

Progressive Batching for Efficient Non-linear Least Squares - Supplementary Material

Huu Le¹, Christopher Zach¹, Edward Rosten², and Oliver J. Woodford²

¹ Chalmers University, Sweden

² Snap, Inc., London & Santa Monica

A More Results on Dense Image Alignment

In this section, we provide additional results on the dense image alignment experiment. Fig. A plots the evolution of 4 more image pairs in the ETH-3D dataset. Note that L-BFGS performs poorly, hence we omit their results. We also show an example of qualitative result in Fig. B.

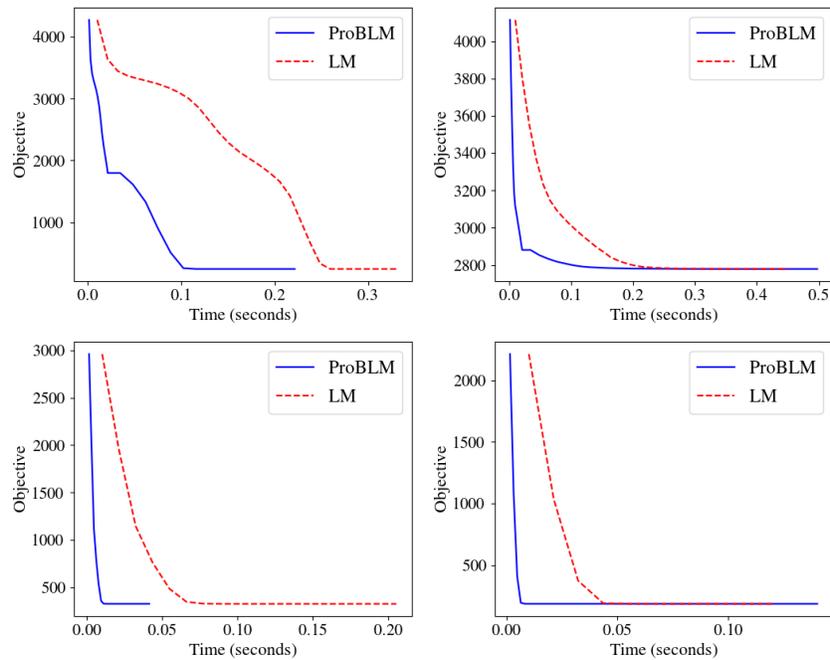


Fig. A: Plots of objective vs. time for our method in comparison with LM on dense image alignment.

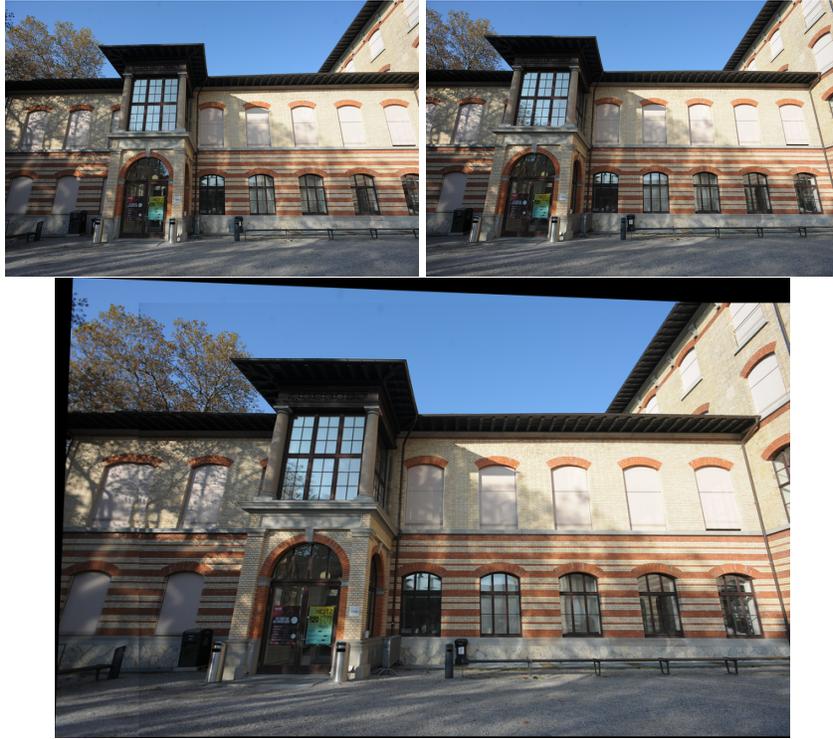


Fig. B: Qualitative results for dense image alignment experiment. Top: Source and target images. Bottom: Alignment result.

B Weak-Perspective Affine Bundle Adjustment

We also test the performance of our algorithm on small-scale affine bundle adjustment problems. The objective function can be written as

$$\min_{\{\mathbf{R}_i, \mathbf{t}_i\}, \{\mathbf{X}_j\}} \sum_{i=1}^C \sum_{j=1}^P \|\pi(\mathbf{R}_i \mathbf{X}_j + \mathbf{t}_i) - \mathbf{m}_{ij}\|^2, \quad (1)$$

where C is the number of cameras and P is the number of 3D points. The parameters $(\mathbf{R}_i, \mathbf{t}_i)$ are respectively the rotation and translation that map a 3D point $\mathbf{X} \in \mathbb{R}^3$ from the world coordinate to the coordinate of i -th camera, and \mathbf{m}_{ij} is the 2D projection of the point \mathbf{X}_j onto the image of camera i . Here we use the weak-perspective affine model, i.e.,

$$\pi([x_1, x_2, x_3]^T) = \begin{bmatrix} x_1 & x_2 \\ \frac{x_1}{\tilde{x}_3} & \frac{x_2}{\tilde{x}_3} \end{bmatrix}^T, \quad (2)$$

where \tilde{x}_3 is a fixed average depth. In our experiments, instead of using a single average depth for all 3D points, we associate each point \mathbf{X}_j with an average

depth \tilde{x}_j , where \tilde{x}_j is assigned with the initial depth of the point \mathbf{X}_j and fixed throughout the optimization process. Observe that, with this affine model, when $\{\mathbf{R}_i\}$ and $\{\mathbf{t}_i\}$ are fixed, the points can be solved in closed form. Therefore, we employ variable projection [1] to first solve for the camera parameters, then update the points using standard linear least squares. In this experiment, we focus on settings where every 3D point is visible in all cameras, which is a common setting in e.g., several monocular SLAM applications. The ‘‘South Building’’ dataset from the COLMAP package³ is used, where we extract 3 adjacent frames and all 3D points that are visible in all extracted frames, resulting in a problem instance containing 802 points in 3D and 2406 measurements. Fig. C (left) plots the objective for conventional LM and our method. Observe that ProBLM also offers favorable results. The same experiment is repeated for 5 views with 212 3D points, and the results is plotted in Fig. C (right). It can also be seen that ProBLM converges faster (although LM has comparable performance in this case because the number of 3D points is smaller than the case of three views shown in the left figure).

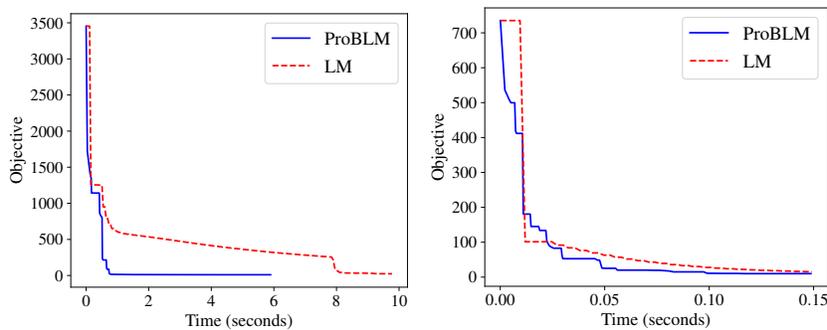


Fig. C: Plots of objective of LM and ProBLM for bundle adjustment. Left: A problem instance with with 3 views. Right: A problem instance with 5 views.

C Algorithm for Relaxed Condition

Algorithm A describes the relaxed version of our method (using Eq. (18)). In Fig. D, we compare the performance of the standard condition (Eq. (13)) and relaxed condition (Eq. (18)). As can be seen, the relaxed condition generally offers better performance.

³ <https://demuc.de/colmap/datasets/index.html>

Algorithm A Relaxed ProBLM

Require: Initial solution $\boldsymbol{\theta}^{(0)}$, initial batch size K_0 , maximum iterations `max_iter`**Require:** Confidence level $\delta \in (0, 1)$, margin parameter $\alpha \in [0, 1)$, η

- 1: Randomly reshuffle the residuals $\{f_i\}$
- 2: Initialization: $t \leftarrow 0$, $K \leftarrow K_0$, $t_0 \leftarrow 0$.
- 3: **while** $t < \text{max_iter}$ and a convergence criterion is not met **do**
- 4: $\mathcal{S}^{(t)} \leftarrow \{1, \dots, K\}$
- 5: Compute $\mathbf{g}_{\mathcal{S}^{(t)}}$ and $\mathbf{H}_{\mathcal{S}^{(t)}}$

$$\mathbf{g}_{\mathcal{S}^{(t)}} := \sum_{i \in \mathcal{S}^{(t)}} \mathbf{J}_i^{(t)} \mathbf{r}_i^{(t)} \quad \mathbf{H}_{\mathcal{S}^{(t)}} := \sum_{i \in \mathcal{S}^{(t)}} (\mathbf{J}_i^{(t)})^T \mathbf{J}_i^{(t)}. \quad (3)$$

and solve

$$\Delta \boldsymbol{\theta}^{(t)} \leftarrow (\mathbf{H}_{\mathcal{S}^{(t)}} + \lambda \mathbf{I})^{-1} \mathbf{g}_{\mathcal{S}^{(t)}} \quad \boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \Delta \boldsymbol{\theta}^{(t)} \quad (4)$$

- 6: **if** $f_i(\boldsymbol{\theta}^{(t+1)}) - f_i(\boldsymbol{\theta}^{(t)}) \geq 0$ **then**
- 7: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}$, and $\lambda \leftarrow 10 \lambda$ ▷ Failure step
- 8: **else**
- 9: Determine current lower and upper bounds a and b , and set

$$U_K^{(t_0, t+1)} \leftarrow \sum_{i \in \mathcal{S}^{(t)}} \max \left\{ a, \left(f_i(\boldsymbol{\theta}^{(t+1)}) - f_i(\boldsymbol{\theta}^{(t_0)}) \right) \right\}. \quad (5)$$

- 10: $p \leftarrow$ Random number between 0 and 1.
 - 11: **if** U_K satisfies Eq. (18) **or** $p \leq \eta$ **then**
 - 12: $\lambda \leftarrow \lambda/10$ ▷ Success step
 - 13: **else**
 - 14: $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)}$ and increase K using Eq. (16). (with S_K replaced by U_K)
 - 15: $t_0 \leftarrow (t + 1)$
 - 16: **end if**
 - 17: **end if**
 - 18: $t \leftarrow t + 1$
 - 19: **end while**
 - 20: **return** $\boldsymbol{\theta}^{(t)}$
-

D RANSAC vs. Robustified ProBLM

In Fig. E, we plot the number of inliers obtained using RANSAC and robustified ProBLM of 5 randomly chosen image pairs from the ETH3D dataset. For every image pair, each method is run 10 times with a fixed time budget (ProBLM is initialized with random initializations), and the reported results are averaged over 10 runs. As can be observed from Fig. E, within the same run-time budget, ProBLM achieves competitive number of inliers without the need of RANSAC. This suggests that our method has the potential of directly fitting the model without requiring RANSAC as a pre-processing step, which is highly relevant for many real-time applications.

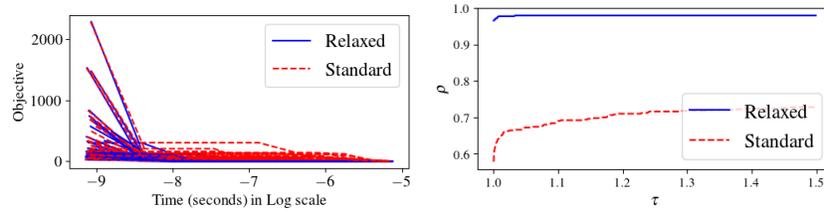


Fig. D: Comparison between the two conditions. Left: Plots of objective vs. run time for 20 runs. Right: Performance profile with a 10ms time budget, summarized over 500 runs.

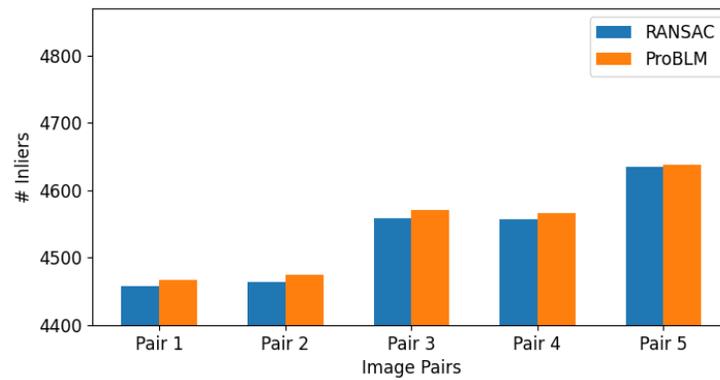


Fig. E: Number of inliers obtained after a fixed time budget of RANSAC and robustified ProBLM.

References

1. Hyeon Hong, J., Zach, C., Fitzgibbon, A.: Revisiting the variable projection method for separable nonlinear least squares problems. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 127–135