

Linear Global Translation Estimation with Feature Tracks: Supplementary Material

Zhaopeng Cui¹

zhpcui@gmail.com

Nianjuan Jiang²

nianjuan.jiang@adsc.com.sg

Chengzhou Tang¹

chengzhout@gmail.com

Ping Tan¹

pingtan@sfu.ca

¹GrUVi Lab

Simon Fraser University

Burnaby, Canada

²Advanced Digital Sciences Center of

Illinois, Singapore

Abstract

This supplementary material contains more details of our feature tracks selection and further experimental results of our method as follows: Section 1 describes the pseudo-code for our feature tracks selection; Section 2 demonstrates experimental evaluation of our method on synthetic data; Section 3 presents the visualization of our results on the Internet data published in [B] and the challenging data *Quad* published by [B].

1 Pseudo-code for feature tracks selection

Algorithm 1 Feature tracks selection.

```

1: Initialize: Sort all feature tracks by their lengths in descending order, and let  $\mathbb{T}$  represent
   the sorted set of tracks;
2: for  $i = 0$  to  $K$  do
3:   Set the current set of cameras  $\mathbb{C}$  to  $\emptyset$ ;
4:   Choose the first track  $t_1$  in  $\mathbb{T}$ , update  $\mathbb{C}$  as  $\mathbb{C} = \mathbb{C} \cup \mathbb{C}_{t_1}$ , and remove  $t_1$  from  $\mathbb{T}$ ;
5:   while  $\mathbb{C} \neq \mathbb{C}^*$  do
6:     Try to find a track  $t$  in  $\mathbb{T}$  which satisfies  $\mathbb{C}_t \cap \mathbb{C} \neq \emptyset$  and  $\mathbb{C}_t \not\subseteq \mathbb{C}$ ;
7:     if It fails to find such  $t$  then
8:       break;
9:     end if
10:    Update  $\mathbb{C}$  as  $\mathbb{C} = \mathbb{C} \cup \mathbb{C}_t$ , and remove  $t$  from  $\mathbb{T}$ ;
11:   end while
12:   if  $\mathbb{C} \neq \mathbb{C}^*$  then
13:     break;
14:   end if
15: end for

```

Let \mathbb{C}^* be the set of all connected cameras, and \mathbb{C}_t be the set of cameras on a feature track t , then our method of feature tracks selection can be summarized as Algorithm 1. K is set as 30 in our experiments.

2 Evaluation on synthetic data

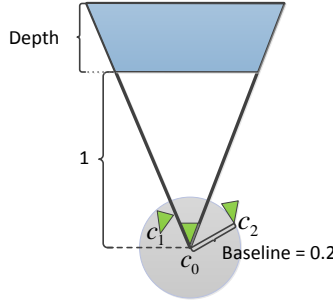


Figure 1: Configuration for synthetic data generation. We randomly place cameras c_0, c_1 , and c_2 and 500 scene points as illustrated here. We then test each method 100 times on the randomly created images, and evaluate their accuracy with respect to the ground truth.

We evaluate the performance of our algorithm on synthetic data with known ground truth. The synthetic data consists of three cameras and 500 scene points generated according to the following setting. As illustrated in Figure 1, camera c_0 is placed at the world origin and camera c_2 is at a random location away from c_0 by 0.2 unit. The position of c_1 is generated in different ways to evaluate different aspects of the SfM system. Each camera has a field of view of 45° and image resolution of 352×288 pixels. The scene points are generated randomly within the viewing volume of the first camera, spanning a depth range of about 0.5 unit. The scene distance from the first camera is about 1 unit.

Two metrics are used to evaluate the accuracy of computed camera poses. The error of relative translation directions t_{err} is the mean angular difference (in degrees) between the estimated and the true baseline directions. The error of camera positions c_{err} is the mean Euclidean distance between the estimated and the true camera positions. We report the average results of 100 random trials for each method.

Comparison with [10]. We compare our method with the method in [10] to demonstrate the robustness of our algorithm with collinear camera motion. We randomly sample c_1 to vary the angle $\angle c_1 c_0 c_2$ from 0.1 to 5 degrees while ensuring c_1 keep equal distances from c_0 and c_2 . Image coordinates of the projected 3D points are perturbed by zero mean Gaussian noise with standard deviation $\sigma = 0.4$ pixels. A comparison of the results from these two methods are shown in Figure 2. According to both error metrics t_{err} and c_{err} , the performance of our method is quite stable under collinear camera motion.

Comparison with [4]. We compare our method with the method in [4] on data with weak association. In this experiment, we fix the c_2 at the point $(-0.2, 0, 0)$ and randomly sample c_1 with $\angle c_1 c_0 c_2$ fixed at 45 degrees. Moreover, we fix the number of feature correspondences between c_1 and c_2 at 10, and adjust the standard deviation σ of the Gaussian noise on feature positions from 0.1 to 1 pixel. This setup simulates situations of weak data association (i.e., the essential matrix between c_1, c_2 is poor). The reconstruction accuracies

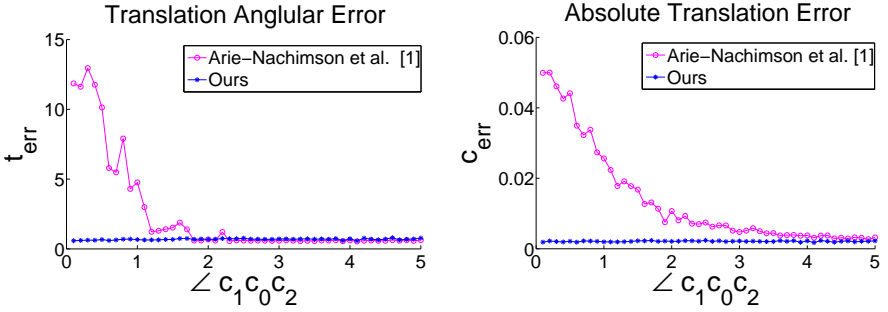


Figure 2: Comparison with the method in [1] under collinear camera motion. t_{err} and c_{err} are the reconstruction errors in relative translation directions and camera positions. The horizontal axes of both charts are the angle $\angle c_1 c_0 c_2$ formed by the cameras c_0, c_1 , and c_2 . It is clear our method is robust to collinear motion, while the method in [1] is not.

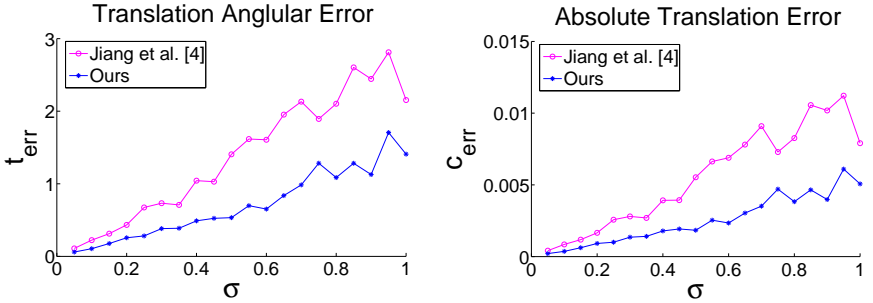


Figure 3: Comparison with the method in [4] on data with weak association. t_{err} and c_{err} are the reconstruction errors in relative translation directions and camera positions. The horizontal axis σ is the standard deviation of the Gaussian noise on image feature locations. Our method is more robust than [4] to heavily contaminated essential matrices.

of both methods are reported in Figure 3. It is clear that as σ increases, our method produces more accurate results than [4] according to both metrics. Note that our method can work even when there are not enough correspondences (e.g. less than 5) between c_1 and c_2 to compute the essential matrix, while the method in [4] will fail in this case.

3 Visualization of results on Internet data

Figure 4 visualizes reconstruction results of our method on the Internet data published in [6]. For quantitative evaluation, please refer to Tables 3 and 4 in the paper.

We also try our method on the challenging *Quad* data published by [6], which consists of 6514 images and has poor epipolar geometries. Our method generates a distorted result for this example without geotags. However it produces a reasonable result as shown in Figure 5, when camera orientations from Bundler [6] are used (instead of those computed by rotation averaging). The rotation averaging method in [6] does not guarantee the global optimal result because the rotation manifold constraint makes the problem complicated. This experiment seems to suggest that rotation estimation is the current bottleneck for global SfM.

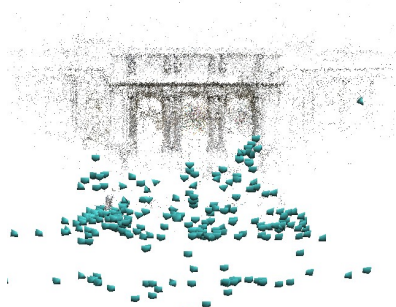
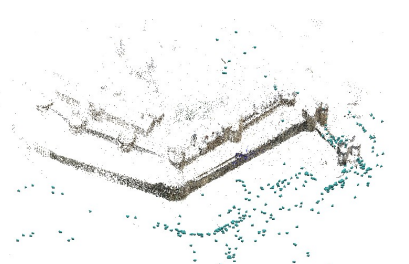
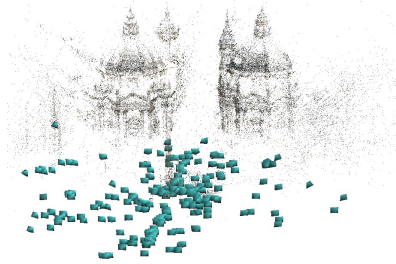
(a) *Almo*(b) *NYC Library*(c) *Ellis Island*(d) *Notre Dame*(e) *Montreal N.D.*(f) *Vienna Cathedral*(e) *Tower of London*(f) *Piazza del Popolo*

Figure 4: Reconstruction results on the data released by [5].



Figure 5: The reconstruction result for *Quad*. The rotations from Bundler [5] are used for this dataset.

References

- [1] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Proc. 3DPVT*, 2012.
- [2] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proc. ICCV*, pages 521–528, 2013.
- [3] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, pages 3001–3008, 2011.
- [4] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *Proc. ICCV*, 2013.
- [5] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. on Graph.*, 25:835–846, 2006.
- [6] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proc. ECCV (3)*, pages 61–75, 2014.