IEEE Robotics and Automation Letters (RAL) paper presented at the
2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Iteratively Reweighted Midpoint Method for Fast Multiple View Triangulation

Yang Kui[1], Fang Wei[2], Zhao Yan[1], Deng Nianmao[1]

***Abstract*— The classic midpoint method for triangulation is extremely fast, but usually labelled as inaccurate. We investigate the cost function that the midpoint method tries to minimize, and the result shows that the midpoint method is prone to underestimate the accuracy of the measurement acquired relatively far from the 3D point. Accordingly, the cost function used in this work is enhanced by assigning a weight to each measurement, which is inversely proportional to the distance between the 3D point and the corresponding camera center. After analyzing the gradient of the modified cost function, we propose to do minimization by applying fixed-point iterations to find the roots of the gradient. Thus the proposed method is called the *iteratively reweighted midpoint* method. In addition, a theoretical study is presented to reveal that the proposed method is an approximation to the Newton's method near the optimal point, and hence inherits the quadratic convergence rate. At last, the comparisons of the experimental results on both synthetic and real datasets demonstrate that the proposed method is more efficient than the state-of-the-art while achieves the same level of accuracy.**

I. Introduction

Triangulation is the task of estimating the 3D coordinate of a physical point, given its observations in multiple camera views. And the task can be solved by finding the intersection of multiple known rays originating from each camera. In the absence of noises, this problem is trivial and can be dealt with many linear methods. In practice, however, those rays will not intersect in space due to various sources of noises. The goal becomes the search of the point that best fits the measurements.

Triangulation is a fundamental building block for many applications like *Structure from Motion (SfM)* and *Simultaneous Localization And Mapping (SLAM)*, on which there has been a large number of literatures during the past few years. For the specific case of two views, the most well known triangulation method may be the *midpoint* method [1]. Given two views of the 3D point, the midpoint method chooses the midpoint of the common perpendicular to the two rays as the best estimation. Although the midpoint method is straightforward, it has been known to be suboptimal and may result in a bad estimation [1]. The optimal closed form solutions for two-views triangulation have already been achieved by polynomial root finding in [2], and latter accelerated in [3]. These methods solve the triangulation problem by explicitly computing the complete set of the stationary points of the cost function. They have been extended to the case of three views in [4], where a polynomial of degree 47 must be solved. Nevertheless, they are not applicable to the more general cases with more than three views since the degree of the resulting polynomial grows quadratically with the number of views. Besides, there is another traditional method for solving multiple views triangulation called the *direct linear transform*, which constructs a set of linear equations for each observation, and then solves the linear equations by applying singular value decomposition. Due to the ignorance of the error minimized by the obtained solution, the results provided by the direct linear transform may be far from the optimal when the input is noisy [1].

Starting from an inaccurate initial point, the triangulation result can be refined by minimizing the ℓ_2 norm, i.e., the mean squares, of the reprojection errors using gradient descent methods [5]. Unfortunately, it is known that this cost function is not convex, and the gradient descent methods may get stuck at suboptimal local minima [6]. To overcome this drawback, researchers have proposed to utilize the ℓ_∞ norm, in other word the maximum, of the reprojection errors as the cost function [7]–[11]. The ℓ_∞ norm of the reprojection errors is quasiconvex and thus contains a single global minimum. Although these ℓ_∞ norm methods avoid the problem of local minima, their computational costs are considerable, since the convex programs have to be done at each iteration to determine the update direction [11]. In practice, as addressed in [6], it is scarce for ℓ_2 norm methods to be trapped at local minima, while the ℓ_2 norm methods are significantly less time-consuming than ℓ_∞ norm methods. Therefore, the ℓ_2 norm methods are better balanced between speed and accuracy, and they are popularly used in recent state-of-the-art SLAM systems [12]–[15]. Nevertheless, the ℓ_2 norm methods are still too slow for large-scale reconstruction problems or SLAM systems where there are a significant number of observations for each 3D point to be handled, and hence there is room for further improvement.

Although the basic principles in triangulation have already been extensively studied, efficient and accurate methods for large-scale problem are strongly needed. In this work, we re-examine this problem and propose an extended midpoint method for fast multiple view triangulation. The classic midpoint method is inaccurate in some cases because its linear cost function is inclined to underestimate the accuracies of the measurements acquired relatively far from the 3D points. In order to get rid of the bias in different measurements, we

*This work was supported by National Natural Science Foundation of China (No. 61803035).

[1] K. Yang, Y. Zhao, and N. Deng are with the School of Instrumentation Science and Opto-electronics Engineering of Beihang University, No.37 Xueyuan Road, Beijing, P.R.China, 100191. yang3kui@gmail.com

[2]W. Fang is with the School of Automation, Beijing University of Posts and Telecommunications, No.10 Xitucheng Road, Beijing, P.R.China, 100876.

propose to assign each term a weight that decreases with the corresponding distance. Since the distances between the 3D point and the cameras are unknown before the location of the 3D point is determined, this new cost function is not linear any more. And for this nonlinear cost function, the method of fixed point iterations is applied to find the root of the gradient which is needed by minimization. We thus name our method the *Iteratively Reweighted MidPoint (IRMP)* method. Further study reveals the connection between the IRMP method and minimizing the proposed cost function by Newton's method [5]. Experimental results validate that the proposed IRMP method can improve the accuracy of the classic midpoint method. When compared with the state-of-the-art, the IRMP method is several times faster without compromising the accuracy.

In section II, the theoretical aspect of the IRMP method will be introduced in detail. In section III, the experimental results on synthetic and real data will be presented and discussed. At last, the conclusions will be drawn in section IV.

II. Iterative Reweighted Midpoint Method

Throughout this work, matrices, vectors, and scalars are denoted by bold capital letters, bold lowercase letters and plain lowercase letters, respectively. Points in 3D space are represented using column vectors. $\mathbf{I}$ represents the identity matrix, and $(\cdot)^{\mathrm{T}}$ denotes the transpose of a matrix. And all the quantities are expressed in a uniform world coordinate system unless stated otherwise. Besides, we consider a calibrated centric camera model, where an image point can be represented by a unit column vector originated from the corresponding camera center.

A. Multiple View Midpoint Method

Given a 3D point $\mathbf{p}$ and a set of its unit measurement vectors $\{\mathbf{b}_i\}$ by cameras centered at $\{\mathbf{o}_i\}$, the triangulation problem is to find the best estimation of the 3D point $\mathbf{p}$. In the absence of noises, we should have $\mathbf{b}_i = (\mathbf{p}-\mathbf{o}_i)/\|\mathbf{p}-\mathbf{o}_i\|$, where $\|\cdot\|$ denotes the ℓ_2 norm, or equally speaking the length, of a vector. Meanwhile, the rays with directions $\{\mathbf{b}_i\}$ and original points $\{\mathbf{o}_i\}$ should intersect at $\mathbf{p}$. However, with noises in the measurements, these back-projected rays do not exactly intersect in space. In the specific case of two views, the midpoint of the common perpendicular to the two rays is chosen as the best estimation. When extended to the more general cases of multiple views [16], the midpoint method determines the optimal point by minimizing the sum of squares of the distances from this point to all the rays. In a mathematical way, the midpoint method minimizes

$$e(\mathbf{p}) = \sum\nolimits_{i=1}^{N} \left\| \left(\mathbf{I} - \mathbf{b}_i\mathbf{b}_i^{\mathrm{T}}\right)(\mathbf{p}-\mathbf{o}_i)\right\|^2. \tag{1}$$

Here, $\mathbf{b}_i\mathbf{b}_i^{\mathrm{T}}$ is the orthogonal projection matrix from the point $\mathbf{p}$ to the i-th ray. Let $\mathbf{B}_i = \mathbf{I}-\mathbf{b}_i\mathbf{b}_i^{\mathrm{T}}$. $\mathbf{B}_i$ always has the eigenvalues of $\{1,1,0\}$ indicating its positive semidefinite attribute. Besides, there are two useful relations, i.e. $\mathbf{B}_i^{\mathrm{T}} = \mathbf{B}_i$ and ${\mathbf{B}_i}^{\mathrm{T}}\mathbf{B}_i = \mathbf{B}_i$, which will be exploited in latter discussion. With $\mathbf{B}_i$, the cost function can be rewritten as

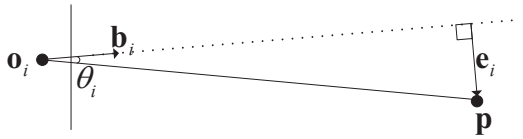


Fig. 1. The geometrical relationship between different variables used in this work. $\mathbf{o}_i$ is the center of the i-th camera. $\mathbf{b}_i$ is the unit measurement vector of the 3D point $\mathbf{p}$ in the i-th camera. $\mathbf{e}_i$ is the error vector from $\mathbf{p}$ to the measurement $\mathbf{b}_i$. Mathematically, $\mathbf{e}_i = \mathbf{B}_i(\mathbf{p}-\mathbf{o}_i) = (\mathbf{I}-\mathbf{b}_i\mathbf{b}_i^{\mathrm{T}})(\mathbf{p}-\mathbf{o}_i)$, and also $\mathbf{e}_i \perp \mathbf{b}_i$. θ_i is the error angle. Obviously, $\sin(\theta_i) = \|\mathbf{e}_i\|/\|\mathbf{p}-\mathbf{o}_i\|$.

$$e(\mathbf{p}) = \sum\nolimits_{i=1}^{N} \|\mathbf{B}_i(\mathbf{p}-\mathbf{o}_i)\|^2. \tag{2}$$

Although the formulation of the cost function used here is slightly different from the original one in [16], they are equivalent to each other in the sense of geometry. In [17], the authors explored the possibility of using $\ell_q (1 \le q < 2)$ norm to replace the ℓ_2 norm in the cost function. In this work, we concentrate on the ℓ_2 norm formulation, since it is the optimal choice under the Gaussian noise model. Now with $\{\mathbf{b}_i\}$ and $\{\mathbf{o}_i\}$ already known, the midpoint method regards solving a least square problem, which is equivalent to solving the linear equations

$$\left(\sum\nolimits_{i=1}^{N} \mathbf{B}_i\right)\mathbf{p} = \left(\sum\nolimits_{i=1}^{N} \mathbf{B}_i\mathbf{o}_i\right), \tag{3}$$

where $\sum_{i=1}^{N}\mathbf{B}_i$ is a 3×3 matrix and $\sum_{i=1}^{N}\mathbf{B}_i\mathbf{o}_i$ is a 3×1 vector. Only if not all the observation vectors are parallel, $\sum_{i=1}^{N}\mathbf{B}_i$ is full-rank and thus a unique solution can be obtained without much effort. To summarize, the midpoint method is extremely efficient, since it only requires to construct and solve a set of 3 linear equations by one time.

B. Unbiased Cost Function

The midpoint method is fast by sacrificing the accuracy. The errors in midpoint method are described in a Euclidean way while the camera projection constitutes a projective space. As shown in Figure 1, when $\mathbf{p}$ moves further away from the camera center $\mathbf{o}_i$ along the line $\mathbf{o}_i\mathbf{p}$, the error computed by (2), i.e., $\|\mathbf{B}_i(\mathbf{p}-\mathbf{o}_i)\|^2$, grows larger. However, all the points lying on $\mathbf{o}_i\mathbf{p}$ project to the same image point in the i-th camera, and thus the measure b_i should have the same error for all the points on $\mathbf{o}_i\mathbf{p}$. Therefore, the error formulation (2) is biased, and is prone to overestimate the error level for 3D points relatively far from the camera center. As a result, the naive midpoint method becomes inaccurate when the distances from the 3D point to different cameras vary considerably.

With this observation in mind, we propose to add a weight, which is inversely proportional to the distance between the 3D point and the camera center, for each term in (2) to rebalance the errors. The cost function is thus reformed as,

$$e(\mathbf{p}) = \sum\nolimits_{i=1}^{N} \|w_i(\mathbf{p})\mathbf{B}_i(\mathbf{p}-\mathbf{o}_i)\|^2, \tag{4}$$

with

$$w_i(\mathbf{p}) = 1/\|\mathbf{p} - \mathbf{o}_i\|.$$

For convenience, we denote the cost term with respect to the i-th camera as

$$e_i(\mathbf{p}) = \|w_i(\mathbf{p})\,\mathbf{e}_i(\mathbf{p})\|^2 = \frac{\|\mathbf{B}_i(\mathbf{p} - \mathbf{o}_i)\|^2}{\|\mathbf{p} - \mathbf{o}_i\|^2}. \quad (5)$$

The geometric relationship between different variables used in this work is illustrated in Figure 1.

The weight term w_i has a twofold influence on the cost function. On one hand, as shown in Figure 1, we have $\sin(\theta_i) = \|\mathbf{B}_i(\mathbf{p} - \mathbf{o}_i)\|/\|\mathbf{p} - \mathbf{o}_i\|$, and consequently the cost function represents a summation of squares of the sine values of different error angles θ_i. In this way, the cost function is unbiased for all the points lying on the same projection line. Moreover, when the error is small, the difference between $\sin(\theta_i)$ and θ_i is insignificant, so the proposed cost function provides a very close approximation of the exact angular error. On the other hand, however, minimizing the cost function (4) is no longer a linear least square problem, and hence adds complexity to the procedure of solution.

It should be noticed that the cost function formulated by equation (4) does not guarantee the solution to meet the chirality rules. In other word, the solution may locate behind some of the observing cameras. However, this only happens when the initial guess falls behind the camera or the observing cameras are singularly configured, which is rarely happened in real applications and can be easily handled by preprocessing.

C. Fixed Point Iterations

In order to minimize (4), its gradient is derived in closed form as

$$\nabla e(\mathbf{p}) = 2\sum\nolimits_{i=1}^{N}\left(w_i^2(\mathbf{p})(\mathbf{p} - \mathbf{o}_i)^{\mathrm{T}}(\mathbf{B}_i - e_i(\mathbf{p})\mathbf{I})\right). \quad (6)$$

Let $\mathbf{p}^*$ be the optimal solution, by writing explicitly the optimality condition $\nabla e(\mathbf{p}^*) = 0$, we have

$$\left(\sum\nolimits_{i=1}^{N}\mathbf{A}_i(\mathbf{p}^*)\right)\mathbf{p}^* = \sum\nolimits_{i=1}^{N}(\mathbf{A}_i(\mathbf{p}^*)\mathbf{o}_i + \mathbf{r}_i(\mathbf{p}^*)), \quad (7)$$

with $\mathbf{A}_i(\mathbf{p}) = w_i^2(\mathbf{p})\mathbf{B}_i$ and $\mathbf{r}_i(\mathbf{p}) = w_i^2(\mathbf{p})e_i(\mathbf{p})(\mathbf{p} - \mathbf{o}_i)$.

Equation (7) can be solved by fixed point iterations as the following algorithm. In the first step, an initial guess of the solution $\mathbf{p}_0$ can be obtained by solving (3). Then at each step, the estimation of the optimal point $\mathbf{p}_n$ is updated by solving

$$\left(\sum\nolimits_{i=1}^{N}\mathbf{A}_i(\mathbf{p}_{n-1})\right)\mathbf{p}_n = \sum\nolimits_{i=1}^{N}(\mathbf{A}_i(\mathbf{p}_{n-1})\mathbf{o}_i + \mathbf{r}_i(\mathbf{p}_{n-1})). \quad (8)$$

Since $\mathbf{p}_{n-1}$ is already known, (8) is a set of linear equations similar to (3) and the solution is trivial. This step is repeated until $\|\mathbf{p}_n - \mathbf{p}_{n-1}\|$ is smaller than a threshold. Then the converged $\mathbf{p}_n$ is a root of (7), and thus is the solution of the triangulation problem only if the starting point $\mathbf{p}_0$ is close enough to the global optimal. At present, we assume that this algorithm always converges, and will present a detailed discussion on the convergence property later. The proposed algorithm is called the *Iteratively Reweighted Mid-Point (IRMP)* method.

It should be noticed that minimizing (4) is a typical nonlinear least square problem, and thus can be solved using traditional gradient descent methods, such as the Gauss-Newton method, the Levenberg–Marquardt method or the Dogleg method [5]. When compared with them, the proposed algorithm consumes less computational resources at each iteration. This is because the proposed algorithm only updates the weight $w_i(\mathbf{p})$ as well as an additional term $\mathbf{r}_i(\mathbf{p})$ at each iteration (both $\mathbf{B}_i$ and $\mathbf{B}_i\mathbf{o}_i$ remain unchanged during the iterations and thus can be precomputed to reduce runtime), while traditional methods require the recomputation of the jacobian and the hessian matrix or even demand to solve an additional optimization problem to determine the update direction or the step length.

D. Approximation to Newton's method

The proposed algorithm is actually an approximation to the classic Newton's method at the neighborhood of the optimal solution. To make it clear, (6) and (8) are combined to produce

$$\left(2\sum\nolimits_{i=1}^{N}\mathbf{A}_i(\mathbf{p}_{n-1})\right)(\mathbf{p}_n - \mathbf{p}_{n-1}) = -\nabla e(\mathbf{p}_{n-1})^{\mathrm{T}}. \quad (9)$$

Meanwhile, when minimizing (4) using Newton's method, the iteration direction is determined by

$$\nabla^2 e(\mathbf{p}_{n-1})(\mathbf{p}_n - \mathbf{p}_{n-1}) = -\nabla e(\mathbf{p}_{n-1})^{\mathrm{T}}. \quad (10)$$

Here $\nabla^2 e(\mathbf{p}_{n-1})$ is the hessian matrix of $e(\mathbf{p}_{n-1})$, which can be derived in closed form as

$$\nabla^2 e(\mathbf{p}) = \sum\nolimits_{i=1}^{N}(2\mathbf{A}_i(\mathbf{p}) + \mathbf{D}_i(\mathbf{p}) + \mathbf{E}_i(\mathbf{p})), \quad (11)$$

where

$$\mathbf{D}_i(\mathbf{p}) = -4\left(\mathbf{A}_i(\mathbf{p})\mathbf{C}_i(\mathbf{p}) + \mathbf{C}_i(\mathbf{p})\mathbf{A}_i(\mathbf{p})\right), \quad (12)$$

$$\mathbf{E}_i(\mathbf{p}) = 2e_i(\mathbf{p})w_i^2(\mathbf{p})\left(4\mathbf{C}_i(\mathbf{p}) - \mathbf{I}\right), \quad (13)$$

and

$$\mathbf{C}_i(\mathbf{p}) = \frac{(\mathbf{p} - \mathbf{o}_i)(\mathbf{p} - \mathbf{o}_i)^{\mathrm{T}}}{\|\mathbf{p} - \mathbf{o}_i\|^2}. \quad (14)$$

When $\mathbf{p}$ moves toward the global optimal, both $\sin(\theta_i)$ and $e_i(\mathbf{p})$ approach to zero. Consequently, $\mathbf{E}_i(\mathbf{p})$ is negligible compared to $\mathbf{A}_i(\mathbf{p})$ and can be dropped in (11) at the neighborhood of the global optimal. Another important observation is that $\mathbf{D}_i(\mathbf{p})$ is also constituted by negligible tiny values near the global optimal, which can be clarified by investigating the norm of $\mathbf{A}_i(\mathbf{p})\mathbf{C}_i(\mathbf{p})$, namely,

$$\|\mathbf{A}_i(\mathbf{p})\mathbf{C}_i(\mathbf{p})\| = \frac{\left\|\mathbf{A}_i(\mathbf{p})(\mathbf{p} - \mathbf{o})(\mathbf{p} - \mathbf{o})^T\right\|}{\|\mathbf{p} - \mathbf{o}\|^2} \quad (15)$$

$$\leq \frac{\|\mathbf{A}_i(\mathbf{p})(\mathbf{p} - \mathbf{o})\|}{\|\mathbf{p} - \mathbf{o}\|}\frac{\|\mathbf{p} - \mathbf{o}\|}{\|\mathbf{p} - \mathbf{o}\|} \quad (16)$$

$$= w_i^2(\mathbf{p})\sin(\theta_i) \to 0. \quad (17)$$

The same result also holds for $\mathbf{C}_i(\mathbf{p})\mathbf{A}_i(\mathbf{p})$, and as a consequence $\|\mathbf{D}_i(\mathbf{p})\|$ is close to zero. Now, we can conclude that

$$\nabla^2 e(\mathbf{p}) \approx 2\sum\nolimits_{i=1}^{n} \mathbf{A}_i(\mathbf{p}) \tag{18}$$

near the optimal point. Recalling (9) and (10), it indicates that the proposed IRMP method is an approximation to minimize the cost function (4) using Newton's method when the starting point is close enough to the global optimal.

E. Convergency

Recalling $\mathbf{A}_i(\mathbf{p}) = w_i^2(\mathbf{p})\mathbf{B}_i$ and the attributes of $\mathbf{B}_i$ discussed before, $\mathbf{A}_i$ is symmetric and its eigenvalues are $\{w_i^2(\mathbf{p}), w_i^2(\mathbf{p}), 0\}$, while the eigenvector corresponding to the eigenvalue 0 is $\mathbf{b}_i$. $\sum_{i=1}^{N} \mathbf{A}_i(\mathbf{p}_n)$ would most likely to be a positive definite matrix because of the dominant places of the positive eigenvalues in each $\mathbf{A}_i$, with the only exception that all the measurements $\mathbf{b}_i$ are parallel (suggesting the triangulation problem itself is ill-conditioned). Therefore, as long as the triangulation problem is well defined, (9) produces a descent direction and the proposed IRMP method can at least reach a local minimum. In practice, the ratio between the minimum and the maximum of the eigenvalues of $\sum_{i=1}^{N} \mathbf{A}_i(\mathbf{p}_n)$ can be utilized as a criterion to determine whether the triangulation problem is ill-conditioned.

When $\mathbf{p}$ is out of the scope of the global mininum, the contributions of $\mathbf{D}_i(\mathbf{p})$ and $\mathbf{E}_i(\mathbf{p})$ may bring negative eigenvalues to the hessian matrix (11), and therefore the cost function used in the IRMP method may not be convex. When the starting point is far from the global minimum, the iteration characterized by (8) may get suck at local minima. As addressed in [6], this is a common drawback of using the ℓ_2 norm based cost functions in multiple view geometry.

Fortunately, the IRMP method starts from the solution of the classic midpoint method, which is the unique global minimum of (2) and should be close to the global minimum of (4) when the triangulation problem is well defined. As we have discussed before, near the global minimum, the proposed IRMP method is an approximation of the Newton's method, and thus inherits the quadratic convergence rate as the Newton's method has. At the meantime, the IRMP method requires less computation at each iteration, and thus is faster than using the naive Newton's method to minimize (4). Several 2D toy examples are presented in Figure 2, which validate the analysis above.

III. Experiments

In this section, experimental results on both synthetic and real data are presented to validate the efficiency and the accuracy of the proposed IRMP method. The proposed IRMP method is compared with three benchmark algorithms. The first one is the naive *Multiple View Middle Point (MVMP)* method. To the best of our knowledge, though usually labelled as inaccurate, the MVMP method is currently the fastest method for multiple view triangulation. Besides, as the proposed IRMP method is derived from the MVMP method, it would be interesting to compare their behaviors under different conditions. The second algorithm chosen as a benchmark is minimizing the cost function (4) using the *NN (Naive Newton's)* method. As illustrated in the previous section, the IRMP method can be viewed as an approximation of the NN method by dropping some insignificant terms in the hessian matrix. In this section, some experimental results will be presented to show the influence of the approximation on the accuracy and efficiency of the triangulation results. The third benchmark algorithm is the gradient-based minimization of the ℓ_2 norm based reprojection errors, which is abbreviated as the *GMRE (Gradient Minimization of the Reprojection Errors)* method. This method is well balanced between the efficiency and the accuracy and thus popularly used in recent state-of-the-art SLAM systems, including [12]–[15]. The result from the MVMP method is used to initialize the other three methods. The reprojection error, so-called the golden standard [1], is used as a criterion to compare the accuracies of different algorithms. The same camera intrinsics are used in all the synthetic experiments, with an image size of 1024×1024 pixels and a focus length of 400 pixels. In the GMRE implementation, the reprojection error is minimized using Gauss-Newton iterations. All the experiments were done on an Intel 3.70GHz i7-8700K CPU with 16GB memory. In the first three experiments, we implemented all the algorithms by ourselves using MATLAB on a single thread without any special optimization. In the last experiment, we compared the C++ implementation of the IRMP method with the C++ implementation of the GRME method in SVO [12]. The process of triangulation using the GRME method is traditionally called the structure-only bundle adjustment in SVO.

A. Monte Carlo simulation

We set up a Monte Carlo simulation to validate the accuracy and the efficiency of the IRMP method. In the simulation, the observed 3D point was fixed at (0, 0, 0), and 100 cameras were located at random aspects to observe the 3D point. The distances between the 3D point and the cameras were randomly generated with the continuous uniform distribution in the range of $[d_\mathrm{s}, d_\mathrm{m}]$. The ratio $\gamma = d_\mathrm{m}/d_\mathrm{s}$ was used to describe the dispersion of the distances between the 3D point and the cameras. The space point was then projected on the camera planes, and the results were corrupted using Guassian noise with a 10-pixels covariance. After that, the 3D point was reconstructed using the corrupted measurements by all the four triangulation methods, respectively. At last, the triangulation results were projected back to the image planes to compute the average reprojection errors. In this experiment, γ is altered from 1 to 100 to investigate its influence on the details of triangulation. For each γ, the simulation was repeated for 10,000 times and the average result was reported.

As shown in Figure 3, when $\gamma = 1$, all the four methods produced similar results. However, the accuracy of the MVMP method degrades significantly when γ increases, which validates the theoretical analysis presented in the previous section. Moreover, the experimental results reveal

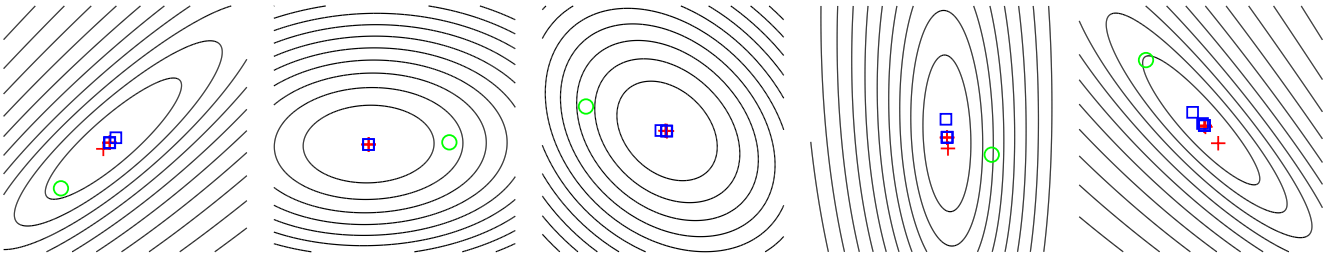
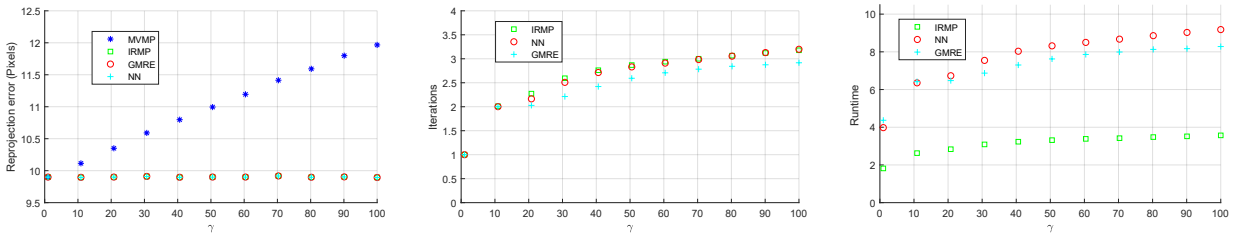


Fig. 2. Several 2D toy examples on the convergence property of the IRMP method. The black curve is the contour of (4) near its global minimum. The green circle is the solution of the classic midpoint method, which is the global minimun of (2) and usually at the neighborhood of the global minimum of (4). The blue squares represent the trace of the IRMP method while the red crosses represent the trace of minimizing (4) using naive Newton's method.



(a) Accuracy (b) Iterations take before converge (c) Runtime (normalized by dividing the runtime of the MVMP method)

Fig. 3. The Monte Carlo simulation on the accuracy and the efficiency of the IRMP method.

an almost linear relationship between γ and the reprojection errors of the MVMP method. In the presented simulation, the IRMP method can always refine the accuracy to an acceptable level. And the differences between the accuracies of the IRMP method, the NN method, and the GMRE method are insignificant. Besides, The iterations of the IRMP method needed for convergence increases when γ grows larger. The average iterations the IRMP method, the NN method, and the GRME method take before converge are almost identical while the IRMP method has higher efficiency because it takes less computation at each iteration.

If we continue to increase γ in the simulation, the MVMP method may give extremely bad initializations, starting from which the rest three iterative methods may diverge. A two-phase initialization is suggested to handle these failure cases. After the first initialization p_0 was computed by the MVMP method, γ can be computed using p_0. If $\gamma > 100$, those observations located relatively in large distance will be eliminated to make $\gamma \leq 100$. And then a second initialization p_1 is computed using the rest observations by the MVMP method. After that all the observations are used to compute the final solution by the IRMP method starting from p_1. We found that $\gamma > 100$ rarely happens in real applications.

B. Evaluation in Synthetic Scenarios

We synthesized four different scenarios:

- A: The camera moves along a curve trajectory towards the 3D points, as shown in Figure 4(a). This simulates the scenario where the robot moves toward the object it observes or the flying robot lands [18].
- B: The camera moves along a curve trajectory through the 3D points, as shown in Figure 4(b). This simulates that the robot freely explores a space with randomly distributed landmarks, which is the target of most SLAM systems [19].
- C: The camera moves on a circle around the 3D points, as shown in Figure 4(c). This simulates 3D modelling with a rotating platform [20].
- D: Both the cameras and the 3D points are randomly distributed, as shown in Figure 4(d). This simulates large-scale 3D reconstruction with the images from various sources [21].

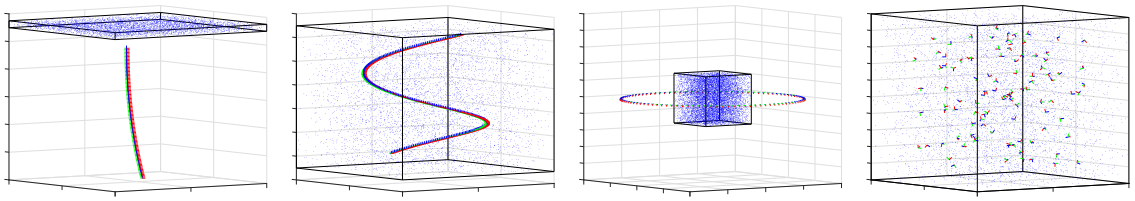
TABLE I

THE COMPARISONS OF TOTAL RUNTIME IN SECONDS USING SYNTHETIC DATA.

Configuration			Runtime(seconds)			
Name	Points	Cameras	MVMP	IRMP	NN	GMRE
A	5000	100	1.602	**4.377**	11.591	11.217
B	5000	100	1.361	**3.021**	9.103	9.476
C	5000	100	2.275	**5.240**	12.975	12.937
D	5000	100	0.558	**1.5571**	3.146	3.234

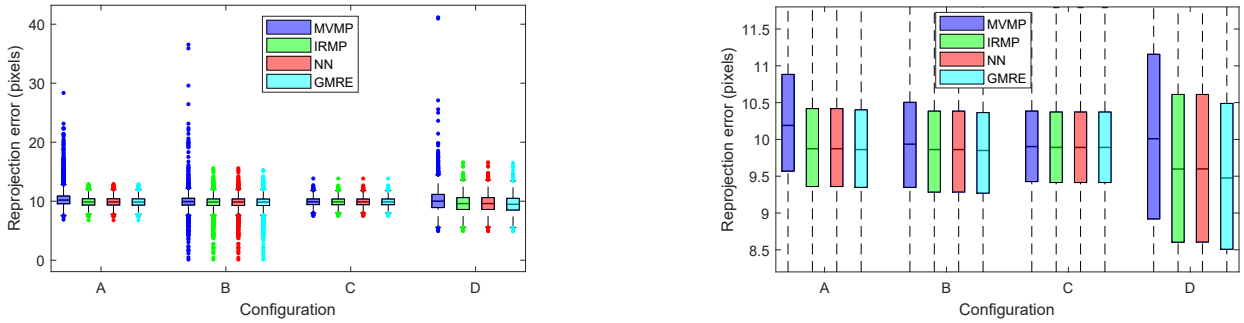
As shown in Figure 4, 5000 uniformly distributed random 3D points were generated within the black box and projected back to 100 cameras. Then those projections within the corresponding camera's scope and passed the chirality check were corrupted using Guassian noise with a 10-pixels covariance. After that, those 3D points with more than 2 valid projections were triangulated for four times by MVMP, IRMP, NN, GMRE, respectively. At last, the triangulation results were projected back to the image planes again to compute the average reprojection errors.

Table (I) shows the comparisons of total runtime used to triangulate the 3D points by different algorithms. It is



(a) Configuration A (b) Configuration B (c) Configuration C (d) Configuration D

Fig. 4. The synthetic data configurations. The 3D points are drawn as blue dots. Their bounding boxes are drawn as black cubes. The cameras has red x axis, green y axis as well as blue z axis.



(a) The boxplots of the reprojection errors of different algorithms (b) Zoom in of Figure 5(a)

Fig. 5. The comparisons of reprojection errors in pixels in synthetic scenarios.

not surprising that the MVMP method is always the fastest algorithm in all four configurations. The proposed IRMP method is about 2 to 3 times slower than the MVMP method, but is still about 2 to 3 times faster than the GMRE method and the NN method. Figure 5 shows the comparisons of the reprojection errors. It can be seen that the proposed IRMP method is almost as accurate as the GMRE method and the NN method. The MVMP method, however, behaves much worse in A, B and D. As we have discussed before, the MVMP method is inaccurate because it has a biased cost function. In A, B and D, some of the 3D points are observed by cameras from tremendously different distances, thus are not correctly triangulated by the MVMP method. In contrast, the IRMP method can refine those 3D points with bad initiations to an acceptable accuracy level. Another interesting phenomenon is that the MVMP method is as accurate as the other three methods in C. This is because the distances between a 3D point and all the observing cameras have smaller differences in C, alleviating the bias of the cost function (2). In the most extreme case where a 3D point has the same distances to all the cameras observing it, the MVMP method is equivalent to the proposed IRMP method, and hence has the same accuracy.

C. Evaluation in Real Scenarios

The algorithms were tested on publicly available datasets [22]–[25] for large scale 3D reconstruction. The camera matrixes and the feature correspondences supplied with these datasets were used in the experiments. As demonstrated in Table II, the MVMP method produces the fastest as well as the most inaccurate results on all the datasets. The proposed IRMP method brings about 0.01 average pixel error reduction against the MVMP method, which is insignificant at the first sight. But if we take a close look at every specific point in the datasets, the accuracy improvement brought by the IRMP can be as large as 6.312 pixels (see columns with the header MER in Table II, which record the max error reductions against the MVMP method). When compared with the state-of-the-art GMRE methods, the proposed IRMP method is about 2-3 times faster while achieves fairly close accuracy. Some of the triangulation results by the IRMP method are demonstrated in Figure 6.

D. Application to SLAM

We have integrated the C++ implementation of the proposed IRMP method [1] into our own SLAM system, which consists of a stereo SVO based front-end [12] and an ORB-SLAM based [13] loop closure mechanism. In the original SVO implementation, the procedure of multiple view triangulation is done by the GMRE method. In this experiment, the IRMP method was used to substitute the GMRE method. The SLAM systems were evaluated on public available dataset KITTI [26], and the results are illustrated on Figure 7. All the accuracy comparison results were generated using the toolbox published along with a recent tutorial on quantitative trajectory evaluation for visual odometry [27]. As illustrated in Figure 7, the accuracies of the IRMP-based SLAM and the GMRE-based SLAM are very close. Meanwhile, the IRMP-based SLAM took about 0.533 millisecond on average to compute multiple view triangulation for each frame while the GMRE method took 1.217 milliseconds.

[1] https://github.com/yang3kui/triangulation_IRMP

TABLE II

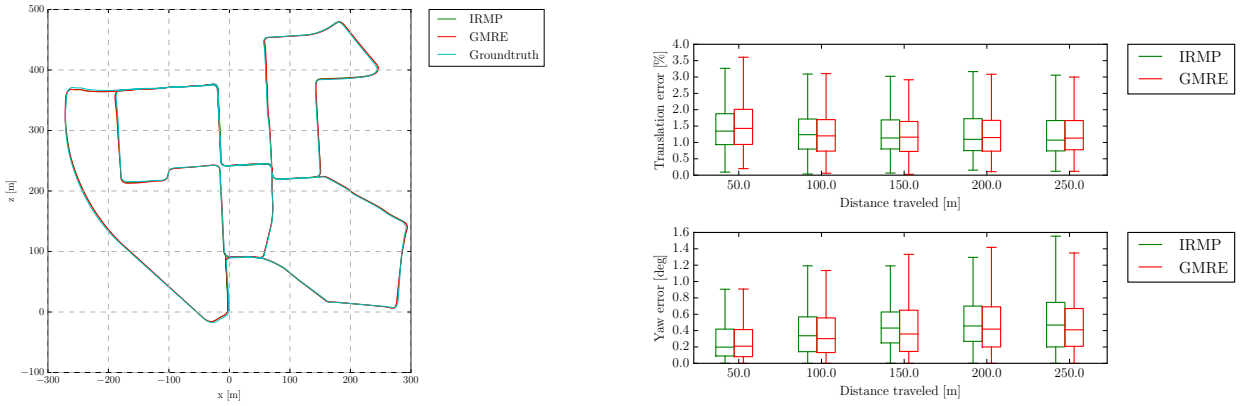
THE COMPARISONS OF RESULTS ON LARGE SCALE REAL DATASETS. MER IS ABBREVIATED FOR THE MAX ERROR REDUCTION AGAINST THE MVMP METHOD FOR A SPECIFIC INSTANCE.

Datasets			Runtime(seconds)				Reprojection errors(pixels)				MER(pixels)		
Name	Points	Cameras	MVMP	IRMP	NN	GMRE	MVMP	IRMP	NN	GMRE	IRMP	NN	GMRE
A	354134	800	13.201	**31.176**	87.197	93.562	0.816	**0.805**	0.805	0.805	**5.327**	5.327	5.329
B	159055	1208	12.392	**29.312**	82.527	88.643	1.088	**1.078**	1.078	1.077	**2.161**	2.161	2.161
C	231507	1498	20.010	**47.744**	130.689	140.804	0.807	**0.799**	0.799	0.798	**6.312**	6.312	6.314
D	53857	761	6.510	**14.208**	50.248	54.215	0.942	**0.936**	0.936	0.936	**0.879**	0.880	0.881
E	9099	34	0.399	**0.868**	2.198	2.399	0.397	**0.386**	0.386	0.386	**0.901**	0.901	0.901
F	139951	290	8.501	**17.999**	54.969	60.073	0.970	**0.968**	0.968	0.967	**0.413**	0.413	0.414
G	74423	368	5.118	**12.340**	37.822	40.415	1.032	**1.023**	1.023	1.022	**0.953**	0.953	0.957

Full names of the Datasets [22], [23]: A → Aos Hus ; B → Lund Cathedral; C → San Marco; D → Örebro Castle; E → Park gate, Clermont-Ferrand; F → Buddah Statue; G → Skansen Lejonet, Gothenburg;

(a) Dinosaur [24] (b) NotreDame [25] (c) Örebro Castle [22], [23] (d) East Indiaman [22], [23] (e) Buddah Statue [22], [23]

Fig. 6. The images and the triangulation results of the proposed IRMP method. The images on the top row are from the corresponding datasets and the blue points clouds on the bottom row are the triangulation results of the IRMP method.



(a) The birdview trajectories result from the IRMP-based SLAM and the GMRE-based SLAM are presented together with the ground truth.

(b) The comparison of accuracies, including translation error and yaw error, between the IRMP-based SLAM and the GMRE-based SLAM.

Fig. 7. Comparison of SLAM results on KITTI [26].

IV. CONCLUSION

In this work, we clarified that the classic midpoint method is inaccurate due to its biased cost function. Therefore, in order to improve the accuracy, we proposed to assign a weight to each measurement to rebalance the cost function. The proposed unbiased cost function can be minimized using fixed point iterations at a quadratic convergence rate near the optimal point. Besides, at each iteration, the proposed method consumes less computational resources than traditional gradient descent methods, and thus is faster than the state-of-the-art. Experimental results on synthetic and real data are also presented, which validate that the proposed method improves the accuracy when compared with the classic midpoint method and is more efficient than the state-of-the-art while achieves comparable accuracy. Therefore, the proposed method can be integrated to accelerate future SLAM or SfM systems in practice.

Some of the basic principles developed in this work are also applicable to other multiple view geometry problems, such as the perspective-n-point problem and the camera

resectioning problem. We will study them in future work.

ACKNOWLEDGMENT

The authors would like to thank Dr. Li Zhu from Beihang university and Dr. Zhang Zichao from University of Zurich, for their helps to improve the quality of this work.

REFERENCES

[1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[2] R. I. Hartley and P. Sturm, "Triangulation," *Comput. Vis. Image Underst.*, vol. 68, no. 2, pp. 146–157, Nov. 1997.

[3] P. Lindstrom, "Triangulation made easy." in *CVPR*. IEEE Computer Society, 2010, pp. 1554–1561.

[4] H. Stewenius, F. Schaffalitzky, and D. Nister, "How hard is 3-view triangulation really?" in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1, Oct 2005, pp. 686–693 Vol. 1.

[5] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[6] R. Hartley, F. Kahl, C. Olsson, and Y. Seo, "Verifying global minima for l2 minimization problems in multiple view geometry," *Int. J. Comput. Vision*, vol. 101, no. 2, pp. 288–304, Jan. 2013.

[7] R. I. Hartley and F. Schaffalitzky, "L-infinity minimization in geometric reconstruction problems," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[8] F. Kahl and R. Hartley, "Multiple-view geometry under the L_infty-norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1603–1617, Sept 2008.

[9] A. Eriksson and M. Isaksson, "Pseudoconvex proximal splitting for l-infinity problems in multiview geometry," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 4066–4073.

[10] Q. Zhang, T. J. Chin, and D. Suter, "Quasiconvex plane sweep for triangulation with outliers," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 920–928.

[11] Q. Zhang and T. J. Chin, "Coresets for triangulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 15–22.

[13] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. [Online]. Available: https://doi.org/10.1109/TRO.2017.2705103

[14] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[15] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018. [Online]. Available: https://doi.org/10.1109/LRA.2018.2793349

[16] S. Ramalingam, S. K. Lodha, and P. Sturm, "A generic structure-from-motion framework," *Comput. Vis. Image Underst.*, vol. 103, no. 3, pp. 218–228, Sep. 2006.

[17] K. Aftab, R. Hartley, and J. Trumpf, "Lq-closest-point to affine subspaces using the generalized weiszfeld algorithm," *Int. J. Comput. Vision*, vol. 114, no. 1, pp. 1–15, Aug. 2015.

[18] O. Araar, N. Aouf, and I. Vitanov, "Vision based autonomous landing of multirotor uav on moving platform," vol. 85, 08 2016.

[19] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec 2016.

[20] "3d scanners," http://3dprintingsystems.com/products/3d-scanners/, accessed September 3, 2018.

[21] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011.

[22] O. Enqvist, F. Kahl, and C. Olsson, "Non-sequential structure from motion," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011, pp. 264–271.

[23] C. Olsson and O. Enqvist, "Stable structure from motion for unordered image collections," in *Image Analysis*, A. Heyden and F. Kahl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 524–535.

[24] "Oxford visual geometry group. multi-view and oxford colleges building reconstruction," http://www.robots.ox.ac.uk/ vgg/data/data-mview.html, accessed September 3, 2018.

[25] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[27] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.