# Batched Incremental Structure-from-Motion

Hainan Cui[1], Shuhan Shen[1, 2], Xiang Gao[1, 2], and Zhanyi Hu[1, 2]

[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing , China
[2]University of Chinese Academy of Sciences, Beijing, China
{*hncui, shshen, xiang.gao, huzy*}*@nlpr.ia.ac.cn*

## Abstract

*The incremental Structure-from-Motion (SfM) technique has advanced in both robustness and accuracy, but the efficiency and scalability remain its key challenges. In this paper, we propose a novel batched incremental SfM technique to tackle these problems in a unified framework, where two iteration loops are contained. The inner loop is a tracks triangulation loop, where a novel tracks selection method is proposed to find a compact subset of tracks for the bundle adjustment (BA). The outer loop is a camera registration loop, where a batch of cameras are simultaneously added to alleviate the drifting risk and reduce the running times of BA. By the tracks selection and batched camera registration, we find these two iteration loops converge fast. Extensive experiments demonstrate that our new SfM system performs similarly or better than many of the state-of-the-art SfM systems in terms of camera calibration accuracy, while is more efficient, robust and scalable for large-scale scene reconstruction.*

## 1. Introduction

Structure-from-Motion (SfM) technique has been successfully used for the reconstruction of large uncontrolled image collections [1, 23]. A typical pipeline for SfM usually consists of four steps: image feature detection and matching, camera poses initialization, tracks triangulation and bundle adjustment. Based on the difference of camera poses initialization manner, SfM could be roughly divided into two categories: incremental and global.

Incremental SfM [32, 34, 35, 38, 42] usually begins from two-view reconstruction, then iteratively registers cameras, and performs bundle adjustment (BA) for each registration to refine the camera poses and scene structure. While accurate, it may suffer from drift, inefficiency and poor scalability when it is used for large-scale scene reconstruction. In comparison, global SfM [9, 10, 11, 12, 14, 39] utilizes the epipolar geometry graph (EG), where nodes correspond to cameras and edges link matched camera pairs, to estimate the camera poses simultaneously. While global rotation averaging [7, 18, 19, 30] has been well studied in the literatures, global translation averaging [13, 33, 39] suffers from epipolar geometry outliers, which may leave many cameras un-calibrated.


Figure 1. Our result on Quad [9], where 5971 images are registered out of 6514 images. (a) shows the result with tracks selection (257K points), (b) shows the result without tracks selection (3048K points). The red cones show calibrated camera poses.

Arguably, the robustness and accuracy advantages in the incremental SfM mainly benefits from RANSAC technique to discard bad epipolar geometries and repeated bundle adjustment to refine camera poses. However, when the system handles large-scale scene reconstruction, the incremental manner suffers from error accumulation, which is a main factor to cause the scene drift [24]. In addition, the repeated time-consuming bundle adjustment makes both reconstruction efficiency and scalability poor. Thus in this paper, we propose a batched incremental SfM method (BSfM for short), here the 'batch' means that a batch of cameras are simultaneously registered and refined in each camera registration step, rather than adding cameras one by one in many state-of-the-art incremental SfM systems [34, 38, 42]. In this way, each camera registration step could be considered

2475-7888/17/31.00 ©2017 IEEE
DOI 10.1109/3dv.2017.00032

as using a global manner to reduce the drifting risk. Our main contribution includes: (1) **batched camera registration**, (2) **batched tracks selection**.

The batched camera registration means that instead of performing carefully next view selection [15, 34, 42], we propose to simultaneously calibrate a batch of cameras in each camera registration step, namely for these cameras that have sufficient 2D-3D correspondence, we treat them equally and compute their corresponding camera poses simultaneously. Since the error accumulation is the inherent problem in the incremental manner, the camera pose registered by the scene points in former iteration is more accurate than that registered by the points in latter ones. Thus, our batched camera registration aims to make the cameras utilize more accurate scene points for the registration. In addition, by registering multiple cameras in each iteration, the times of camera registration will be reduced, and concomitantly, the error accumulation problem will be alleviated.

The batched tracks selection means that in each camera registration step, only a subset of tracks is selected and refined, rather than considering all the visible tracks. For large-scale scene reconstruction, the tracks are usually redundant for camera calibration and memory-consuming for BA, thus tracks selection is vital for both SfM efficiency and scalability. To our best knowledge, we are the first to fuse the tracks selection technique into incremental SfM. To guarantee the success of next camera registration step, our tracks selection considers two aspects: one is to cover the calibrated cameras to refine their camera poses, the other is to cover the cameras that are going to be registered in the next step. In the selection process, we prefer the longer tracks because they have a larger consistency than those of short ones. Furthermore, by using tracks selection, the number of constraint equations in BA is dramatically reduced and the space scalability is greatly improved.

Note that instead of performing incremental SfM on the image feature matches [34, 35, 38], we propose to use tracks in our whole batched SfM pipeline. The reason contains the following three aspects: (1) since the length of track is usually long, using tracks could get more candidates in the camera registration step, which benefits for our batched camera registration; (2) in the tracks construction process, many erroneous or ambiguous feature matches have been filtered in advance [29, 31, 37]; (3) a track is a collection of feature matches, its consistency is more convincing to be a real inlier than that of a feature match, which is more helpful for our tracks selection and batched BA.

Fig. 1 illustrates our reconstruction result on Quad [9]. To show the effectiveness of our tracks selection, we show the result reconstructed by embedding our tracks selection module in Fig. 1$(a)$, and show that without tracks selection module in Fig. 1$(b)$. Qualitatively, the scene structures are similar, and quantitatively, the median camera position accuracies in $(a)$ and $(b)$ are both 0.69m. The similar scene structure and similar calibration accuracy indicate that the tracks are too redundant for the SfM task solving, and from Fig. 1 we can see that our method only uses 8% of the full tracks. For the reconstruction efficiency, benefiting from batched camera registration and tracks selection, our batched SfM is 7 times faster than COLMAP [9], and 80 times faster than Bundler [35].

## 2. Related Work

**Initial Seed Reconstruction**: Incremental SfM usually initializes the scene model with a carefully seed selection [4, 17, 35]. Bundler [35] aims to find an image pair with enough feature match inliers but includes few homographic feature matches. Haner *et al*. [17] presents a selection rule based on covariance propagation, and points out that a well-determined camera should have both small estimated covariance and low re-projection error. However, such methods do not consider the space distribution of camera positions. To make more cameras be added in the batched camera registration step, we utilize the epipolar geometry graph to select a good image pair with more neighbors.

**Image Registration**: Based on a metric reconstruction, new images are registered by solving the Perspective-n-Point (PnP) problem [16, 25, 26] using the 2D-3D correspondences. Conventional incremental SfM methods [15, 34, 38, 42] register one camera at each camera registration step, then perform either local bundle adjustment or global bundle adjustment. For example, Bundler [35] registers the camera with the most visible correspondences, and COLMAP [34] simultaneously keeps track of the number of visible points and their distribution in each candidate image. While accurate, this 'one by one' manner accumulates the error constantly, and usually performs BA too many times. Hence, we propose a 'batch by batch' manner to improve the SfM efficiency and alleviate the drifting problem.

**Tracks Triangulation & Selection**: After camera registration, a newly registered image could increase scene coverage by extending 3D points through tracks triangulation [3, 22, 27, 28]. Conventional incremental SfM methods [34, 38, 42] usually refine all the triangulated tracks in BA, which suffers from both time-consuming and memory-consuming. In fact, the tracks are usually redundant, especially for those high-resolution images, a mass of tracks usually puts too much burden on the BA, while are not helpful for the camera calibration, and sometimes many tracks outliers may impede the SfM task solving. Thus, we propose to make a tracks selection for the BA.

**Bundle Adjustment**: BA [8, 40] is a joint non-linear refinement of camera parameters and point parameters that minimizes the discrepancy between scene structure and measured 2D image features. The LM [20] algorithm is usually used to solve this problem, and the special structure

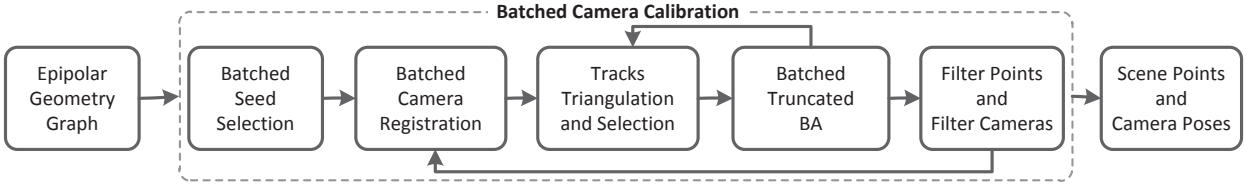



Figure 2. Pipeline of our batched incremental SfM system.

of parameters motivates the Schur complement trick [5]. As proposed in VSFM [42], by indirect algorithm – Preconditioned Conjugate Gradient (PCG) [6, 43], the time-complexity of large-scale BA could be close to linear. Based on this approach, we propose to cluster the images first, then truncate the LM optimization with a fixed maximum number of iterations to further speed-up the camera calibration.

## 3. Batched Incremental SfM

Fig. 2 shows the pipeline of our BSfM, where two loops are included. The inner loop is to iteratively run the tracks triangulation, selection and BA, and the outer loop is the iterative camera registration. Note that only when the inner loop converges, the next iteration of outer loop could begin.

Each edge of the epipolar geometry graph (EG) contains corresponding feature matches and epipolar geometry, including the relative rotation $\mathbf{R}_{ij}$ and translation $\mathbf{T}_{ij}$, which is obtained by essential matrix decomposition. Based on the EG, tracks are constructed by the union-find algorithm [31].

### 3.1. Batched Seed Selection

To obtain a good initial scene reconstruction, the camera seeds should have both accurate camera poses and wide baselines. In addition, in order to make more cameras be registered in the next step, the visible scene of a good camera pair should be seen by as many cameras as possible.

Given the EG, global camera rotations $\mathbb{R} = \{\mathbf{R}_i, i = 1...N\}$ could be estimated from relative rotations [7]. As proposed in [14, 39, 41], these estimations are sufficiently accurate to be an initial guess for camera orientations. Thus to find a good camera pair, we filter the grossly erroneous epipolar edges first. For an epipolar edge $E_{ij}$, the geodesic error [19] is computed by $acos(\|\mathbf{R}_{ij} - \mathbf{R}_j\mathbf{R}_i^T\|_F)$. Then, those epipolar edges with a large geodesic error (*e.g.*, >15deg), are filtered. To demonstrate the effectiveness of this filtering step, we evaluate both the edges before and after filtering by computing the geodesic error $acos(\|\mathbf{R}_{ij} - \mathbf{R}_{ij}^{gt}\|_F)$, where $\mathbf{R}_{ij}^{gt}$ is the ground-truth of $\mathbf{R}_{ij}$. Fig. 3 shows the cumulative distribution functions (CDF) on the geodesic error of two public datasets Alamo and RomanForum [41], where 'EG' and 'EG_Filter' respectively

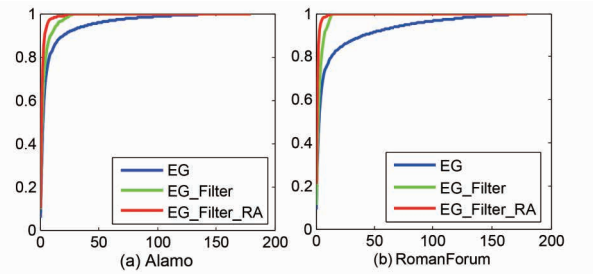



Figure 3. The cumulative distribution function (CDF) on the geodesic rotation error of datasets Alamo and RomanForum [41].

denotes the CDF before and after filtering, from which we can see that the ratio of edge inliers is increased by filtering. Next, we replace the relative rotations $\mathbf{R}_{ij}$ as: $\mathbf{R}_{ij} = \mathbf{R}_j\mathbf{R}_i^T$, the corresponding CDF is shown by the red curve in Fig. 3, we can see that the ratio of real inliers increases further.

To achieve a robust reconstruction, selected camera seeds should also have a wide baseline. Given camera intrinsic parameters and global camera rotations, we get a ray for each image feature, and then the triangulation angle between any pairwise feature matches is obtained. For each epipolar edge, we use the median triangulation angle of its feature matches to measure the length of baseline. However, it is undesirable when the direction of baseline is nearly parallel with the optical axis, *i.e.* moving the camera along with the optical axis. Thus, when the normalized relative translation on an epipolar edge has a large z-axis component, the corresponding pair of cameras is ignored in the seeds selection. Besides, in order to register more cameras in the next step, camera seeds should have more neighbors. To this end, EG are augmented by tagging each epipolar edge with a property $\rho_{ij}$ to denote the density of neighboring cameras. Let $n_i$ be the number of neighbors of camera $i$ in the EG, then $\rho_{ij}$ is measured by $\min\{n_i, n_j\}$.

Overall, a batch of camera seed candidates are selected by the following steps. First, those epipolar edges that satisfy anyone of the following three conditions are filtered: (1) a small median angle (*e.g.*, smaller than 3.0 degrees); (2) an erroneous relative rotations (*e.g.*, geodesic error is larger than 15.0 degrees); (3) a large z-axis component of relative

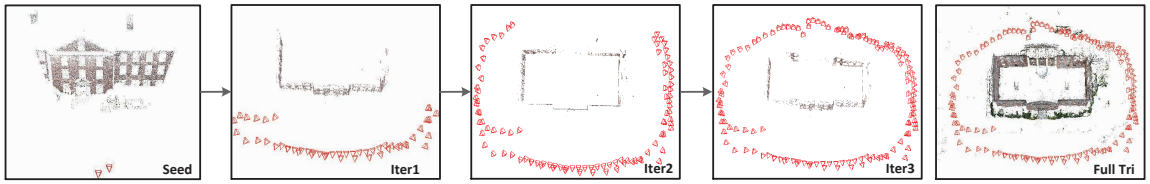



Figure 4. An illustrative reconstruction process of our BSfM system, where the camera registration step (outer loop) consists of 3 iterations . To show the sparsity of selected tracks, we triangulate the full tracks by the final calibrated camera poses and show the result in the last. Note that this full triangulation step is not necessary in our system, it is only used to better show the scene structure.

translation (*e.g.*, $|\mathbf{T}_{ij}^{z}| \geq 0.9$). By such a filtering step, we get an initial batch of camera seeds. Then, the relative rotation on each epipolar edge is replaced by the global camera rotation estimations. Finally, by sorting $\rho_{ij}$, a ranked batch of camera seeds are obtained, and we reconstruct the camera seeds along the ranking list. For each pair of cameras, the relative translation is refined first by the inliers of epipolar geometry, then we triangulate the feature matches and perform a two-view BA. When the ratio of scene point inliers is larger than $\gamma_1$ (*e.g.*, 0.5) and the number of inliers is larger than $\gamma_2$ (*e.g.*, 200), the seed reconstruction is considered as a success, otherwise we check the next candidate.

### 3.2. Batched Camera Registration

In stead of performing carefully next view planning, which has been studied in many literatures [15, 17, 34], we propose to add a batch of cameras in the registration step to reduce the error accumulation risk as much as possible.

Given reconstructed scene points, any cameras that observe sufficient triangulated points could be registered in this step. In our work, to benefit from RANSAC technique, all the cameras that could see more than 12 scene points are considered as a candidate for the camera registration. For each candidate $C_i$, we utilize the P3P algorithm [16] to get its initial camera pose $\{\mathbf{R}_i, \mathbf{T}_i\}$. If the ratio of inliers is larger than 0.5, we consider it is possibly right and perform the following optimization.

By keeping both the camera intrinsic parameters $\mathbf{K}_i$ and 3D scene points inliers $\mathbf{X}_j$ fixed, the initial camera poses $\{\mathbf{R}_i, \mathbf{T}_i\}$ is further refined by minimizing the discrepancy between the observed 2D image features $\mathbf{x}_{ij}$ and 3D reconstructed scene points $\mathbf{X}_j$:

$$\min_{\mathbf{R}_i, \mathbf{T}_i} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\mathbf{x}_{ij} - \gamma(\mathbf{K}_i, \mathbf{R}_i, \mathbf{T}_i, \mathbf{X}_j)\|_{huber}, \qquad (1)$$

where $\gamma()$ function is the projecting function. As the initial camera poses estimation has already found the maximum number of inliers, thus for the robustness concern, when the geodesic rotation error between the initial estimation and after optimization is small (*e.g.*, geodesic error is smaller than 5.0 degrees), we consider this camera registration is correct, otherwise it is considered as failed. Since the registration for different cameras is independent, the batched registration is performed in parallel, and we find its time-cost is negligible when compared to that of BA.

### 3.3. Tracks Triangulation and Selection

After camera registration, a batch of newly registered cameras could extend the scene by tracks triangulation. Here we use a RANSAC-based triangulation method to compute the 3D point position for each track covering two or more calibrated views. For each iteration in RANSAC, we randomly choose two visible views, and then compute the angle between two projection rays. If the angle is larger than 2.0 degrees, it is currently considered as well-conditioned and use the DLT [20] to triangulate. Then, both the number of its consistent measurements and the cheirality of corresponding views are checked. Note that all the cheirality [21] of visible calibrated cameras in the track should be positive. For each track, we find a best point that has the largest number of consistent measurements, and when the maximum projection error of its visible views is smaller than $\delta$, we consider it as a current track inlier. However, tracks are usually redundant and an extremely amount of tracks usually makes BA time-consuming and memory-consuming. To improve the SfM efficiency and scalability, we propose to find a compact subset of tracks in the guarantee of keeping camera calibration accuracy.

Considering a conventional camera model, 3DOF (degree of freedom) for camera rotation, 3DOF for camera translation, 1DOF for camera focal length and 2DOF for camera distortion, the minimum number of constraints in BA to refine a camera is only 5. However in fact, the tracks are usually cover each camera hundreds of times. Thus, in order to reduce the number of constraints in BA, we propose to find a subset of tracks to cover each camera $K$ times. Since the manner of SfM is incremental, we should not only cover the cameras that have already been calibrated, but also cover the cameras that are going to be registered in the next. Let $S$ be the camera set that we need to cover currently.

For each track, we check the number of its visible views in $S$, and denote them as effective views. Based on the num-

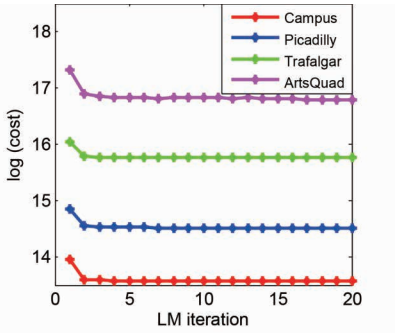



Figure 5. Four examples of the BA cost change, where Campus [14], Piccadilly and Trafalgar [41], ArtsQuad [9] has 1040, 2176, 4886 and 5971 images, respectively.

ber of effective views, current track inliers are ranked first in a descending order. When some tracks with the same number of effective views, they are re-ranked by their corresponding re-projection error. Then, we check each track along the ranking list, and select it if one of its effective views is not covered by $K$ times. The iteration converges when all effective views are covered by $K$ times or all the tracks have been checked. The reason of choosing longer tracks first contains two-fold: one is the longer tracks cover more views, which further reduces the number of tracks in BA; the other is that a longer track has a larger consistency than short ones, which is more likely be a real inlier. Fig. 4 shows an illustrative reconstruction result on the dataset Building [44], from which we could see that only 3 iterations are needed in our batched camera registration (outer loop), and the tracks selected in the iteration are extremely sparse when compared with the full tracks.

### 3.4. Batched Bundle Adjustment

To mitigate the impact of accumulated errors and refine inaccurate camera pose estimations, we perform a batched BA after each camera registration. To utilize the consistency of camera models, the cameras are initially clustered by the priori intrinsic parameters. If two cameras with the same priori focal length, principle point, and the manufacturer, we consider them as a same model and only refine one camera intrinsic model in BA.

Conventional BA [2] performs iterative LM in the optimization, and the convergence criterion is usually critical. Fig. 5 shows the change of BA cost on four large-scale datasets, from which we could see all the BA processes have intuitively converged in about 5–10 iterations, while actually, the critical convergence criterion is still not reached, and lots of time are spent in the latter iterations. BA has already adopt linear approximation to the non-linear problem solving, thus the convergence could not be considered as find the real minimum. In our system, since a more accurate subset of tracks is selected for BA, we find that the cost in BA usually decrease very fast in the first several iterations, while decreases slowly afterwards. Thus, in order to speed-up BA, we truncate it by a fixed maximum number of LM iterations ($T$), and we find that it is practical in all of our testing datasets (in our work, $T$ is set to 10). Furthermore, considering the potential outliers, the Huber function [20] is further employed as the robust loss function in our BA:

$$\min_{\mathbf{K}_s,\mathbf{R}_i,\mathbf{T}_i,\mathbf{X}_j} \sum_{s=1}^{S}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\|\mathbf{x}_{ij}-\gamma(\mathbf{K}_s,\mathbf{R}_i,\mathbf{T}_i,\mathbf{X}_j)\|_{huber}, \tag{2}$$

where $S$ is the number of camera intrinsic models, and $\delta_{ij}=1$ if camera $i$ observes $X_j$, otherwise $\delta_{ij}=0$.

### 3.5. Iterative Re-triangulation and Re-selection

After BA, the camera poses become more accurate, thus we propose to perform a re-triangulation step for tracks to decrease the accumulated error, and a further re-selection step to speed-up the next BA. The convergence criterion on this inner loop of our BSfM is set to evaluate the IoU (Intersection over Union) value on the selected tracks:

$$IoU=\frac{H^i\bigcap H^{i-1}}{H^i\bigcup H^{i-1}}, \tag{3}$$

where $H^i$ denotes the selected tracks in the $i^{th}$ iteration. When the IoU is larger than $0.9$, the inner loop is considered as converged. In our work, we find this inner loop usually converges in 2–4 iterations.

### 3.6. Cameras Filtering and Points Filtering

Since only a small fraction of points are used for camera registration, sometimes the calibrated camera parameters are not reliable. Considering the robustness, we filter them by using some priori constraints. For the camera intrinsic parameters, the refined focal length cannot be changed too much, and the calibrated distortion parameters cannot be too large. Thus, when the change of focal length over the priori focal length is larger than 80%, or one of the distortion parameters is larger than 1.0, the camera is considered as a failed calibration. Those failed cameras are set unregistered and could be re-registered in the following iterations. Then, based on the refined camera poses, those points satisfying anyone of the following conditions are filtered: (1) the maximum projection error is larger than 8.0 pixels; (2) the maximum triangulation angle is smaller than 2.0 degrees; (3) one of visible views has a negative cheirality.

## 4. Experiments

All the experiments are performed on a PC with an Intel Xeon E5-2603 CPU and 32G RAM. For the fairness, all methods in comparison use the same feature matches for reconstruction. The parameter $K$ in the tracks selection is set to 100.

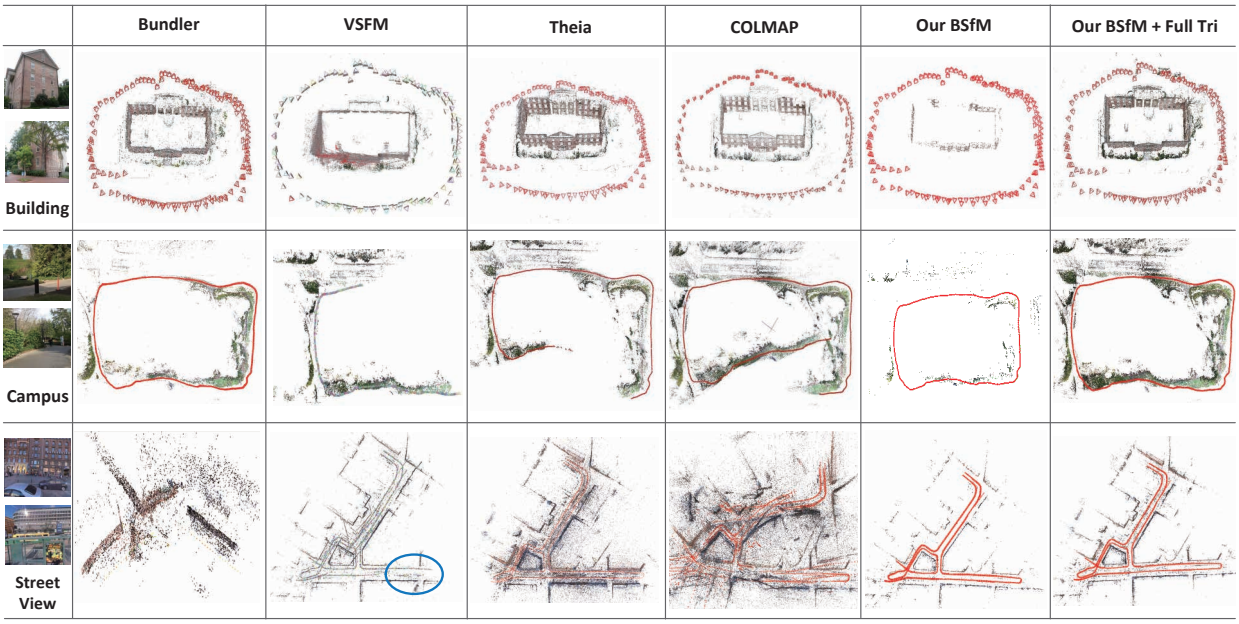



Figure 6. Reconstruction results on three sequential image data: Building [44], Campus [14] and StreetView [12].

Table 1. Camera calibration accuracy on benchmark images [36]. $C_{err}$ denotes the median camera position errors in millimeters. $R_{err}$ denotes the median camera rotation error in degrees.

| Method | Accuracy (mm — deg) | | | | | |
|---|---|---|---|---|---|---|
| | FountainP11 | | HerzJesuP25 | | CastleP30 | |
| | $C_{err}$ | $R_{err}$ | $C_{err}$ | $R_{err}$ | $C_{err}$ | $R_{err}$ |
| Bundler [35] | 7.0 | 0.28 | 21.9 | 0.25 | 206.1 | 0.36 |
| VSFM [42] | 36.0 | 0.28 | 55.0 | 0.29 | 264.0 | 0.40 |
| Theia [38] | **1.9** | 0.08 | **4.7** | 0.07 | 21.5 | **0.05** |
| COLMAP [34] | 4.9 | 0.30 | 23.6 | 0.40 | 99.3 | 0.34 |
| our BSfM | **1.9** | **0.06** | **4.7** | **0.04** | **20.6** | 0.06 |

### 4.1. Evaluation on Benchmark Data

To quantitatively evaluate our method, we perform our method on three median-scale benchmark image datasets [36], and show the results in Table 1. From this table, we can see that for the camera positions accuracy, our method achieves the best among the five methods in comparison. For the camera rotation accuracy comparison, we utilize the median geodesic error [19] as the evaluation criterion. From the results, our method performs similar with Theia, while is superior than the other three SfM approaches.

We also evaluate our method on the large-scale image dataset Quad [9], where 348 camera positions are measured by a survey-quality differential GPS (with an accuracy of about 10cm). The reconstruction result is shown in Fig. 1, and we achieve the best camera position accuracy: DISCO [9] 1.16m, Bundler [35] 1.01m, VSFM [42] 0.89m, COLMAP [34] 0.85m, and our BSfM 0.69m.

### 4.2. Evaluation on Sequential Data

We demonstrate our BSfM system on three sequential image datasets, including Building with 128 images from [44], Campus with 1040 images from [14], and StreetView with 2468 images from [12]. We compare our method with four state-of-the-art incremental SfM methods: Bundler [35], VSFM [42], Theia [38], COLMAP [34]. Fig. 6 shows the reconstruction results on these datasets. To qualitatively compare with other methods, we triangulate all tracks by our calibrated camera poses and show the result in the last column.

For the Building, most methods successfully reconstruct the scene, while VSFM fails in the left-bottom of scene. In addition, we achieve the best efficiency: Bundler 3.6 hours, Theia 0.6 hours, COLMAP 0.5 hours and our BSfM 0.08 hours. Fig. 7 shows the IoU (Eq. 3) and the number of selected tracks in the camera registration, from which we could see that the inner loop converges fast (when $IoU \geq 0.9$) and for each new outer iteration, there is a large drop of IoU because new tracks are selected by the newly registered cameras. Furthermore, with the iterations in the outer loop going on, we find the number of selected tracks decreases, and the reason is that after each camera registration, we select much longer tracks. The median track length for the three outer iterations is 17, 19, 21, respectively.

For the Campus, it is more challenging because many trees exist in the scene and the camera motion trajectory is a loop. As a result, many erroneous feature matches are easily appeared in the matched trees. From the reconstruction results in Fig. 6, we can see that VSFM, Theia and COLMAP

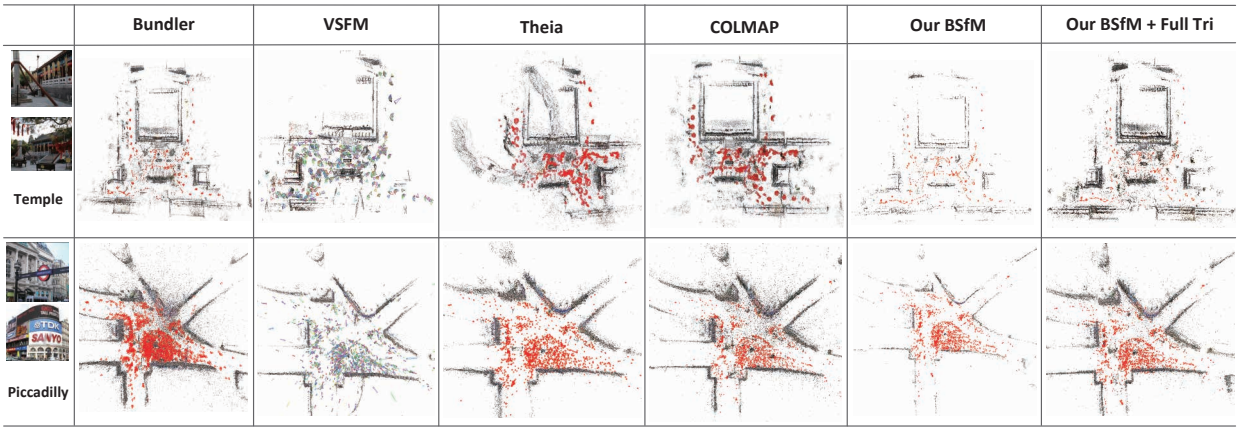



Figure 8. Reconstruction results on unordered image data: Temple [12] and Piccadilly [41].

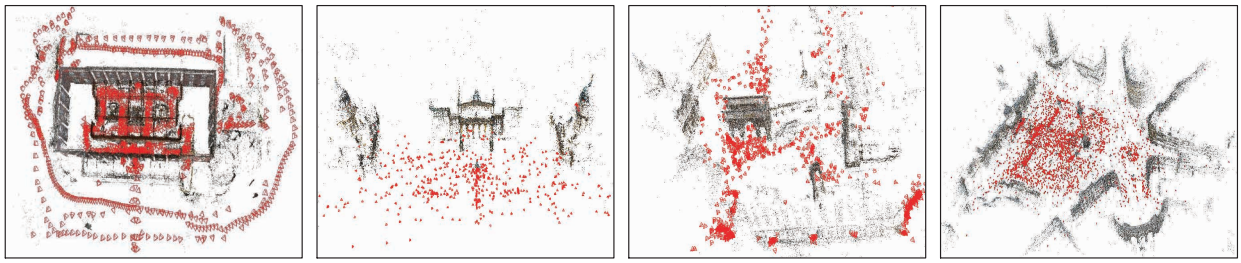



Figure 9. (a) graham-hall [34] is a sequential data with 1273 images, captured from outdoor to indoor; (b) [41] is an ambiguous dataset with many symmetric textures; (c)-(d) [41] are unordered image datasets, with 1134 and 5433 images, respectively.

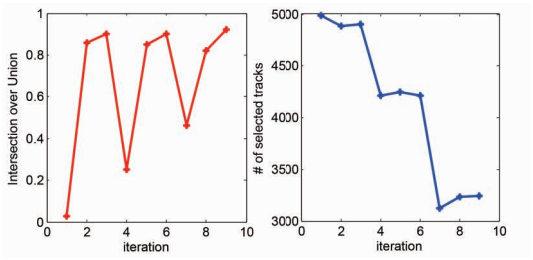



Figure 7. The left figure shows the change of IoU, and the right shows the number of selected tracks in the camera registration.

fail on this dataset, while Bundler [35] and our BSfM successfully reconstruct the scene and achieve the loop. Considering the SfM efficiency, the time-cost of Bundler is 61.0 hours, while only 0.4 hours is spent by our BSfM system.

For the StreetView, many symmetric textures exist in the facade of buildings, hence the feature matches are easily contaminated. From Fig. 6, we can see that both Bundler and COLMAP fail on this dataset, and the reconstructed scene is incomplete in the VSFM (marked by a blue ellipse), while Theia and our BSfM could successfully reconstruct the scene. By comparing the efficiency, the time-cost of Theia is 3.5 hours, while our BSfM only needs 1.0 hours.

In conclusion, for these sequential testing datasets, our method achieves the best efficiency and robustness.

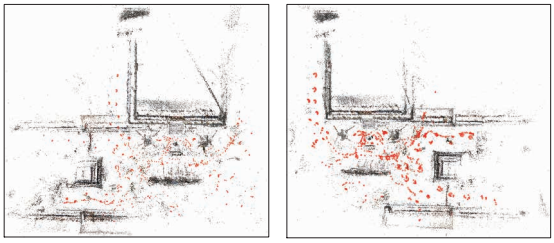
### 4.3. Evaluation on Unordered Data

To further demonstrate the scalability of our method, we evaluate it on two large-scale unordered internet image datasets [41], including Piccadilly with 2508 images and Trafalgar with 5433 images, and a challenging dataset Temple [12] with many symmetric textures and trees.

For the qualitative comparison, Fig. 8 shows the reconstruction results on both Temple and Piccadilly, from which we can see all the incremental SfM methods reconstruct the Piccadilly successfully, while for the challenging dataset Temple, VSFM, Theia and COLMAP fail. The reason for this failure is that some feature matches outliers (*e.g.*, symmetric textures) are considered as inliers in their SfM processes, then many cameras use these points for the camera poses estimation. With the accumulation of error, the drifting problem became more and more severe. While instead of using feature matches, we prefer to add cameras by using a subset of tracks with a larger consistency. The error accumulation problem is alleviated by our batched camera registration, and our tracks selection makes a further step to select more track inliers into BA.

Table 2. Camera calibration accuracy on the unordered image data. The median and mean position errors in meters are denoted by $\tilde{x}$ and $\bar{x}$, respectively. The number of cameras [41] is $N_i$, and the number of calibrated cameras is $N_c$. The time-cost in seconds is $T$. The number of original tracks is $N_t$, and the number of selected tracks in the last iteration is $N_s$.

| Dataset | | | Theia [38] | | | | COLMAP [34] | | | | Our BSfM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | $N_i$ | $N_t$ | $N_c$ | $\tilde{x}$ | $\bar{x}$ | $T$ | $N_c$ | $\tilde{x}$ | $\bar{x}$ | $T$ | $N_c$ | $\tilde{x}$ | $\bar{x}$ | $T$ | $N_s$ |
| Piccadilly | 2508 | 624K | 1824 | 0.6 | **1.1** | 3698 | 2132 | **0.3** | 1.3 | 105601 | **2176** | **0.3** | 1.2 | **3554** | 76K (12.9%) |
| Trafalgar | 5433 | 1158K | 3873 | 2.6 | **4.0** | 10210 | 4760 | **1.4** | 4.5 | 144840 | **4886** | **1.4** | 4.4 | **9579** | 121K (10.5%) |



(a) without tracks selection (b) without batched registration

Figure 10. Reconstruction results on the dataset Temple.

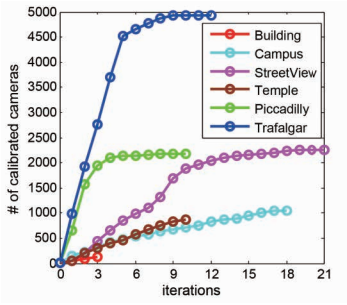



Figure 11. The number change of calibrated cameras in the outer loop.

To verify the importance of our tracks selection and batched camera registration, we reconstruct the dataset Temple in two other situations: (1) without selecting tracks; (2) without batched camera registration. From Fig. 10, we can see both reconstructions fail, indicating that these two modules are both important to tackle the SfM task on scene with symmetric textures. For the SfM efficiency, the time-cost of Bundler is 23.8 hours, while ours is only 0.4 hours. The reconstruction result on another ambiguous dataset Gendarmenmarkt [41] is shown in Fig. 9$(b)$.

The quantitative results comparison on the Piccadilly and Trafalgar is shown in Table 2. Note that for these public internet datsets, 1DSfM [41] uses the calibration results of the state-of-the-art incremental SfM system Bundler [35] as the reference ground-truth, thus we do not show the result of Bundler. In addition, since VSFM [42] does not support that using ceres-solver [2] for the bundle adjustment, for the fairness, we only show the comparison with Theia and COLMAP. From this table, we can see that our method calibrates the most number of cameras in both datasets, while achieves a similar accuracy, indicating that our method is more robust than the others. Especially for the dataset Trafalgar, our method calibrates 1000+ cameras more than that of Theia. Comparing the SfM efficiency, our BSfM is about 30 times faster than COLMAP in dataset Piccadilly, while 15 times faster in dataset Trafalgar.

For both two datasets, the tracks used in our camera registration is only about 10% of the full tracks, namely about 90% memory footprint is saved in BA. The change of the calibrated cameras number is shown in Fig. 11. At the beginning of the camera registration, we have already reconstructed the camera seeds, hence the number of calibrated camera for all datasets is 2. Since the selected camera seeds have many neighbors, we can see the number of calibrated cameras significantly increases after the first iteration, especially for those internet image datasets, most cameras usually captured the same buildings. For example, more than 900 images are calibrated in the first iteration of Trafalgar, and with the iterations going on, we can see the number of calibrated cameras increases fast, and more than 4500 images are calibrated by only 6 iterations.

Overall, for these unordered image datasets, our method performs more robust, efficient and scalable than many state-of-the-art SfM methods. Fig. 9 shows four other large-scale reconstruction results, where the full triangulation step is performed for better visualization. More reconstruction results are shown in our supplementary material.

## 5. Conclusion

In this paper, a batched SfM algorithm is proposed to make a further step towards a robust, accurate and efficient incremental system. With batched camera registration and tracks selection, our method outperforms many of the state-of-the-art SfM methods in terms of robustness and efficiency. Especially for the tracks selection module, it makes a large improvement on the saving of memory footprint. Extensive experiments demonstrate the efficiency, scalability and versatility of our batched SfM system.

## Acknowledgement

This work was supported by the Natural Science Foundation of China under Grants 61333015, 61421004 and 61632003.

## References


[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 5, 8

[3] C. Aholt, S. Agarwal, and R. Thomas. A qcqp approach to triangulation. *ECCV*, pages 654–667, 2012. 2

[4] C. Beder and R. Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Joint Pattern Recognition Symposium*, pages 657–666. Springer, 2006. 2

[5] D. C. Brown. *A solution to the general problem of multiple station analytical stereotriangulation*. D. Brown Associates, Incorporated, 1958. 3

[6] M. Byröd and K. Åström. Conjugate gradient bundle adjustment. *ECCV*, pages 114–127, 2010. 3

[7] A. Chatterjee and V. Govindu. Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 1, 3

[8] S. Choudhary, S. Gupta, and P. Narayanan. Practical time bundle adjustment for 3d reconstruction on the gpu. In *Trends and Topics in Computer Vision*, pages 423–435. Springer, 2012. 2

[9] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12):2841–2853, 2013. 1, 2, 5, 6

[10] H. Cui, S. Shen, and Z. Hu. Robust global translation averaging with feature tracks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3727–3732. IEEE, 2016. 1

[11] H. Cui, S. Shen, and Z. Hu. Global fusion of generalized camera model for efficient large-scale structure from motion. *Science China Information Sciences*, 60(3):038101, 2017. 1

[12] H. Cui, S. Shen, Z. Hu, et al. Efficient large-scale structure from motion by fusing auxiliary imaging information. *IEEE Transactions on Image Processing (TIP)*, 22:3561–3573, 2015. 1, 6, 7

[13] Z. Cui, N. Jiang, C. Tang, and P. Tan. Linear global translation estimation with feature tracks. In *BMVC*, 2015. 1

[14] Z. Cui and P. Tan. Global structure-from-motion by similarity averaging. In *ICCV*, pages 864–872. IEEE, 2015. 1, 3, 5, 6

[15] E. Dunn and J.-M. Frahm. Next best view planning for active model improvement. In *BMVC*, pages 1–11, 2009. 2, 4

[16] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(8):930–943, 2003. 2, 4

[17] S. Haner and A. Heyden. Covariance propagation and next best view planning for 3d reconstruction. In *ECCV*, pages 545–556. Springer, 2012. 2, 4

[18] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *CVPR*, pages 3041–3048. IEEE, 2011. 1

[19] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision (IJCV)*, 103:267–305, 2013. 1, 3, 6

[20] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 4, 5

[21] R. I. Hartley. Chirality. *International Journal of Computer Vision (IJCV)*, 26(1):41–61, 1998. 4

[22] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding (CVIU)*, 68(2):146–157, 1997. 2

[23] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*, pages 3287–3295. IEEE, 2015. 1

[24] F. V. K. Cornelis and L. V. Gool. Drift detection and removal for sequential structure from motion algorithms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1249–1259, 2004. 1

[25] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR*, pages 2969–2976. IEEE, 2011. 2

[26] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision (IJCV)*, 81(2):155–166, 2009. 2

[27] H. Li. A practical algorithm for L triangulation with outliers. In *CVPR*, pages 1–8. IEEE, 2007. 2

[28] F. Lu and R. Hartley. A fast optimal algorithm for l 2 triangulation. *ACCV*, pages 279–288, 2007. 2

[29] A. Lulli, E. Carlini, P. Dazzi, C. Lucchese, and L. Ricci. Fast connected components computation in large graphs by vertex pruning. *IEEE Transactions on Parallel and Distributed systems*, 28(3):760–773, 2017. 2

[30] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, pages 1–8. IEEE, 2007. 1

[31] P. Moulon and P. Monasse. Unordered feature tracking made fast and easy. In *CVMP*, page 1, 2012. 2, 3

[32] P. Moulon, P. Monasse, and R. Marlet. Adaptive structure from motion with a contrario model estimation. In *ACCV*, pages 257–270. Springer, 2013. 1

[33] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In *CVPR*, pages 2674–2683. IEEE, 2015. 1

[34] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113. IEEE, 2016. 1, 2, 4, 6, 7, 8

[35] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Internation Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008. 1, 2, 6, 7, 8

[36] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, pages 1–8. IEEE, 2008. 6

[37] L. Svärm, Z. Simayijiang, O. Enqvist, and C. Olsson. Point track creation in unordered image collections using gomory-hu trees. In *ICPR*, pages 2116–2119. IEEE, 2012. 2

[38] C. Sweeney. Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org. 1, 2, 6, 8

[39] C. Sweeney, T. Sattler, T. Hollerer, M. Turk, and M. Pollefeys. Optimizing the viewing graph for structure-from-motion. In *ICCV*, pages 801–809. IEEE, 2015. 1, 3

[40] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000. 2

[41] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *ECCV*, pages 61–75. Springer, 2014. 3, 5, 7, 8

[42] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134. IEEE, 2013. 1, 2, 3, 6, 8

[43] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, pages 3057–3064. IEEE, 2011. 3

[44] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, pages 1426–1433, 2010. 5, 6