Monocular 3D Scene Reconstruction

Hans Magnus Espelund Ewald
Stanford University
Department of Electrical Engineering
hmewald@stanford.edu

Abstract

Structure from motion is the most widely used principle for obtaining 3D scene information with a standard monocular camera. In this report we present an explorative project into a simple structure from motion processing system using two-view geometry and keypoint matching. The resulting system shows some promise, but remains quite unstable. We show some preliminary results and make a few suggestions on useful improvements.

1. Introduction

In this project we seek to approach the problem of 3D scene reconstruction with a simple solution that estimates both structure and motion. An efficient system of this kind is important in many respects, allowing machines to extract full geometric maps of unknown environments, and also keep track of their own positions within that environment. One interesting application of this technology is in autonomous robotics.

1.1. Related Work

Naturally, there has been much work on various solutions to this problem. The use of specialized depth cameras is one way to approach it, e.g. with light field cameras, stereo sensors or time-of-flight cameras. Directly obtaining depth information means that the 3D location of any image point captured is more or less immediately identifiable from each image - a big advantage. Most systems also make use of inertial sensors to support their estimations of camera pose changes. Finally, insights from multi-view geometry and Structure from Motion (SfM) techniques are the common tool that is used to generate 3D scene information from multiple images. Contemporary 3D mapping and localization systems generally combine all three of these tools into a single effective solution, for example as seen in Google's Project Tango devices.

In this project we focus on the latter of these three technological aspects of 3D reconstruction. 3D computer vision is a well explored topic in sources such as [4], [6] and [5]. These works will serve as comprehensive sources for this work.

1.2. Project Goal

With this premise, we now define our project goal to be the exploratory implementation of a simple structure from motion technique with a basic camera. The idea is find out how viable this reduced approach is in comparison to the high-end solutions mentioned above. We can expect our system to be very efficient in terms of computation and hardware required, so the interesting aspect is whether or not the performance will be able to hold up. We hope to gain some insight into the challenges that arise from this problem, and thereby to allow future improvements and new solutions.

2. Proposed Technique

Here we present the processing system that constitutes our simple monocular approach. The theory is based on the following geometric equation:

$$u_i x_i' = K R_i (X - t_i) \tag{1}$$

This formula describes the mapping of any 3D point $X \in \mathbb{R}^3$ onto a set of homogenous image coordinates $x_i' \in \mathbb{R}^3$ by a camera with the intrinsic matrix $K \in \mathbb{R}^{3\times3}$, positioned at the point $t_i \in \mathbb{R}^3$ and oriented according to the rotation matrix $R_i \in \mathbb{R}^{3\times3}$. Note that u_i is a positive scalar that results from norming the third component of x_i' to 1. t_i and R_i together form the camera pose, and we will refer to each pair of these values as such.

With this relation between the image coordinates of 3D points from different camera views, we have a clear formulation of our goal. For a given set of images taken from different and unknown camera poses, our approach needs to identify the poses representing the camera's motion. This part of the problem is known as visual odometry. Then we need to find results for 3D points in the scene as well.

Our simple and fast approach considers two images at a time and thus generates a sequential estimate of the pose:

$$R_{i+1} = R_{i+1,i} \cdot R_i \ , \ t_{i+1} = t_{i+1,i} + t_i \quad \text{with } R_0 = I_3, t_0 = 0 \tag{2}$$

This means we need to find a solution for estimating the pose change R_{ji}, t_{ji} between two given images i and j.

2.1. Keypoint Detection and Matching

Our solution to this pairwise odometry problem is based on the identification and matching of visual keypoints in the image. We use the well-established SIFT (scale-invariant feature transform) method to identify the image coordinates of matching points x'_i and x'_j, which are both captured from the same 3D point X. The method uses the concept of scale-space in order to make its features scale-invariant, and the feature descriptors are computed from the local gradients present in the neighborhood around the keypoint. Before the descriptor is extracted, the orientation of the keypoint is also set to match the direction of the image gradient in the point, meaning that the features are also rotation-invariant. These properties make SIFT keypoints suitable for our problem here. Refer to [2] for the original source on the SIFT method. Figure 1 shows a test image with the positions, scales and orientations of detected SIFT features.

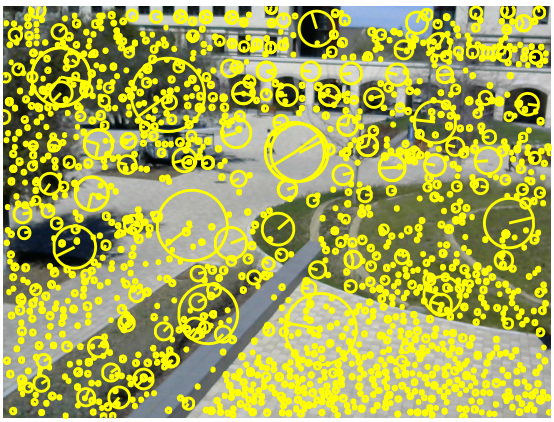

Figure 1. Example image with SIFT keypoints identified and marked.

2.2. Homography Estimation

The next part of our solution is to find a linear homography transform between the image points of images j and i:

$$x'_j = H_{ji} x'_i \ , \ H_{ji} \in \mathbb{R}^{3\times3} \tag{3}$$

The homography we use here can project the image i onto the image j by introducing translation, rotation, rescaling and perspective change. This yields the best results when the images have a large planar surface, but it is a reasonable approximation in most other cases as well. The homography matrix has 8 degrees of freedom, meaning that a unique solution can be computed from 4 point matches between the two images. Provided that the environment has enough SIFT-detectable features available, the number of keypoint matches is generally far higher than 4. This large availability of information allows for statistical methods to be used to increase robustness. This is done through the random sample consensus (RANSAC) algorithm, which samples random groups of 4 keypoint matches and computes the resulting homography. The quality of each randomly computed homography is determined by the amount of matching keypoint pairs that it correctly projects between the images - these are the inliers. The homography that produces the most inliers is chosen as the best estimate, while the outliers of that homography are classified as false matches and removed. This technique is well explained in the context of panoramic stitching in [1].

Figure 2. Example image with SIFT keypoints identified and marked.

2.3. Homography Decomposition

Since the homography describes a perspective change between two views, it contains information on how the camera's pose was changed. To exploit this fact, we seek to decompose the identified homography matrix H_{ji} into a camera pose change R_{ji}, t_{ji}.

An analytical method proposed in [3] is used in this project. In general, this method produces up to four unique solutions, owing to the fact that a plane in perspective view can be tilted in two directions along each of its two axes while still producing the same top-down projection.

We can apply our triangulation method to all four of these unique solutions and generate depth estimates for the keypoints. This allows us to exclude solutions that result in negative depths, which leaves two of the four solutions, of which we can so far only guess one of them to use. This introduces considerable unreliability into the sequen-

tial odometry process, especially since errors will propagate. Figure 3 shows an example of the homography decomposition, where we see the four possible solutions for poses that realize the given homography transform.

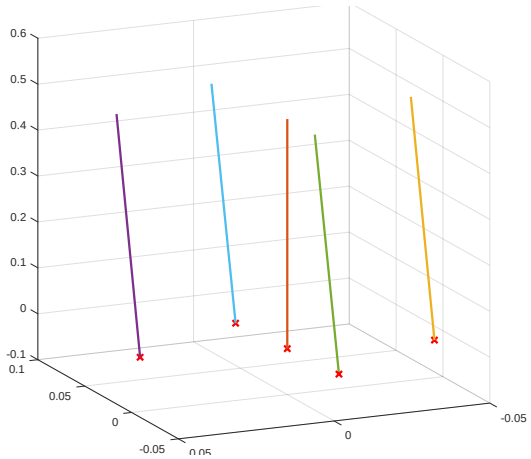

Figure 3. Example homography decomposition result. The central cross and line marks image i's position and orientation while the four surrounding camera poses are the four possible solutions from the homography decomposition.

2.4. Triangulation

Having identified two subsequent camera poses R_i, t_i and R_j, t_j, we can now gain some sparse 3D scene information from the identified inlier keypoint pairs. For each inlier pair of matched image points x'_i and x'_j we look to solve for the 3D point X by reforming (1):

$$X = u_i R_i^{-1} K^{-1} x'_i + t_i \tag{4}$$

Inserting the points and poses of both images i and j into (4) and setting them equal yields:

$$\begin{aligned} & u_j R_j^{-1} K^{-1} x'_j + t_j = u_i R_i^{-1} K^{-1} x'_i + t_i \\ \iff & t_j - t_i = u_i R_i^{-1} K^{-1} x'_i - u_j R_j^{-1} K^{-1} x'_j \\ \iff & b = Au \\ & \text{with } b = t_j - t_i \ , \ u = \begin{bmatrix} u_i \\ u_j \end{bmatrix} \\ & \text{and } A = \begin{bmatrix} R_i^{-1} K^{-1} x'_i & -R_j^{-1} K^{-1} x_j \end{bmatrix} \end{aligned} \tag{5}$$

This is an overdetermined linear equation system, since two lines in general do not have an intersection in 3D space. However, an estimated solution is possible with a least squares approximation:

$$u_{LS} = (A^T A)^{-1} A^T b \tag{6}$$

This approximate triangulation result can now be used to identify the 3D point corresponding to each matched keypoint pair. Figure 4 illustrates how our solution deals with the line intersection problem.

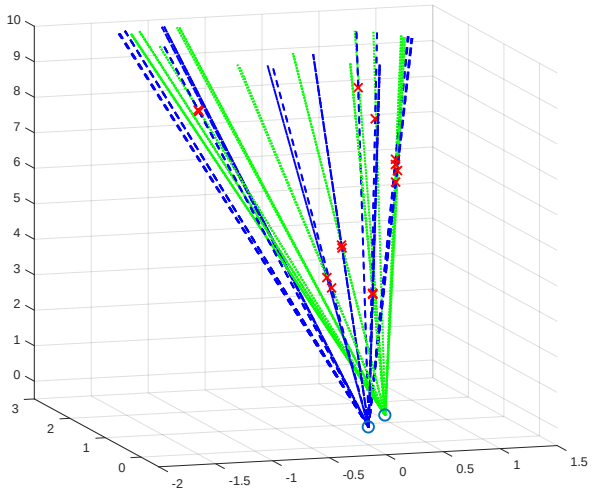

Figure 4. Triangulation example for the matched keypoint pairs of two images.

3. Implementation

In this short and exploratory study we prioritize convenience in our test setup.

3.1. Camera and Calibration

The camera used in the project is an inexpensive Logitech C270 webcam as seen in 5. We set constant focus and resolution and capture a set of pictures of a 9 by 7 checkerboard to calibrate the camera. We use the MatLab Camera Calibration App to speed up the process as shown in figure 6. We include images from different depths and at different skew angles, and we also make sure to use the camera's entire field of view over the calibration set. The result is a complete estimate of the camera's intrinsic parameters, including lens distortion.

A feature of the camera to keep in mind is its small aperture, so we make sure to use our system in daylight condition in order to allow shorter exposure times and suffer less motion blur.

3.2. Software Libraries

We build our software system in MatLab, and use the included image processing functions for the homography estimation with RANSAC outlier rejection.

We also include the image processing library VLfeat, which has good implementations of SIFT keypoint detection and matching.

Finally, we include the .mex build of OpenCV which can be found under the name mexopencv. This library includes

Figure 5. The webcam we are using for our tests.

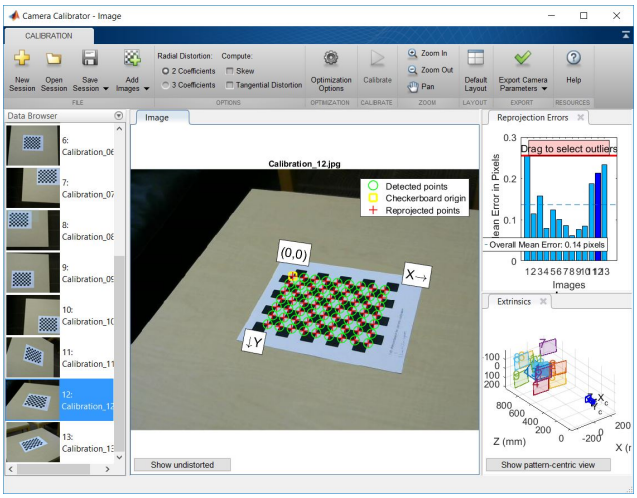

Figure 6. The calibration app for estimating the camera parameters from a set of images of checkerboard patterns.

a good implementation of the homography to pose decomposition.

4. Experimental Results

We test our system on a set of ten images taken from Stanford University's Huang Engineering Center in bright daylight conditions, producing the images shown in figure 7. The sequential odometry for all ten images produces the camera poses seen in figure 8. We can see that the erratic nature of the multiple solutions from the homography decomposition disrupts what should be a straight line of camera movement. This also ruins the results of the combined triangulation from the entire image set.

We show an example of a two-view triangulation in the figures 9 and 10, displaying two different 3D perspectives of the resulting point cloud. We can see that point cloud does reflect the location of the detected keypoint matches.

The mediocre performance of our current system means that there is little sense in doing more detailed benchmarks. We see that the sequential approach, lack of filtering and uncertainty in the homography decomposition are crippling problems that need to be addressed before any deeper evaluation makes sense.

Figure 7. Image test set taken while walking in a straight line.

5. Improvements and Future Work

Having pointed out the main flaws of our approach in the previous section, we now seek solutions that could address

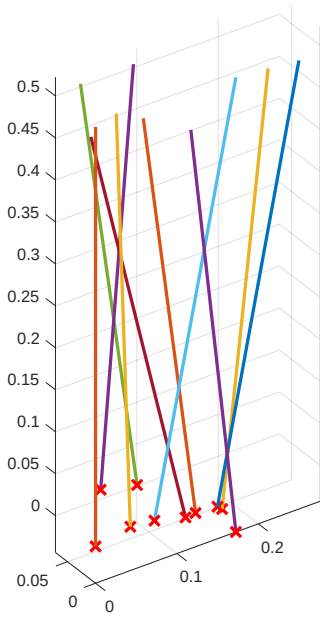

Figure 8. Odometry result for the ten test images.

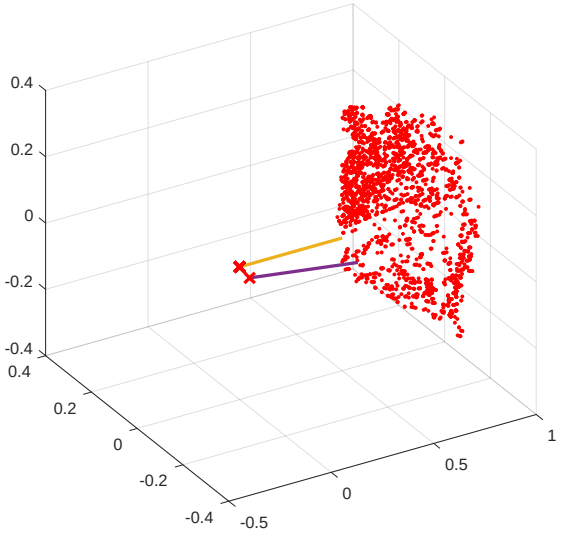

Figure 9. Reconstruction of the 3D positions of keypoints from the first two images of our test set.

them. A promising way to make the algorithm more stable from image to image would be to consider more than two images at once. Matching keypoints from each image in the group at the same time would allow us to look for a combined solution for all relative camera poses in question.

One way of reaching the solution could be through using our homography decomposition to get all possible relative poses. We could then try all possible combinations

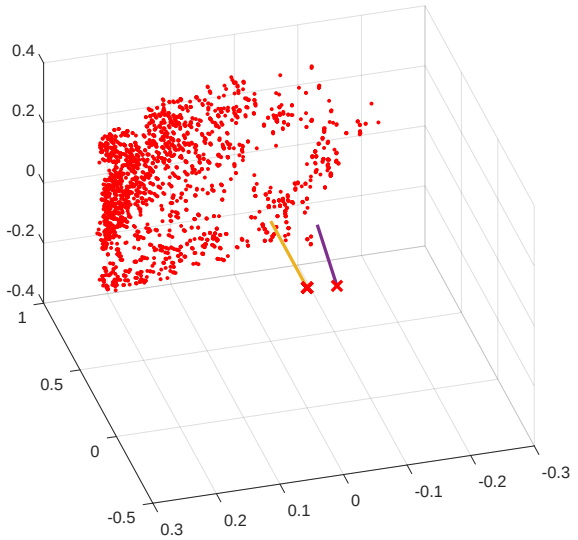

Figure 10. A different view of the reconstruction of the 3D positions of keypoints from the first two images of our test set.

and choose the one that places all 3D keypoints at positive depths relative to the camera poses (i.e. in front of the camera).

Other than that, a state-of-the-art solution to the multiple view structure from motion problem is bundle adjustment. This method is based on finding a solution to the optimization problem

$$\min_{R,t} \sum_{i} \left(u \cdot x_i' - K R_i (X - t_i)\right)^2 \tag{7}$$

for all matched sets of keypoints $x_1', ..., x_n'$. A solution to this problem is difficult but quite well explored - the usual solution involves using the Levenberg-Marquardt Algorithm which solves nonlinear damped LS optimization problems. Our gained insights into homography decomposition may still prove useful to use as initial conditions for the bundle adjustment solutions.

Next, with the algorithm stabilized, we would want to find a way to get more dense 3D reconstruction than we can get from just the keypoints.

6. Conclusion

Although the time constraints and complexity of the problem limited the results achieved in the scope of just this problem, the insights gained do seem worthwhile. The results suggest that if the homography decomposition were stabilized with one of the suggested methods, the odometry at least could reach a good quality. Generating dense and accurate 3D scene maps would likely take a significant amount of additional work with this system.

All in all however, the results can be seen in a positive light. We have somewhat explored the possibilities of this simple camera setup and validated our low-cost approach to this high-end problem.

Acknowledgements

Finally, thanks go to Professor Gordon Wetzstein for being the great teacher who brought this author into the world of image processing. In this project, it is thanks to Gordon's encouragement that I have now also taken my first steps into 3D computer vision and structure from motion, and for that I am grateful.

References

[1] M. Brown and D. G. Lowe. Recognising panoramas, 2003.

[2] D. G. Lowe. Object recognition from local scale-invariant features, 1999.

[3] E. Malis and M. Vargas. Deeper understanding of the homography decomposition for vision-based control. [research report] rr-6303.

[4] L. Quan and T. Kanade. *Image-Based Modelling*. Springer US, 2010.

[5] A. Z. R. Hartley. *Multiple view geometry in computer vision*. Cambridge University Press, 2004.

[6] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag London, 2011.