Supplementary Material for "An Analysis of Sketched IRLS for Accelerated Sparse Residual Regression," ECCV 2020

Daichi Iwata[1], Michael Waechter[1], Wen-Yan Lin[1], and Yasuyuki Matsushita[1]

Osaka University

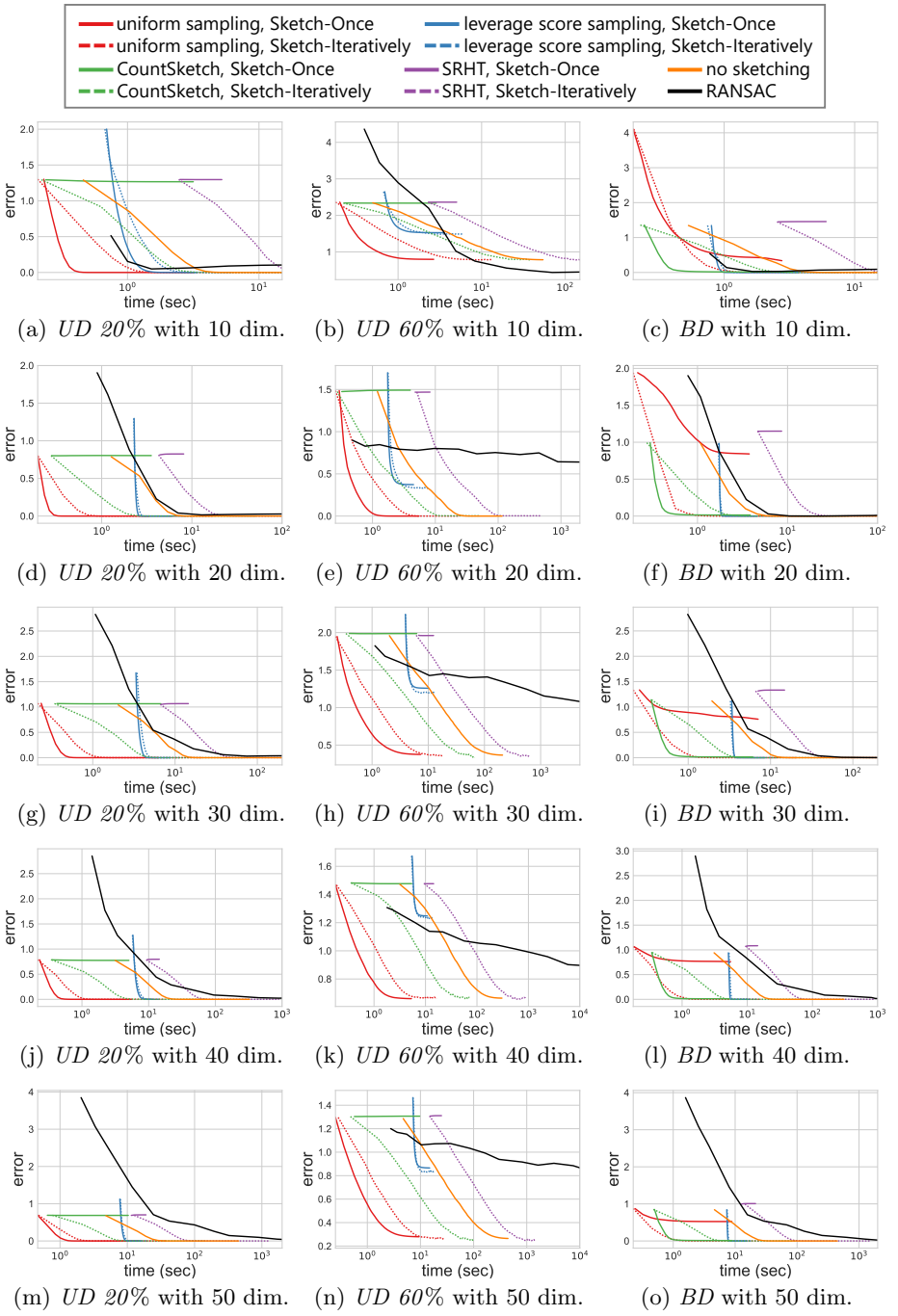
In this supplementary material,

1. we show and discuss additional results for the residual minimization experiments described in our main paper's Sec. 5.1.
2. We further show the versatility of our method by introducing two more, highly practically relevant tasks;
   - matrix completion for Sec. 5 and
   - rotation averaging for Sec. 6.

Residual minimization (paper Sec. 5.1)

In the paper, we showed residual minimization results for matrix $\mathbf{A} \in \mathbb{R}^{10^6 \times 40}$. To evaluate the behavior of our method and RANSAC in more detail, we show results with different sizes of matrix $\mathbf{A}$. In this additional experiment, we fixed the number of $\mathbf{A}$'s rows and varied the number of its columns (the number of unknowns / dimensions) from $\{10, 20, 30, 40, 50\}$. For each dimension, we then generated three types of data: *uniform data 20%* (*UD 20%*), *uniform data 60%* (*UD 60%*), and *biased data* (*BD*) as described in the paper. Each result is an average of 10 trials with different random seeds.

The results are summarized in Fig. S1. While RANSAC works well for cases with a small number of dimensions, its convergence becomes significantly slower as the number of dimensions increases. Figure S1b shows a rare case where RANSAC reaches a lower error than the ℓ_1 minimization but overall, ℓ_1 minimization outperforms RANSAC in these settings and our sketched IRLS often accelerates the computation by many orders of magnitude. Note that the x-axis is always logarithmic.



(a) *UD 20%* with 10 dim. (b) *UD 60%* with 10 dim. (c) *BD* with 10 dim.

(d) *UD 20%* with 20 dim. (e) *UD 60%* with 20 dim. (f) *BD* with 20 dim.

(g) *UD 20%* with 30 dim. (h) *UD 60%* with 30 dim. (i) *BD* with 30 dim.

(j) *UD 20%* with 40 dim. (k) *UD 60%* with 40 dim. (l) *BD* with 40 dim.

(m) *UD 20%* with 50 dim. (n) *UD 60%* with 50 dim. (o) *BD* with 50 dim.

Fig. S1: Error $\frac{1}{d}\|\mathbf{x}^* - \mathbf{x}^{(t)}\|_2$ over time in residual minimization with ℓ_1 minimization (with/without sketching) and RANSAC. Dimensions indicate the number of unknowns $\mathbf{x}$ in $\mathbf{A}\mathbf{x} \simeq \mathbf{b}$.

Matrix completion with low-rank assumption (paper Sec. 5)

This task is similar to low-rank approximation (paper Sec. 5.2), except that the matrix has *missing* elements and the goal is to recover them by assuming that the original matrix has low-rank structure. This task is highly relevant, for example, in recommendation system [7] and image inpainting [1].

Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be a matrix of rank $r < \min(m, n)$ with missing elements. We estimate $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ using the known elements in $\mathbf{M}$ by solving

$$\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} \left|\mathbf{M}_{ij} - (\mathbf{U}\mathbf{V}^\top)_{ij}\right|,$$

where $_{ij}$ is the $(i,j)^{\text{th}}$ element of a matrix and Ω is the index set of $\mathbf{M}$'s known elements.

We generated $\mathbf{M} \in \mathbb{R}^{3\cdot10^4 \times 3\cdot10^4}$ with $\text{rank}(\mathbf{M}) = 30$, randomly eliminated 70 % of the elements to be treated as missing, and then corrupted the low-rank structure by flipping signs of 10 % of the not missing elements. The sketching parameter s is set to $s = 30,000 \times 3\,\% = 900$. To assess our algorithm, we define the *error* as in Sec. 5.2 experiment of the paper. The result is summarized in Fig. S2. Sketch-once and sketch-iteratively with uniform sampling converge more than 2 times faster than canonical IRLS. For CountSketch, sketch-iteratively reaches higher accuracy than sketch-once.

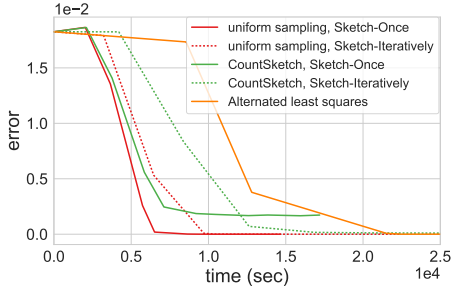
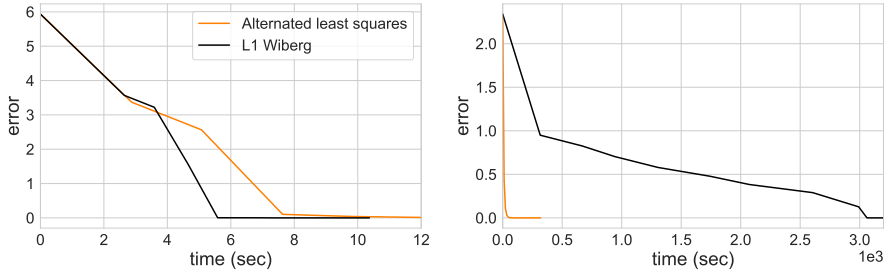


Fig. S2: Error over time in ℓ_1 matrix completion on synthetic data

As a comparison method, we chose the ℓ_1 Wiberg method [4]. While our method is based on Ke and Kanade's [6] Alternated Least Squares (ALS), ℓ_1 Wiberg uses the Gauss-Newton method when updating $\mathbf{U}$. The ℓ_1 Wiberg method is effective for small matrices, but we found it to be difficult to use for large matrices due to poor memory efficiency and optimizer limitations. We show a comparison between canonical ALS and ℓ_1 Wiberg for small matrices.

We generated two matrices: (a) $\mathbf{M} \in \mathbb{R}^{25\times25}$ with $\text{rank}(\mathbf{M}) = 2$, randomly eliminated 25 % of the elements and added noise to 10 % of the non-missing elements and (b) $\mathbf{M} \in \mathbb{R}^{100\times100}$ with $\text{rank}(\mathbf{M}) = 5$, randomly eliminated 50 % and added noise to 10 % of the non-missing elements. The results are summarized in Figures S3a and S3b. In Fig. S3a the ℓ_1 Wiberg converges faster than ALS. However, ALS becomes faster as the matrix size increases as shown in Fig. S3b. ALS is further accelerated by our Sketched IRLS method as we showed above.



(a) 25×25 matrix (b) 100×100 matrix

Fig. S3: Error over time in ℓ_1 matrix completion on small synthetic datasets, 25×25 (*left*) and 100×100 (*right*).

Rotation averaging (paper Sec. 6)

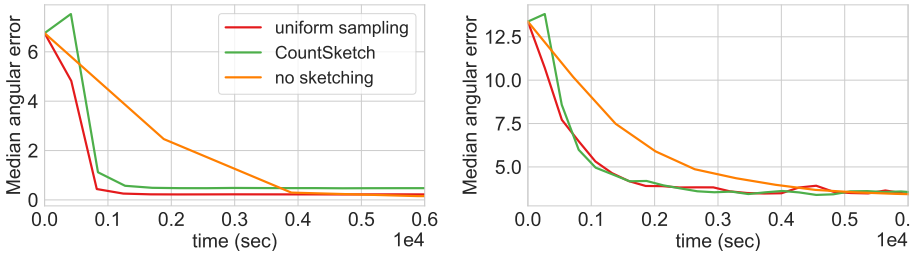
In structure from motion, a famous method is bundle adjustment, which simultaneously optimizes the scene's 3D structure and camera poses. While bundle adjustment needs a good initialization and solves a nonlinear problem, in recent years an approach that divides the problem into camera rotation estimation and structure estimation has attracted attention. Rotation averaging finds each camera's absolute rotation from the relative rotations between cameras. We denote the absolute rotation of camera i as $\mathbf{R}_i$ and the relative rotation between cameras i and j as $\mathbf{R}_{ij}$. Then, $\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}$ holds. Now we consider N absolute rotations $\mathbf{R}_1, \dots, \mathbf{R}_N$ and the problem is

$$\min_{\mathbf{R}_1,\dots,\mathbf{R}_N} \sum_{(i,j)\in\Omega} d(\mathbf{R}_{ij}, \mathbf{R}_j \mathbf{R}_i^{-1}).$$

$d(\cdot,\cdot)$ is a distance function and Ω is the set of camera pairs with known relative rotations. From the relationship between the Lie group and the Lie algebra in 3D rotation, we can express rotation matrices as $\mathbf{R} = e^{[\mathbf{r}]_\times}$, where $[\cdot]_\times$ creates a skew-symmetric matrix and $\mathbf{r} \in \mathbb{R}^3$. Considering exponential relationship for matrices, $e^{\mathbf{X}} e^{\mathbf{Y}} = e^{\mathbf{X}+\mathbf{Y}}$ does not generally hold unless the matrices $\mathbf{X}$ and $\mathbf{Y}$ are commutative. When $\mathbf{X}$ and $\mathbf{Y}$ are noncommutative, we can use the Baker-Campbell-Hausdorff (BCH) formula, *i.e.*, $e^{\mathbf{X}} e^{\mathbf{Y}} = \exp\left[(\mathbf{X}+\mathbf{Y}) + \frac{1}{2}[\mathbf{X},\mathbf{Y}] + \dots\right]$, where $[\cdot,\cdot]$ is a commutator. Using the first-order approximation of the BCH formula, we can transform $\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^{-1}$ into $\mathbf{r}_{ij} \approx \mathbf{r}_j - \mathbf{r}_i$ [5]. Let $\mathbf{r}_{\text{abs}} = \left[\mathbf{r}_1^\top, \dots, \mathbf{r}_N^\top\right]^\top$. Then, we can rewrite $\mathbf{r}_{ij}$ as

$$\mathbf{r}_{ij} = [\dots, -\mathbf{I}, \dots, \mathbf{I}, \dots]\, \mathbf{r}_{\text{abs}}, \tag{1}$$

where $\mathbf{I}$ is an $\mathbb{R}^{3\times 3}$ identity and all other parts are zero. We construct Eq. (1) for all relative rotations and stack them, resulting in an overdetermined linear problem $\mathbf{b} = \mathbf{A}\mathbf{r}_{\text{abs}}$. Chatterjee and Govindu [2] described the algorithm in detail and showed the effectiveness of ℓ_1 minimization. In the following we speed it up with matrix sketching.



(a) Synthetic dataset (b) Quad dataset

Fig. S4: Median angular error over time in rotation averaging on synthetic data (*left*), on the Quad data (*right*)

Result In this experiment, we used two types of datasets. The first one, a synthetic dataset with known ground truth, has $6,000$ cameras and $589,789$ camera pairs with known rotations. We added two types of noise: small Gaussian noise $\mathcal{N}(0°, 1°)$ for each axis to all relative rotations and big uniform random noise in the range $[-180°, 180°]$ for each axis to $4\,\%$ of all relative rotations.

The second is the Quad dataset [3], which consists of $5,530$ cameras and $222,044$ pairs. This dataset also provides the result of bundle adjustment and we treat it as ground truth, as Chatterjee and Govindu [2] also did. We set the sketching sizes s to $589,789 \times 20\,\%$ and $222,044 \times 40\,\%$, respectively.

Figures S4a and S4b show the convergence behavior on the synthetic and the Quad dataset in terms of median angular error over time. Uniform sampling sketching works well and converges 3–4 times faster than no sketching in Fig. S4a. In Fig. S4b, both sketching methods show slightly unstable behavior due to the amount of noise but both also descend fast and the final solution quality is very similar to that without sketching.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. SIGGRAPH (2000) 3
2. Chatterjee, A., Govindu, V.M.: Robust relative rotation averaging. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **40**(4), 958–972 (2018) 4, 5
3. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 5
4. Eriksson, A., van den Hengel, A.: Efficient computation of robust weighted low-rank matrix approximations using the L_1 norm. Transactions on Pattern Analysis and Machine Intelligence (PAMI) **34**(9), 1681–1690 (2012) 3
5. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2004) 4
6. Ke, Q., Kanade, T.: Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2005) 3

7. Ramlatchan, A., Yang, M., Liu, Q., Li, M., Wang, J., Li, Y.: A survey of matrix completion methods for recommendation systems. Big Data Mining and Analytics **1**(4), 308–323 (2018) 3