GPU-Based Homotopy Continuation for Minimal Problems in Computer Vision

Chiang-Heng Chien
School of Engineering
Brown University
chiang-heng_chien@brown.edu

Hongyi Fan
School of Engineering
Brown University
hongyi_fan@brown.edu

Ahmad Abdelfattah
Innovative Computing Laboratory
University of Tennessee
ahmad@icl.utk.edu

Elias Tsigaridas
INRIA
elias.tsigaridas@inria.fr

Stanimire Tomov
Innovative Computing Laboratory
University of Tennessee
tomov@icl.utk.edu

Benjamin Kimia
School of Engineering
Brown University
benjamin_kimia@brown.edu

In this supplementary material, we aim to show the effectiveness of the proposed 4-view triangulation problem using a synthetic multiview dataset from [2] and a real dataset [4]. Evaluating from a synthetic multiview dataset, we randomly select 200 points from random 4 pairs of views. To simulate the noise on the observed points and the calibration, the poses are perturbed with $\mathcal{N}(0, 2.0)$ and the observed points are perturbed with $\mathcal{N}(0, 1.0)$. We compare our 4-view triangulate result with 3-view triangulation [3], Figure 1, using the proposed GPU-HC. It is clear to see that the error between the optimized 2D points and the ground truths of 4-view triangulation is smaller than 3-view triangulation. Quantitatively, the mean error of 4-view triangulation is 2.9712 pixels, while in 3-view triangulation, the error is 4.2074 pixels. Therefore, adding one more view to do triangulation would drop the projection error significantly.

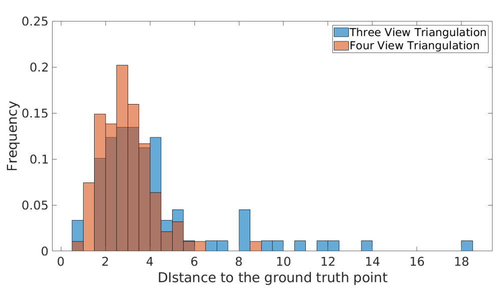
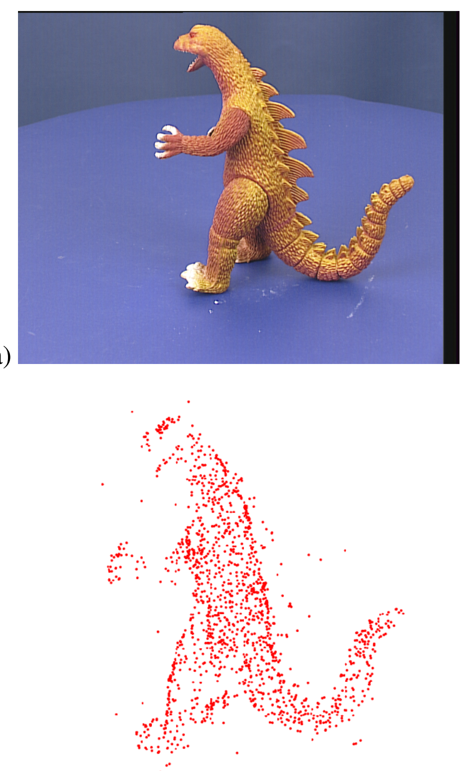


Figure 1: The 2D error histogram of 3-view and 4-view triangulation with synthetic Multiview curve dataset [1].

Employing a real dataset, we use the dinosaur sequence from [4] which contains 36 images with 4,983 corresponding points. To deploy our 4-view triangulation method, only



(a)

(b)

Figure 2: Triangulation result using GPU-HC on 4-view triangulation method. Note that this more accurate triangulation has not been tried because it is not solvable by previous techniques.

1,516 points that are co-visible by more than 3 views are used. The triangulation result is shown in Figure 2. Note that after GPU-HC computes the optimal positions of the image points, the projection lines from all four images always intersect in the 3D space. In such a case, we are free to select any two views to find the position of the 3D points given the camera extrinsic matrix of these two views. The processing time of this whole sequence using the proposed GPU-HC is around 10.6 seconds.

References

[1] R. Fabbri, T. Duff, H. Fan, M. Regan, D. d. C. de Pinho, E. Tsigaridas, C. Wampler, J. Hauenstein, B. Kimia, A. Leykin, et al. Trifocal relative pose from lines at points and its efficient solution. *arXiv preprint arXiv:1903.09755*, 2019. 1

[2] R. Fabbri, T. Duff, H. Fan, M. H. Regan, D. d. C. d. Pinho, E. Tsigaridas, C. W. Wampler, J. D. Hauenstein, P. J. Giblin, B. Kimia, et al. TRPLP-trifocal relative pose from lines at points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12073–12083, 2020. 1

[3] Z. Kukelova, T. Pajdla, and M. Bujnak. Fast and stable algebraic solution to l2 three-view triangulation. In *2013 International Conference on 3D Vision-3DV 2013*, pages 326–333. IEEE, 2013. 1

[4] A. Zisserman. VGG Multi-view Data. https://www.robots.ox.ac.uk/~vgg/data/mview/. 1