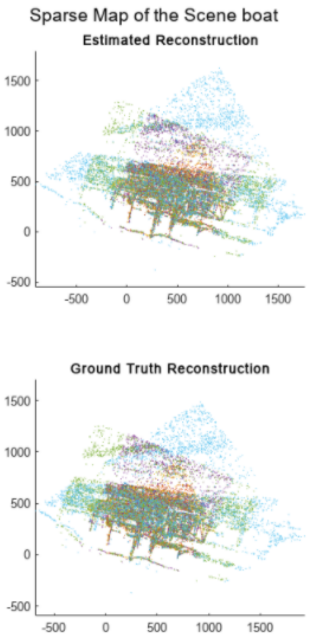

Figure 7: Reconstruction 2D sparse map of dataset ’boat’. Image 2 provides the global coordinate system. Each color corresponds to SIFT features from the same image.

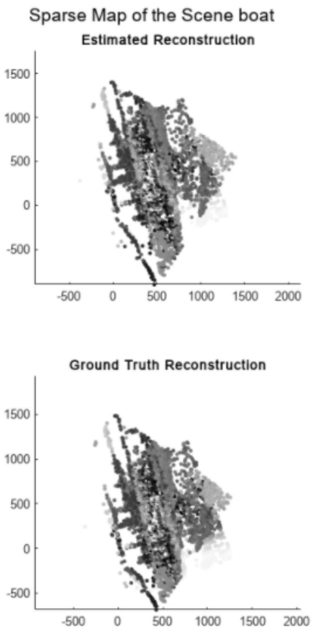

Figure 8: Reconstruction 2D sparse map of dataset ’boat’. Image 2 provides the global coordinate system.

2. Theoretical Part

2.1. Bundle Adjustment in 2D

Let $\mathbf{x}_j = (x_j, y_j)$ be a 2D point in the global coordinate system, $\mathbf{x}_{ji} = (x_{ji}, y_{ji})$ the corresponding point in the coordinate system of image I_i and H_i the homography from the global coordinate system to the coordinate system of I_i.

We perform the projective transformation: homogeneize the coordinates of $\mathbf{x}_j$, apply the transformation $H_i\mathbf{x}_j$ and dehomogeneize to get $\mathbf{x}_{ji_{\text{est}}}$, which consists of the estimated coordinates of $\mathbf{x}_j$ in the coordinate system of the image I_i.

The reprojection error for this particular point in image I_i is calculated as $e_{ij} = ||\mathbf{x}_i - \mathbf{x}_{ji_{\text{est}}}||$. During Bundle Adjustment, the goal is to minimize the sum of squared reprojection errors over all sets of M 2D points of all N images:

$$\min_{H_1,H_2,\dots,H_N} \sum_{i=1}^{M}\sum_{j=1}^{N} e_{ij}^2 = \sum_{i=1}^{M}\sum_{j=1}^{N} ||\mathbf{x}_i - \begin{pmatrix} \frac{H_i^1 \mathbf{x}_{j\text{Hom}}}{H_i^3 \mathbf{x}_{j\text{Hom}}} \\ \frac{H_i^2 \mathbf{x}_{j\text{Hom}}}{H_i^3 \mathbf{x}_{j\text{Hom}}} \end{pmatrix} ||^2$$

2.2. Generative models

(i) We optimize the Empirical Lower Bound (ELBO) rather than $\log(p(\text{x}))$ when training a VAE. Computing $\log(p(\text{x}))$ explicitly requires knowing the posterior distribution $p(\text{z}|\text{x})$. Simply put, this distribution is too complex and computationally intensive to compute directly. Since the ELBO provides a lower bound on the $\log(p(\text{x}))$, by maximizing the ELBO, we indirectly improve $\log(p(\text{x}))$.

The ELBO divides the optimization problem into two sub-problems: maximize the likelihood of $p(\text{z}|\text{x})$ (reconstruction loss) and minimize the KL divergence between between $q(\text{z}|\text{x})$ and $p(\text{z})$ (similarity between learned and prior distribution). Both are manageable to optimize.

(ii) VAEs use both encoding and decoding processes. The encoder maps input data to a probability distribution in the latent space. The decoder generates data by sampling from this latent space, which might be Gaussian.

Diffusion probabilistic models do not involve encoding/decoding processes or latent variables. They model the transition from the initial data distribution to the final data distribution through steps of adding/removing noise.

2.3. Triangulation

(i) Given a set of k images with projection matrices P_i and 2D pixel positions x_i, we get k camera equations $\lambda_i x_i = P_i X, \lambda > 0$, where X is the 3D reconstructed point (three unkowns). Each point adds three equations and an extra unkown λ_i. Since we have $3+k$ unknowns, the system is overdetermined for $k \geq 2$ and underdetermined for $k = 1$. Considering this, for $k \geq 2$, we could discard equations to get a quadratic system and later use a minimal solver.

However, since it is common that image points cannot be measured exactly, it might be useful to keep all $3k$ equations and use least mean squares to find the best estimate of X. This is done by minimizing the reprojection error (difference between the true positions and predicted points by projecting the estimated X back onto the k image planes).

(ii) There is indeed a case where there are several 3D points that project exactly to x_1 in the first image plane and to x_2 in the second image plane. The projection lines l_i from each x_i intersect at X. Any point lying on the line l_i will project exactly to x_i in image i, resulting in ambiguity. If this were to happen, we cannot derive the exact positions of points x_1 and x_2 without additional constraints.

References

[1] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.