Structure from Motion using Uncalibrated Cameras

Avinash Bhaskaran
Georgia Institute of Technology

Anusha Sridhar Rao
Georgia Institute of Technology

Abstract

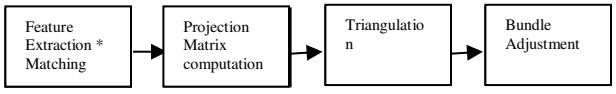
The problem addressed in this paper is the reconstruction of an object in the form of a 3D model from images taken with uncalibrated cameras. Along with the 3D reconstruction of the object, the project also aims at estimating camera motion. The structure from motion pipeline has 5 keys steps, namely key point detection, feature matching, motion estimation triangulation and bundle adjustment. In this project we also survey different implementation schemes for each module of the pipeline and weigh the pros and cons of each method.

1. Introduction

The 3D model of an object can be reconstructed using correspondences in multiple images from a camera and then triangulation the 3D points. For triangulation of the 3D points, the relative positions of the camera and their intrinsic parameters have to be known. A projective approach is used for 3D reconstruction which can also be used to jointly optimize over camera parameters too.

1.1. Structure from Motion Pipeline

The structure from motion pipeline is shown in figure 1. The first stage involves finding key points in consecutive images. Upon extracting features from the images, features are matched between images. Using the correspondences, fundamental matrix is computed which is in turn used to compute the essential matrix. Using the essential matrix, projection matrices are computed. This projection matrix is used to obtained 3D points. These 3D points are triangulated to refine the 3D point location. Upon triangulation, the camera poses are further refined using bundle adjustment.



1.2. Feature Extraction & Matching

Feature extraction refers to the process of selection points in an image that would yield good features and compute descriptors for the same. A descriptor is a vector of numbers that describe a surrounding environment around a feature point in an image. The features we experimented with in our project are SURF, SIFT and BRISK. The quality of the features extracted affect the projection matrix that we later compute which in turn affects the 3D reconstruction.

Upon extracting the features, the features between successive images are matched. Feature matching refers to the process of finding a corresponding feature between image pairs using its descriptors. FLANN based matchers and brute force matchers were experimented.

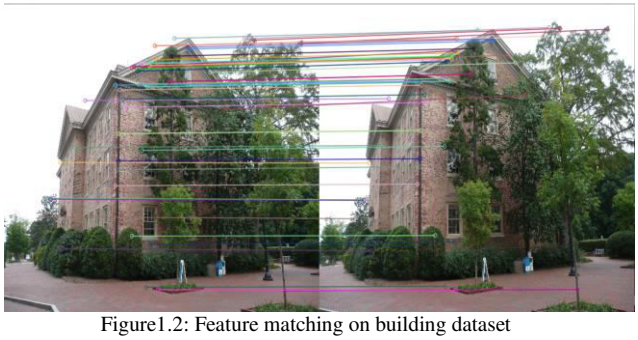


Figure1.2: Feature matching on building dataset

1.3. Compute Essential Matrix and Projection Matrix for First Camera Pair

The next step after computing the key points and matches would be to estimate the fundamental matrix. The following methods were implemented to choose the best fundamental matrix: 8-point algorithm, LMEDS and RANSAC.

The essential matrix and the projection matrix are computed as follows:

1. Estimate fundamental matrix using the matches
2. Perform single value decomposition on the

fundamental matrix

3. Reduce the rank of the fundamental matrix to 2.
4. Compute essential matrix

$$E = K'^{T}FK$$

Where,
E= essential matrix
K= camera matrix

5. Compute Projection Matrix

Assuming the first camera matrix to be P=[I|0]

In order to compute the second camera matrix P', E can be considered as E=SR, where S and R are the skew symmetric and rotation matrices respectively.

$$\text{SVD(E)} = \text{Udiag}(1,1,0)V^T$$

$$\text{R} = \text{UW}V^T \text{ or } \text{U}W^{-1}V^T$$

Translation T is given by,

$$\text{T} = \pm U_3$$

There are 4 possible solutions for the projection matrix (for different combinations of rotations and translation). Out of these 4 possible combinations, only one of them is physically plausible. The correct projection matrix is the one that produces reconstructed points with a positive Z value (points are in front of the camera).

1.4. Structure from Motion using 2 Views

There are 2 approaches to reconstruct the 3D structure from multiple views. The first approach involves iteratively performing the 2 view structure from motion algorithm (shown in fig 1.1) over multiple views. The second approach involves using perspective projection. In this section we will discuss the first approach in detail.

At the stage we have the extrinsics and intrinsic parameters for the first 2 views (since we are using a single camera, the intrinsic will remain the same for all views). The next step would be to get the 3D points estimate for the 2 views using the projection matrix. Upon getting the initial estimate of the 3D points, these points have to be triangulated. In this project we have implemented a linear triangulation algorithm. Triangulation is iteratively performed over every image pair in a similar fashion.

However, one of the drawbacks of this approach is that the reconstruction is only up to scale. The motion we obtained between 2 views is going to have an arbitrary unit of measurement, that is, it is not in centimeters or inches but simply a given unit of scale. Our reconstructed cameras we will be one unit of scale distance apart. This has big implications when we extend the 2 view algorithm to multiple view as stated earlier in this section, as each pair of cameras will have their own units of scale, rather than a common one.

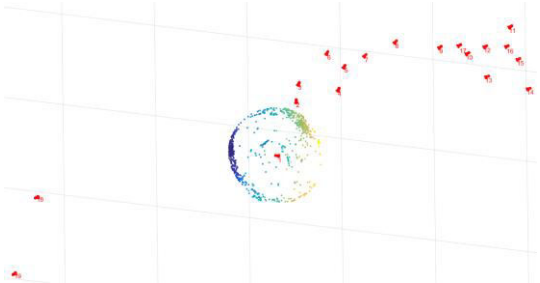


Figure 1.3: Camera Position and 3D reconstruction on the Dino Dataset. Camera pose estimation is incorrect due to scale limitation of the algorithm

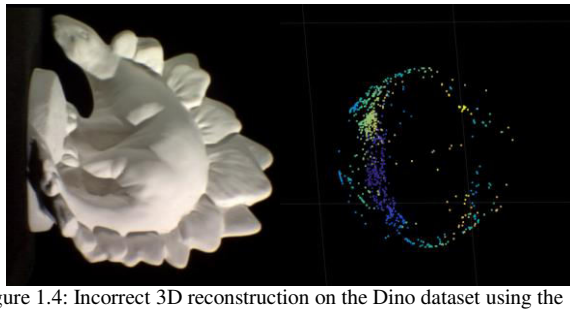


Figure 1.4: Incorrect 3D reconstruction on the Dino dataset using the 2 view approach

2. Perspective-N-Point with Uncalibrated Cameras

Using Perspective-N-Point algorithm for multi view reconstruction takes care of the scale issue that existed in the algorithm described in the previous section. Figure shows the pipeline for the same.

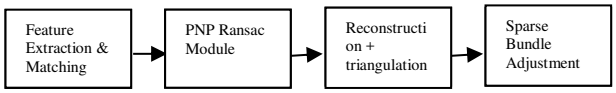


Figure 2.1: Structure from motion pipeline using PNP

We have the projection matrices of the first 2 views from section. As mentioned earlier, we perform triangulation to get the 3D points. Now that we have a baseline structure, we compute the projection matrix of the next view using the 3D points that correspond to the 2D points of the new frame which is in turn used as an input to the PNP module. The PNP module returns the projection matrix of the new frame. We perform reconstruction followed by triangulation again and the process is repeated iteratively to get the projection matrix of successive views.

The method involves bookkeeping i.e. we need to keep a track of the 3D points reconstructed using the previous 2 frames that correspond to the 2D points in the new frame. For every point in the 3D point cloud, we maintain a vector denoting the 2D points in came from. We then use feature matching to get a matching pair.

Also, we observed that in some cases using PNP didn't give us a plausible projection matrix. In such cases, we used PNP Ransac which seemed to be more robust. In cases when PNP Ransac also failed, we switch to the 2 view structure from motion algorithm to compute the projection matrix of the frame for which PNP fails.

2.1. Bundle Adjustment

One of the key steps in the structure from motion pipeline is the bundle adjustment module. Bundle adjustment is used to optimize and refine the 3D structure of the reconstructed object and the estimated camera poses. Since we are working with uncalibrated cameras, the bundle adjustment algorithm optimizes over the camera intrinsics as well.

Assuming there are n 3D points seen in m views, and Xij is the projection of the ith point on image j. Let vij denote the binary variable that is equal to 1 if point i is visible in image j and 0 otherwise. Assume also that each camera j is parametrized by a vector ai and each 3D point I by a vector bi. Bundle adjustment minimizes the total re-projection error with respect to all 3D points and camera parameters,

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} d(Q(a_j, b_i), x_{ij})\text{^}2$$

Sparse bundle adjustment was implemented. There are 2 ways in which the optimization can be performed: Sequential method and Batch optimization.

Sequential method involves performing the optimization with the addition of every new frame. However, sequential methods have drawbacks such as large number of corresponding points must be defined in every view and there has to be a substantial overlap.

Batch optimization works by refining the camera pose and 3D points using all image measurements simultaneously. One of the key advantages in batch optimization is that the reconstruction errors can be distributed meaningfully across all measurements, hence the errors associated with sequence closure can be avoided.

2.2. Dataset

The algorithm was tested on multiple datasets. All the datasets are images taken from a single camera moving around the object of interest. The datasets we experimented with are Middle bury temple and dino dataset, Fountain K6 dataset. Since the algorithm works for uncalibrated cameras as well we try it on a video that we recorded and on a natural dataset where the camera parameters were unknown.

Figure 2.2: Dataset with unknown camera parameters

3. Results

The 2 approaches mentioned in section 1.4 & 2 were implemented to generate the reconstructed model and estimate the camera motion. The reprojection error was calculated in both cases, with and without triangulation and bundle adjustment to gauge the performance of the different algorithms.

Method	Mean Reprojection Error
2 View Reconstruction	1.5
2 View Reconstruction + Bundle Adjustment	NA
Iterative Perspective N Point Algorithm + Linear Triangulation	110.9
Iterative Perspective N Point Algorithm RANSAC + Linear Triangulation	42.7
Efficient N Point RANSAC + Linear Triangulation	124.5
Iterative Perspective N Point RANSAC + Linear Triangulation + Bundle Adjustment (Sequential)	20
Iterative Perspective N Point RANSAC + Linear Triangulation + Bundle Adjustment (Batch Optimization)	**10**

Table: Mean Reprojection error

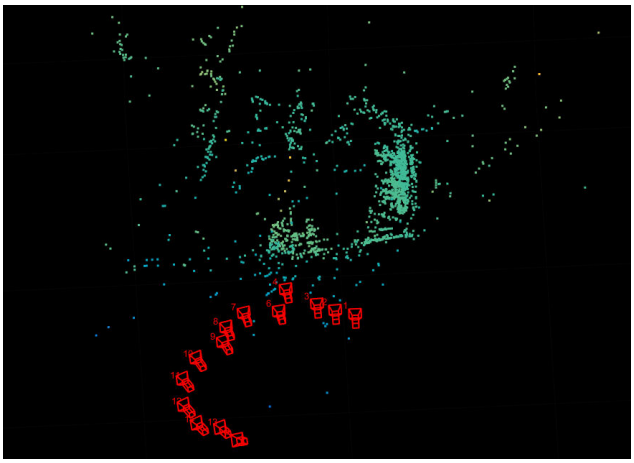


Figure 3.1: Sparse Reconstruction with estimated Camera Motion

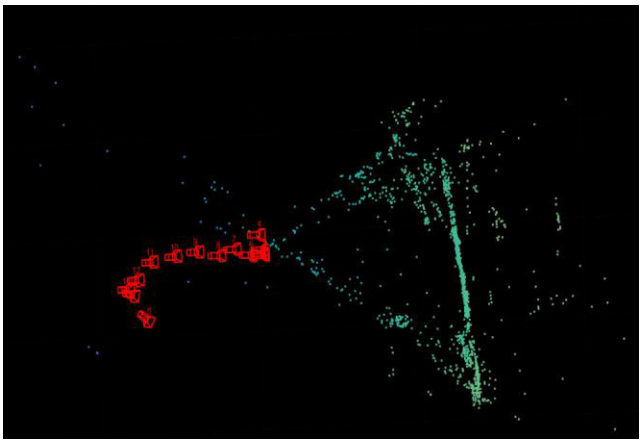


Figure3.2: Sparse Reconstruction with estimated Camera Motion

3.1. Discussion

From the table, it can be concluded that the Perspective N Point RANSAC algorithm with triangulation and bundle adjustments performs the best since the reprojection error is the least and the 3D reconstruction of the scene is closest to the ground truth as compared to the other algorithms. Also, the if pose estimation for a particular view is incorrect, the error doesn't propagate as much as in the case of the other approaches. It was observed that the performance varied with the dataset. Synthetic dataset like the dino and temple dataset perform poorly as compared to the natural images.

The proposed algorithm has some challenges that are yet to be addressed. It is very computationally intensive. In addition to that the matcher used for keypoint matching is very slow. Also, the proposed approach performs relatively well for uncalibrated cameras only when the resolution of the images is very high.

3.2. Future Work

In the algorithm implemented, linear triangulation was used which performed reasonably well. It would be fair to assume that with non-linear triangulation the performance should get better. It was observed that a few conflicting matches, sets the pose estimation far off from the ground truth. These conflicting matches have to be eliminated in the future implementation to get a more accurate motion estimation.

References

[1] Z. Zhang, ªA Flexible New Technique for Camera Calibration,º Technical Report MSR-TR-98-71, Microsoft Research, Dec. 1998. Available together with the software at http://research.microsoft.com/~zhang/Calib/

[2] Klaus Häming and Gabriele Peters. The structure-from-motion reconstruction pipeline - a survey with focus on short image sequences. Kybernetika, 46(5):926–937, 2010

[3] R. I. Hartley and A. Zisserman: Multiple View Geometry in Computer Vision. Second edition. Cambridge University Press 2004