Fusion of Auxiliary Imaging Information for Robust, Scalable and Fast 3D Reconstruction

Hainan Cui, Shuhan Shen, Wei Gao, Zhanyi Hu
{hncui, shshen, wgao, huzy}@ nlpr.ia.ac.cn

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences

Abstract. One of the potentially effective means for 3D reconstruction is to reconstruct the scene in a global manner, rather than incrementally, by fully exploiting available auxiliary information on imaging condition, such as camera location by GPS, orientation by IMU(or Compass), focal length from EXIF etc. However these auxiliary information, though informative and valuable, is usually too noisy to be directly usable. In this paper, we present a global method by taking advantage of such noisy auxiliary information to improve SfM solving. More specifically, we introduce two effective iterative optimization algorithms directly initiated with such noisy auxiliary information. One is a robust iterative rotation estimation algorithm to deal with contaminated EG(epipolar graph), the other is a robust iterative scene reconstruction algorithm to deal with noisy GPS data for camera centers initialization. We found that by exclusively focusing on the inliers estimated at the current iteration, called potential inliers in this work, the optimization process initialized by such noisy auxiliary information could converge well and efficiently. Our proposed method is evaluated on real images captured by UAV(unmanned aerial vehicle), StreetView car and conventional digital cameras. Extensive experimental results show that our method performs similarly or better than many of the state-of-art reconstruction approaches, in terms of reconstruction accuracy and scene completeness, but more efficient and scalable for large-scale image datasets.

1 Introduction

With the progress of modern technology, many imaging devices come with built-in sensors, such as GPS, compass and inclinometer. In addition, UAV (unmanned aerial vehicle), which is usually equipped with GPS and IMU (inertial measurement unit), has become widely available to generate high resolution DSM (digital surface model). Fortunately, sensor data are recorded simultaneously during image acquisition phase and from which approximate camera poses, though too noisy to be directly useful for 3D reconstruction [1, 2], can be obtained.

SfM approaches have been widely used to build 3D scene from images in the past few years. The state-of-art IBA(incremental bundle adjustment) approaches [3–5] start by selecting a few seed images for initial reconstruction, then repeatedly add new images to incrementally reconstruct the scene and refine the result by bundle adjustment. Although such an incremental mode finds its success in a variety of applications, it suffers from drift, large error accumulation, and heavy computational load. Contrary to IBA, many global algorithms [1, 2, 7–10] which simultaneously operate on all images are reported recently, in which the bundle adjustment, a time consuming module, is activated once rather than repeatedly. However, sometimes such global methods do not work well because the estimated parameters are not accurate enough for the bundle adjustment.

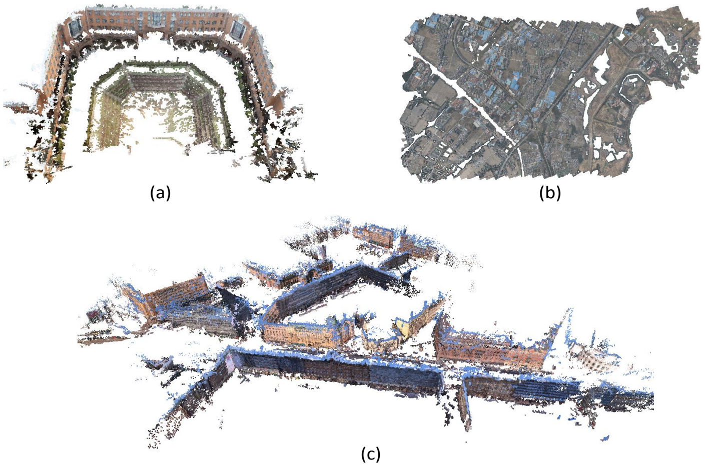

Fig. 1. 3D reconstruction results: (a) reconstruction by a conventional moving digital camera (MP; 144 images); (b) reconstruction by UAV (TK2; 501 images); (c) reconstruction by StreetView car (SV1; 2468 images). In order to better reflect the scene structure, here the results are further reconstructed by dense reconstruction method PMVS2 [6], which is a follow-up step of our method.

In this paper, we present a novel global strategy to solve SfM problem by fully exploiting available noisy auxiliary imaging information, such as GPS, IMU info, and compass angle. One key advantage of our method is its versatility, applicable to both ordered images (Fig. 1*b* and Fig. 1*c*) and unordered images (Fig. 1*a*). Another advantage of our method is its computational efficiency, and it works well for large scene reconstruction as shown in our experiments. For example, our SV2 image dataset contains 16600 images. Our proposed method has three steps. The first one is to build an EG (epipolar graph). The second one is a robust iterative rotation estimation. Since even under RANSAC paradigm, pairwise geometry estimates may still contain gross errors, global camera rotations are iteratively estimated by rotation consistency in this step. The last step is to iteratively perform triangulation and bundle adjustment. In order to tackle the problem of gross errors in pairwise geometry estimates as well as the inaccuracy of initializing camera centers with noisy GPS data, we introduce a concept called "potential inlier" for the iterative optimization process, which constitutes one of our major novelties.

We think although auxiliary imaging information is not accurate enough, it still contains some degree of truthfulness on the imaging condition, and can be used as a good initializer for our potential inliers selection. In our work, a constraint is considered as a *potential inlier* if its residual at the current iteration is less than an adaptive threshold. Note that by such a setting, a potential inlier is not necessarily meant a real inlier, it is merely meant that the probability of a potential inlier to be a real inlier is much larger than a potential outlier. In addition, potential inlier is meaningful only at the current iteration this is because a potential inlier is changeable at its status from iteration to iteration. It is possible a potential inlier at the current iteration changes to a potential outlier at the next iteration, and vice versus. But with a good initialization of potential inliers with auxiliary imaging information and iteratively filtering out the potential outliers, our proposed iterative method can rapidly converge with a few iterations, as demonstrated in our later experiments. To some degree, our proposed iterative method possesses some analogy with the well-known Boosting scheme. In Boosting, by iteratively combining weak classifiers, a strong classifier is obtained. In our method, by iteratively filtering potential outliers, potential inliers converge to real inliers, and the parameters, such as camera poses and 3D scene points, become more and more precise. Unlike Boosting where the convergence is slow due to the less impact of later weak classifiers, our method is quite computationally efficient as demonstrated in our later experiments. This computational efficiency is mainly due to the following two interleaved factors: Firstly, only potential inliers are used, which is a subset of the total constraints. Second, with iteration going on, the set of the selected potential inliers contains less and less real outliers, and the estimated parameters become closer and closer to the correct ones, then less number of iterations is needed.

Our proposed method is validated on various datasets, including images captured by UAV, StreetView car and a moving conventional digital camera. The reconstruction results are compared with those by state–of–art methods, such as Bundler [3], MRF-based [1], VSFM [8], OpenMVG [9] and Linear Method [10].

2 Related Work

Many reported approaches [3–5, 11] to solve SfM problem are based on incremental mode which repeatedly uses bundle adjustment to refine the scene and camera poses. The state-of-art representative is Bundler [3], which may suffer from drift due to the accumulation of errors in addition to its heavy computational load when handling large image dataset. Besides, Bundler's reconstruction result largely depends on the selection rule of the seed images and the order of subsequent image addition. Haner et al. [11] presented a new selection and addition rule which makes use of covariance propagation, and they pointed out that a well-determined camera should have both small estimated covariance and low reprojection error for next view planning. For Bundler, the worst-case running time of image matching part and bundle adjustment part is $O(n^2)$ and $O(n^4)$ in the number of images respectively, which becomes prohibitive when the number of images is large, many attempts are proposed to tackle this problem recently.

For the image matching part, graph-based algorithm [12, 13] are proposed to improve the efficiency by pruning original image set. However, the graph construction is always time consuming, and sometimes the completeness of scene cannot be guaranteed. The other typical solution is to employ image retrieval method to explore candidate matching image pairs [14, 15]. Nister et al. [14] proposed a vocabulary tree based approach to find out potential matching image pairs. Besides, based on the rank of Hamming distance, Cheng et al. [16] proposed a Cascade Hashing strategy to speed up the image matching. For bundle adjustment part, global methods [1, 2, 7–10], which only optimize the reconstruction result once, are considered of great potentiality. These approaches usually take three steps to solve the SfM problem. The first step is to compute camera rotations by rotation consistency, the second is to calculate camera translations, and the third one is to refine camera poses and 3D points by performing a final bundle adjustment. In particular, Jiang et al. [10] proposed a linear method for global camera pose registration from pairwise relative poses. This method requires a large set of precise pairwise geometries to perform the SVD decomposition. However, for many real applications, for example for StreetView images, pairwise geometry estimates are always noisy. As a result, many images may be discarded by [10] because their weak visual connections with other images.

Other works fuse auxiliary imaging information during the SfM solving [17, 18]. Carceroni et al. [17] computed camera rotations by using GPS. Pollefeys et al. [18] reported a real-time SfM in urban scene reconstruction with the support of GPS/IMU sensors. However, these two methods rely on high-precision GPS sensors which are not available in common devices. Several methods [1, 19] are proposed to reconstruct 3D scene by exploiting noisy auxiliary imaging information. Crandall et al. [1] proposed a discrete-continuous optimization method, in which noisy auxiliary info (GPS and vertical vanishing point) is incorporated into the SfM process. Note that VPs (Vertical vanishing points) are used to estimate the tilt angle. They used BP (belief propagation) on a discretized space of camera orientations and 2D camera positions to find a good parameter initialization, then run non-linear least squares and bundle adjustment to refine these estimates. Sinha et al. [19] also proposed a linear SfM method in which vanishing points are incorporated. However, these two methods are not applicable to the SfM problem on UAV images because the VPs cannot be estimated when the UAV faces a large tract of land where evident lines is not available. Although the tilt angle is available in IMU, it is usually unusable because of the influence of gravity. Besides, the extent of scene should be predetermined in [1], and discrete position labels take up a huge storage when the scene covers a large area.

In this paper, we present an efficient and versatile global approach, which is fully exploiting noisy auxiliary imaging information, to improve the SfM solving. Our proposed method is applicable to various kinds of images, including common digital images, UAV images and StreetView images.

3 A global approach by iteratively optimizing potential inliers

Our SfM method, shown in Fig. 2, consists of three main steps. Step1 is a preprocessing step, its main aim is to build an EG (epipolar graph). In this step, an image retrieval technique is used to speed up the image matching. In Step2, global camera rotations are iteratively estimated through rotation consistency. At each iteration, in order to increase the percentage of real edge inliers, gross edge outliers are filtered out. In Step3, camera poses and 3D scene points are iteratively estimated. In this step, we focus on tackling inevitable track outliers and the resulting inaccuracy problem by initializing camera centers with noisy GPS data. Next we elaborate on these three steps.

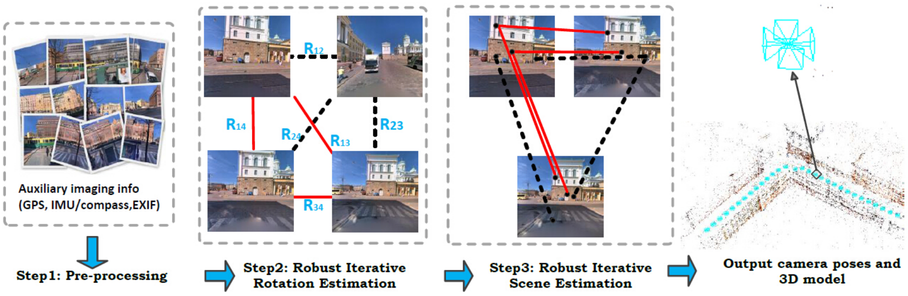
Auxiliary imaging info
(GPS, IMU/compass,EXIF)
Step1: Pre-processing
R12
R14
R24
R13
R23
R34
Step2: Robust Iterative Rotation Estimation
Step3: Robust Iterative Scene Estimation
Output camera poses and 3D model

Fig. 2. The flowchart of our method. Step1: features are detected and then matched across images. Step2: rotation estimation by iteratively optimizing potential pairwise geometry inliers(showed by red solid lines) and discarding gross pairwise geometries (showed by black dotted lines). Step3: scene estimation by iteratively optimizing potential track inliers(showed by red solid lines) and discarding gross tracks (showed by black dotted lines). Finally, camera poses showed by cyan cones and 3D scene points are obtained.

3.1 Step 1: Pre-processing

At first, SIFT points are extracted from images. Note that raw GPS data is in the form of longitude, latitude and altitude defined in the WGS84 coordinate system. For the convenience of further processing, these data are converted into the ECEF (Earth Centered, Earth Fixed) coordinate system, which is usually called the local east-north-up. Here ECEF is used as the global coordinate system.

In order to accelerate the matching process, a vocabulary tree [14] is used to detect candidate matching image pairs. Furthermore, based on GPS, too distant image pairs are discarded. For each candidate pair, we compute SIFT matches using Cascade Hashing strategy [16]. Each 3D scene point is identified by finding their corresponding track−interest points across multiple images which have similar SIFT descriptors. However, sometimes a feature point may be contained by different tracks. Since such tracks are ambiguous when used for subsequent triangulation and bundle adjustment, they are considered unstable and discarded. After matching relevant images, geometric verification based on 5-point algorithm [20] is performed. Two images are considered as a matched pair if the number of their matched SIFT points is more than a threshold (in our work, it is set to 20). Moreover, pairwise relative rotations and translation directions are computed from every matched pair of images.

The final matching result is represented by a graph called EG (epipolar graph), whose vertices $V = \{I_1, I_2 \cdots I_N\}$ correspond to images and edges $E = \{e_{ij} | i, j \in V\}$ link matched image pairs, then the LCC (largest connected component) of EG is extracted and used in the subsequent reconstruction.

3.2 Step 2: Robust Iterative Rotation Estimation

Coarse initial camera rotations defined under the ECEF coordinate system can be easily obtained from camera orientations. For UAV, the orientation is obtained by noisy IMU. For conventional digital camera equipped with compass, the orientation is initialized by compass and tilt angle(VP is calculated by the method [21]). For StreetView car which only equipped with a GPS sensor, the method proposed by Crandall [1] is used to get a rough orientation.

Given a pairwise relative pose estimate (R_{pq}, t_{pq}) between cameras p and q, the problem of rotation estimation can be formulated as a search for the absolute orthonormal rotations R_p, R_q, such that the following constraint is satisfied:

$$R_{pq} = R_p R_q^T \tag{1}$$

Every edge in EG forms such a constraint. Thus, an overdetermined equation system is obtained since EG always consists of redundant edges. Note that the residual of an edge between cameras p and q is measured by the Frobenius norm of $\|R_{pq} - R_p R_q^T\|$. As proposed by Martinec [22], the solution of this overdetermined equation system can be initially computed without considering the orthonormality constraint and then enforced by subsequently projecting the approximate rotation to the closest rotation under Frobenius norm using SVD decomposition. However there always exist outliers, whose relative pose estimates are either incorrect or the epipolar constraints are actually non-existent, in EG.

In order to tackle the inevitable edge outliers in EG and increase the percentage of real edge inliers in the optimization process, we propose a robust rotation estimation algorithm by iteratively and exclusively optimizing the so-called potential edge inliers. An edge in EG is regarded as a potential edge inlier in the i^{th} iteration if its corresponding residual ($\|R_{pq} - R_p R_q^T\|_F$) is less than a threshold $T^{(i)}$. Given a threshold α, $T^{(i)}$ in the i^{th} iteration is computed as follows:

$$T^{(i)} = \min\{T : \frac{\sum_{j=1}^{M} \eta_j^{(i)}}{M} \geq \alpha\} \tag{2}$$

$$s.t. \qquad \eta_j^{(i)} = \begin{cases} 0, & \text{if } r_j^{(i)} > T; \\ 1, & \text{if } r_j^{(i)} \leq T; \end{cases} \tag{3}$$

where $r_j^{(i)}$ is the residual of the j^{th} edge in the i^{th} iteration; $j = 1...M$; M is the number of edges in the LCC of EG. Moreover, the following covering condition should be satisfied: the current potential edge inliers should cover all the vertices in the LCC of EG. If this condition is not satisfied, the threshold α in Eq. (2) should be increased. In our work, initial α is set to 0.9. With this threshold, the goal of Eq. (2) is to calculate a minimal threshold $T^{(i)}$ such that the percentage of potential edge inliers over the total edges is equal or larger than 90%. By ordering edge residuals from small to large, we consider that the last 10% of EGs are erroneous or potential outliers, then discarded in the current iteration. Discarding such EGs will increase the percentage of real edge inliers over used EGs in the optimization.

Note that some real edge inliers may be labelled as potential edge outliers due to the inaccurate camera rotations as well as the empirical threshold α. In order to tackle such inaccuracy problem and make more real edge inliers be used in the optimization, we estimate the absolute camera rotations $\mathbf{R} = \{R_1, ..., R_N\}$ iteratively by minimizing the sum of the residuals of the potential edge inliers, where N is the number of images. In the i^{th} iteration:

$$\mathbf{R}^{(i+1)} = \min\{\mathbf{R} : \sum_{p=1}^{N}\sum_{q=1}^{N} E_{pq}^{(i)} \| R_{pq} - R_p^{(i)} R_q^{(i)T} \|_F\} \tag{4}$$

subject to that each matrix in $\mathbf{R}$ is orthonormal. $E_{pq}^{(i)}$ is set to 1 if the edge between image p and image q is a potential edge inlier in the i^{th} iteration, otherwise set to 0. With the camera rotations become more and more accurate with iteration, more and more real edge inliers will be included in the optimization process. For the sake of efficiency, the iteration is usually stopped when the number of the changes of the potential edge inliers between two consecutive iterations is less than a threshold (in our work, it is set to 20).

3.3 Step 3: Robust Iterative Scene Reconstruction

The camera projection matrix set $\mathbf{P} = \{P_i; i = 1...N\}$, can be approximately initialized as:

$$P_i = K_i R_i [\mathbf{I} \; -C_i] = \begin{bmatrix} f_{exif\_i} & 0 & 0 \\ 0 & f_{exif\_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} R_i [\mathbf{I} \; -C_i] \tag{5}$$

where f_{exif_i} denotes the focal length from the i^{th} image EXIF tag; R_i denotes the estimated absolute rotation of image i in Step 2; $\mathbf{I}$ denotes the identity matrix; C_i denotes the converted GPS of image i. Given the camera projection matrices and a track set of corresponding images, 3D scene points can be initially reconstructed by triangulation and bundle adjustment. However, due to the inaccuracy of the current initialization, mostly one-time bundle adjustment is not sufficient to produce satisfactory reconstruction result, and additional alternated triangulation and bundle adjustment process need to be carried out.

For each track, we pick the image pair which has the maximal baseline among all possible visible image pairs to perform the triangulation. For the robustness concern, a 3D point will not be triangulated if the maximal baseline of its corresponding track is too small, and a 3D point is saved as a candidate for further processing when its current average reprojection error across all visible images is less than 20 pixels and maximal reprojection error across all visible images is less than 100 pixels.

Given the camera projection matrix set $\mathbf{P}$ and the set of currently reliable reconstructed 3D points $\mathbf{X}$, the discrepancy between the measured 2D image point locations and predicted 3D scene points is minimized subsequently. For N images and K tracks, the cost function $\mathcal{G}$ is formulated as the weighted geometric projection errors:

$$\mathcal{G}\left(\mathbf{P},\mathbf{X}\right)=\sum_{i=1}^{N}\sum_{j=1}^{K}v_{ij}\|x_{ij}-\gamma(P_i,X_j)\|^2 \tag{6}$$

where 2D image point locations x_{ij} are the observation of the 3D point X_j in the i^{th} image; v_{ij} is set to 1 if X_j is visible in the i^{th} image, otherwise set to 0. $\gamma(P_i,X_j)$ denotes the projection of X_j in the i^{th} image. Note that in our work only the first two camera radial distortion parameters are used. The nonlinear least square problem defined in Eq. (6) always needs a good parameter initialization. However, converted GPS locations are not precise enough to be used as the camera positions initialization. Besides, there always exist outliers, which are caused by mismatching, in tracks set. Thus, direct optimization on Eq. (6) is not a sensible choice, and an iterative approach is here proposed by only performing optimization on potential track inliers to tackle this problem.

A track is regarded as a potential track inlier in the l^{th} iteration if its average reprojection error across visible images is less than $H^{(l)}$. Given a threshold β, $H^{(l)}$ in the l^{th} iteration is calculated as :

$$H^{(l)}=\min\{H:\frac{\sum_{j=1}^{K}\delta_j^{(l)}}{K}\geq\beta\} \tag{7}$$

$$s.t. \qquad \delta_j^{(l)}=\begin{cases}0, & \text{if } r_j^{(l)}>H;\\ 1, & \text{if } r_j^{(l)}\leq H;\end{cases} \tag{8}$$

where $r_j^{(l)}$ denotes the averaged reprojection error across all visible images of the j^{th} track in the l^{th} iteration; $j=1...K$; K denotes the number of tracks. In addition, these potential track inliers should also satisfy the following covering condition: the visible images of the current potential track inliers should cover all vertices in the LCC of EG. If this condition is not satisfied, the potential track inliers should be recomputed by increasing β. Since there are still outliers present in the obtained potential tracks inliers, we use a robust Huber norm by setting its parameter as 25 pixels on the reprojection error. In our work, β is set to 0.9. Similarly as that in Section 3.2, by ordering average reprojection errors from small to large, the last 10% of the tracks are considered as potential outliers, and they are not used in the optimization of current iteration.

Considering that the focal lengths obtained from image EXIF tags are relatively reliable, an enforcement term is added to the cost function (Eq. (6)). As a result, at the l^{th} iteration, our cost function on potential track inliers is formulated as:

$$\mathcal{F}\left(\mathbf{P}^{(l)}, \mathbf{X}^{(l)}\right) = \mathcal{G}_{huber}\left(\mathbf{P}^{(l)}, \mathbf{X}^{(l)}\right) e_j^{(l)} + \sum_{i=1}^{N} \lambda \left(f_i^{(l)} - f_{exif\_i}\right)^2 \qquad (9)$$

where $f_i^{(l)}$ is the focal length of the i^{th} image in the l^{th} iteration; $e_j^{(l)}$ is set to 1 if the j^{th} track is considered as a potential track inlier in the l^{th} iteration, otherwise set to 0. Conventionally, repeated bundle adjustment is regarded as the most time-consuming part in 3D reconstruction. However, as our following experimental part shows, the time-cost of repeated bundle adjustment in this step is acceptable. The reason is two-fold: on the one hand, only a part of tracks are optimized in each iteration, and the iteration number is always less than 5; on the other hand, the sparse structure of SfM problem is taken into account. In our work, the weighting factor λ in Eq. (9) is set to 10^{-4}, and the version(1.8.0) of ceres-solver [23] is adopted to perform the bundle adjustment.

4 Experiments

The experiments are carried out on a PC with an Intel Core2 i5-2400 3.10GHz CPU(4 cores) and 16G RAM. Our method is evaluated on real images captured by different devices, including (1) an UAV with integrated GPS and IMU sensors; (2) a conventional digital camera with a GPS receiver and compass inside; (3) a StreetView car equipped with a GPS sensor. The specifications of five image datasets are listed in Table 1. Due to the limited space, only the first 4 datasets are compared in detail.

4.1 Comparison Methods and Comparison Criteria

Our method is compared with Bundler [3], MRF-based method [1], OpenMVG [9], VSFM [8] and the Linear Method [10]. Note that since OpenMVG in [24] requires images to have the same initial focal length, OpenMVG cannot be run on MP. In addition, MRF-based approach [1] stresses the importance of tilt. Due to the lack of straight lines in UAV images, MRF-based method cannot be performed on TK1 and TK2.

Both qualitative and quantitative comparisons are carried out. In the qualitative comparison, not only the scene structures are assessed, but the camera trajectories are also compared for the UAV and StreetView images. Gross calibration errors or evident artifacts are the direct indicators of the algorithm′s inadequacy. In the quantitative comparison, we evaluate the accuracy of the reconstructed cameras by comparing their positions to the ground truth locations. For the Arts Quad dataset, the truth GPS locations are publicly available in [1]. The running-time of the evaluated algorithms after image matching part is recorded to compare the computational load.

Table 1. Specifications of image datasets

Name	# of images in LCC	Capturing device	GPS precision	IMU/Compass precision	same initial focal length?
MP	144	Canon 5D Mark III	$5 \sim 10m$	$5 \sim 10°$	no
TK1	145	UAV	$5 \sim 10m$	$5 \sim 10°$	yes
TK2	501	UAV	$5 \sim 10m$	$5 \sim 10°$	yes
SV1	2468	StreetView Car	$3 \sim 5m$	–	yes
SV2	16600	StreetView Car	$3 \sim 5m$	–	yes

4.2 Results and Analysis

Results of Step 3. Since Step3 described in Section 3.3 is the key step in our algorithm, we show its results in Fig. 3 and Fig. 4. It can be clearly seen from Fig. 3 that our method almost converges after four iterations. Since initial parameters are not good enough, only a subset of tracks are regarded as potential track inliers at the first iteration. With iterations going on, more potential track inliers appear in the subsequent iterations, which indicates that the camera poses become more and more precise. Some results with respect to the iteration time are shown in Fig. 4. From the results in the first iteration, it can be seen that one-off bundle adjustment is obviously not enough when camera centers are directly initialized with GPS. Specifically, in the result of MP(Iter_1), both camera positions and scene structure are bad and unreasonable. With the iterations going on, the scene structure becomes more and more precise and reasonable.

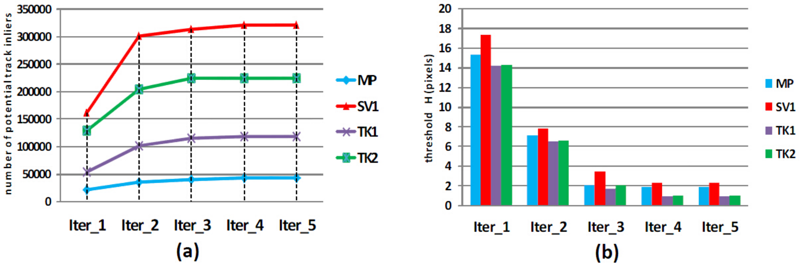

Fig. 3. (a) Number of potential track inliers with respect to the iteration number; (b) the threshold $H^{(l)}$ with respect to the iteration number.

Moreover, tracks are always clean in UAV images as no occlusions exist in the view. However, tracks are always contaminated in images captured by free shooting or StreetView car because of the large changes of view angles or the existence of numerous self-symmetric features. In our experiment, relative to the respective whole tracks, the percentage of final potential track inliers on MP, TK1, TK2 and SV1 is 54.07%, 89.73%, 89.71%, 59.03%, which shows our proposed method is robust to both cluttered and clean scenes.

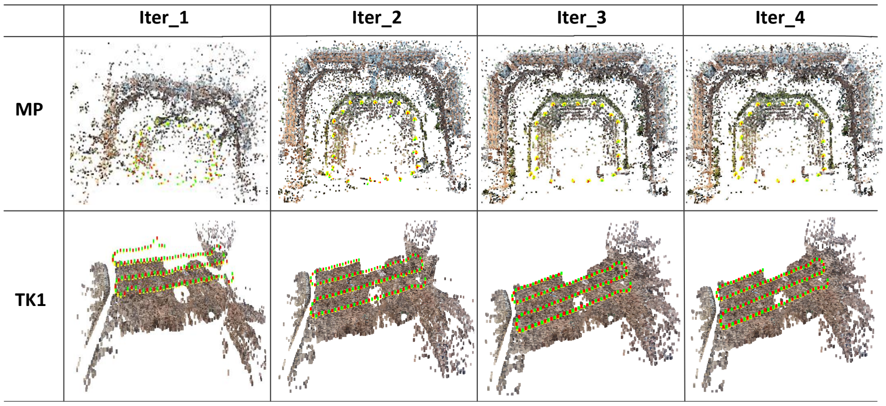

Fig. 4. Sparse reconstruction results with respect to the iteration number. Red and green points denote the camera positions.

Qualitative Comparison. For OpenMVG and Linear Method, accurate camera poses estimation are mainly dependent on the existence of many accurate triplets. However, triplets are not many in UAV and StreetView images because the speed of UAV or street view car is usually fast. Especially for StreetView images, many pairwise geometry estimates are usually not accurate enough. As shown in Fig. 5, these two methods generate obvious error results on TK1, TK2 and SV1. Note that the results on MP are comparable among five methods (Bundler, MRF-based approach, VSFM, Linear Method and our method), which indicates that most existing SfM methods are more suited for this scenario.

For TK1, the results produced by openMVG and Linear Method are obviously incomplete. For the camera trajectory of TK1, one obvious calibration error (a camera is under the scene), which is highlighted by a blue circle, appears in Bundler's result. Compared with VSFM's result on TK1, the camera trajectory of our result is more reasonable(unreasonable jitters appear in VSFM's result highlighted by a blue circle). In addition, for TK2, the result produced by openMVG is appearantly wrong. The reason is that OpenMVG does not account the image distortion. More elaborate reasons are reported by Wu [25]. Furthermore, results on TK2 produced by Bundler and Linear Method are obviously wrong, and there are some obvious calibration errors (sudden leap on camera centers) in VSFM's result, while our result on TK2 is more reasonable than others.

For SV1, the results produced by Bundler, openMVG and Linear Method are obviously incomplete or wrong. In order to make the comparison more evident, the scene structure and camera trajectory of other results are respectively shown in Fig. 6. For the scene structure, some obvious errors, which are highlighted by red circles, appear in the results produced by MRF-based approach and VSFM. For the camera trajectory, our result is more convincing as no obvious jitters

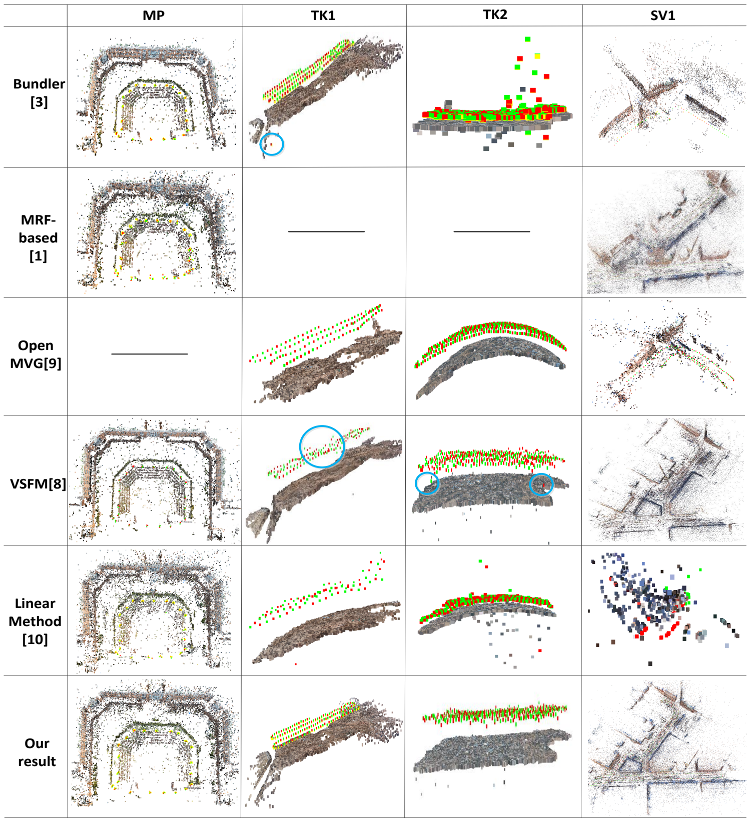

Fig. 5. Sparse reconstruction results on 4 image datasets. Red and green points denote the camera positions. Blue ellipses mark the sampled unreasonable areas in the results.

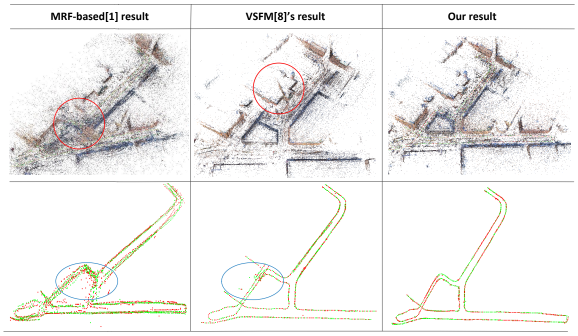

Fig. 6. The first row shows the reconstruction results on SV1 and the second shows the corresponding camera centers. Red circles and blue ellipses mark the unreasonable areas in the reconstructed results.

appear on the route of car. Two blue ellipses mark the unreasonable parts on the results of VSFM and MRF-based method.

The reason of why our results are better than those produced by MRF-based method is mainly due to the following two factors. Firstly, the 2D camera positions in MRF-based method may be not dense enough (in our experiments, a label corresponds a 4m*4m square). As a result, parameter initializations may be not good enough for the bundle adjustment. Secondly, the accuracy of initial translations in MRF-based method largely depend on the initial selected tracks, so it is sensitive to track outliers. In sum, in term of qualitative comparison, our method outperforms the other five ones. Furthermore, our reconstruction result on SV2 is shown in Fig. 7*b* where the area marked by red dotted line is the reconstruction on SV1. Since the other five methods could not work well on SV1, they are not run on SV2. From our results, many dense reconstruction methods can be used. As shown in Fig. 1, dense reconstruction is performed by PMVS2 [6].

Quantitative evaluation. The accuracy of the calibrated cameras is evaluated by comparing their positions with ground truth locations. For the dataset Arts Quad which is publicly available in [1], there are 6514 images in total while 4255 images have geotags, and 348 images with high accurate differential GPS positions are used as ground truth. Since we need GPS to initilize camera centers, our method is only performed on geotaged images. The reconstruction results generated by our method is shown in Fig. 7a, in which 251 out of the 348 ground truth images are found. Then RANSAC is used to estimate a 3D similarity transformation between the 251 camera locations and their ground truth coordinates. The registration result shows that our camera positions have a median error of 1.13 meter, which is comparable with 1.16 meter reported by [1].

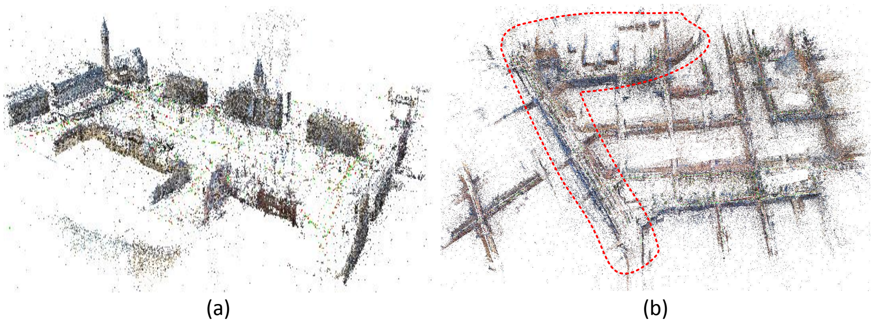

Fig. 7. Sampled reconstructions on Arts Quad (a) and SV2 (b). The area marked by red dotted line in (b) is the reconstruction results on SV1.

Time Efficiency and Scalability. The running-time of OpenMVG [9] is not compared here because its reconstruction results on our four datasets are either

Table 2. Running time of our method compared to other methods. (#) denotes the number of calibrated images by the corresponding method.

	Bundler	MRF-based	VSFM	Linear Method	Our method
MP	20.1 mins(144)	13.2 mins(144)	1.6 mins(144)	**1.1 mins**(144)	1.8 mins(144)
TK1	(142)	–	9.0 mins(145)	(79)	**7.5 mins**(145)
TK2	12.1hours(501)	–	(499)	(360)	**25.5 mins**(501)
SV1	(90)	31.2hours(2468)	(1910)	(179)	**5.0 hours**(2468)

Table 3. Time-cost comparison between MRF-based method and our method on SV1

	Rotations Estimation	Translations Estimation	Triangulation	Bundle Adjustment	Total time-cost
MRF-based [1]	9.0 mins	30.0 hours	4.0 mins	1.0 hours	31.2 hours
Our method	9.0 mins	0 mins	4.0 mins * 5	4.5 hours	**5.0 hours**

incomplete or obviously wrong. As a result, the time-cost of our method are compared with those of other four methods. Neither parallel computation nor GPU acceleration is used here to ensure the fairness of comparison. Note that if the cameras is partly calibrated, only the number of calibrated images of the corresponding method is showed in Table 2. It can be seen from Table 2 that our method performs better than other approaches on TK1, TK2 and SV1.

Our method is about 6 times faster than MRF-based approach on SV1. The detailed comparison of these two global reconstruction methods is shown in Table 3. Obviously, MRF-based approach spends a lot of time in estimating translations, while our main time-cost is spent on bundle adjustment. In the third row of Table 3, 4.0mins*5 is meant that each triangulation spends 4.0 mins and 5 iterations are carried out. The results show that our method has a better scalability than the MRF-based approach.

5 Conclusion

In this paper, we propose an efficient and accurate reconstruction method by fully exploiting auxiliary imaging information. The main novelty of our work is the exclusive use of the so-called potential inliers at each iterative optimization step to effectively deal with the inevitable constraint outliers, which is made possible in turn by employing auxiliary imaging information. Experimental results show that our approach outperforms the state-of-art reconstruction approaches, especially for UAV and StreetView images. In the future work, the iterative convergence of potential inliers to true inliers will be further investigated.

Acknowledgement. This work was supported by National High Technology $R\&D$ Program of China(863 program) under the grant No $2013AA12A202$ and NSFC under the grant No $(61333015, 61273280)$.

References

1. Crandall, D., Owens, A., Snavely, N., Huttenlocher: Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. PAMI **35**(2013) 2841–2853
2. Irschara, A., Hoppe, C., Bischof, H., Kluckner, S.: Efficient structure from motion with weak position and orientation priors. In: CVPRW, IEEE (2011) 21–28
3. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. IJCV **80** (2008) 189–210
4. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. ACM **54** (2011) 105–112
5. Moulon, P., Monasse, P., Marlet, R.: Adaptive structure from motion with a contrario model estimation. In: ACCV. Springer (2013) 257–270
6. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. PAMI **32** (2010) 1362–1376
7. Strecha, C., Pylvanainen, T., Fua, P.: Dynamic and scalable large scale image reconstruction. In: CVPR, IEEE (2010) 406–413
8. Wu, C.: Towards linear-time incremental structure from motion. In: International Conference on 3D Vision. (2013) 127–134
9. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motionsfor robust, accurate and scalable structure from motion. In: ICCV(2013)
10. Nianjuan Jiang, Zhaopeng Cui, P.T.: A global linear method for camera pose registration. In: ICCV(2013)
11. Haner, S., Heyden, A.: Covariance propagation and next best view planning for 3d reconstruction. In: ECCV(2012)
12. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR(2008)
13. Lou, Y., Snavely, N., Gehrke, J.: Matchminer: efficient spanning structure mining in large image collections. In: ECCV(2012)
14. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR(2006)
15. Chum, O., Mikulik, A., Perdoch, M., Matas, J.: Total recall ii: Query expansion revisited. In: CVPR(2011)
16. Jian Cheng, Cong Leng, J.W.H.C.H.L.: Fast and accurate image matching with cascade hashing for 3d reconstruction. In: CVPR(2014)
17. Carceroni, R., Kumar, A., Daniilidis, K.: Structure from motion with known camera positions. In: CVPR(2006)
18. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., et al.: Detailed real-time urban 3d reconstruction from video. IJCV **78** (2008) 143–167
19. Sinha, S.N., Steedly, D., Szeliski, R.: A multi-stage linear approach to structure from motion. In: Trends and Topics in Computer Vision. Springer (2012) 267–281
20. Nistér, D.: An efficient solution to the five-point relative pose problem. PAMI **26** (2004) 756–770
21. Tardif, J.P.: Non-iterative approach for fast and accurate vanishing point detection. In: ICCV(2009)
22. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR(2007)
23. Agarwal, S., Mierle, K., Others: Ceres Solver. (http://ceres-solver.org/)
24. Moulon, P.: openMVG. (https://github.com/openMVG/openMVG/tree/master)
25. Wu, C.: Critical configurations for radial distortion self-calibration. In: CVPR(2014)