Quasiconvex Plane Sweep for Triangulation with Outliers

Qianggong Zhang Tat-Jun Chin David Suter
The University of Adelaide
Adelaide, South Australia, 5005, Australia
qianggong.zhang@adelaide.edu.au, tat-jun.chin@adelaide.edu.au, david.suter@adelaide.edu.au

Abstract

Triangulation is a fundamental task in 3D computer vision. Unsurprisingly, it is a well-investigated problem with many mature algorithms. However, algorithms for robust *triangulation, which are necessary to produce correct results in the presence of egregiously incorrect measurements (i.e., outliers), have received much less attention. The default approach to deal with outliers in triangulation is by random sampling. The randomized heuristic is not only suboptimal, it could, in fact, be computationally inefficient on large-scale datasets. In this paper, we propose a novel locally optimal algorithm for robust triangulation. A key feature of our method is to efficiently derive the local update step by plane sweeping a set of quasiconvex functions. Underpinning our method is a new theory behind quasiconvex plane sweep, which has not been examined previously in computational geometry. Relative to the random sampling heuristic, our algorithm not only guarantees deterministic convergence to a local minimum, it typically achieves higher quality solutions in similar runtimes*[1].

1. Introduction

Triangulation is the task of estimating the 3D coordinates of a scene point from multiple 2D image observations of the point, given that the pose of the cameras are known [14]. The task is of fundamental importance to 3D vision, since it enables the recovery of the 3D structure of a scene.

Most 3D reconstruction pipelines estimate 3D structure and camera poses simultaneously (via bundle adjustment, factorization, or equivalent steps). However, triangulation can play an important role in densifying or refining the 3D structure, by estimating the 3D coordinates of additional image measurements (e.g., extracted from original high-resolution images) based on the optimized camera poses.

Since 2D feature detection and association methods are not perfect, they inevitably create wrong feature correspondences and tracks. In an SfM pipeline, outliers are removed during the robust relative pose estimation step. However, outliers will exist in the additional feature correspondences extracted post-SfM, since they were not subjected to the SfM pipeline (e.g. for efficiency reasons).

For *non-robust* triangulation, the ℓ_∞ paradigm [13] has been influential. Given a set of N 2D image measurements $\{\mathbf{u}_i\}_{i=1}^N$ of the same scene point $\mathbf{x} \in \mathbb{R}^3$, and the associated camera matrices $\{\mathbf{P}_i\}_{i=1}^N$ with each $\mathbf{P}_i \in \mathbb{R}^{3\times 4}$, we estimate $\mathbf{x}$ by minimizing the maximum reprojection error

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \underset{i\in\{1,\dots,N\}}{\text{maximum}} \left\| \mathbf{u}_i - \frac{\mathbf{P}_i^{1:2}\tilde{\mathbf{x}}}{\mathbf{P}_i^3\tilde{\mathbf{x}}} \right\|_p, \\ \text{s.t.} \quad & \mathbf{P}_i^3\tilde{\mathbf{x}} > 0 \;\; \forall i \in \{1,\dots,N\}, \end{aligned} \tag{1}$$

where $\mathbf{P}^{1:2}$ is the first-two rows of $\mathbf{P}$, $\mathbf{P}^3$ is the third row of $\mathbf{P}$, and $\tilde{\mathbf{x}}$ is $\mathbf{x}$ in homogeneous coordinates. The positivity constraints $\mathbf{P}_i^3\tilde{\mathbf{x}} > 0$ ensure that $\mathbf{x}$ lies in front of all the cameras. In the above, $\|\cdot\|_p$ indicates a valid p-norm; usually p is taken to be 1, 2 or ∞.

Algorithms to solve (1) take advantage of quasiconvexity to efficiently find the global minimizer $\mathbf{x}^*$ [15, 16, 25, 23, 3]. Recently, Donné et al. [9] showed that their *polyhedron collapse* algorithm (for $p=\infty$) is the fastest.

A major weakness of the ℓ_∞ paradigm, however, is that the estimate is easily biased by outlying measurements. To fix this issue, the usage of an inherently robust cost function is necessary. A popular robust criterion is least median squares (LMS) [24]; for triangulation, this entails solving

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \underset{i\in\{1,\dots,N\}}{\text{median}} \left\| \mathbf{u}_i - \frac{\mathbf{P}_i^{1:2}\tilde{\mathbf{x}}}{\mathbf{P}_i^3\tilde{\mathbf{x}}} \right\|_p, \\ \text{s.t.} \quad & \mathbf{P}_i^3\tilde{\mathbf{x}} > 0 \;\; \forall i \in \{1,\dots,N\}, \end{aligned} \tag{2}$$

i.e., minimize the median error. LMS provably has a breakdown point of 0.5, which means that it can tolerate *up to* 50% of outliers [24, Chap. 3]. The drawback of LMS, however, is that the median of the reprojection errors is not quasiconvex, and problem (2) becomes intractable in general. The non-differentiability of the median also complicates the usage of standard gradient-based optimization [21].

[1]See supplementary material for demo program.

Existing algorithms for LMS Most practitioners rely on the random sampling heuristic to approximately solve LMS [30, 24]. Specifically, we randomly sample minimal subsets of the measurements to estimate $\mathbf{x}$ (using, e.g., DLT for triangulation), then select the estimate with the lowest median error. A probabilistic upper bound of the number of samples to take can be deduced based on the highest expected outlier rate of 0.5 [1]. Apart from being non-deterministic, a noticeable weakness of random sampling is that it provides no optimality guarantees.

Ke and Kanade [16] used the bisection technique endowed with a *non-convex* feasibility test to solve general quasiconvex LMS problems, which includes (2). For tractability, a relaxed feasibility test which is more conservative is performed, thus the method can only converge to an approximate LMS solution without any certificate of optimality (either local or global).

On the other extreme, combinatorial search algorithms have been proposed to solve LMS exactly [28, 4]. For triangulation, Li [17] exploited the quasiconvexity of the reprojection error to devise a search algorithm that enumerates all local minima of the LMS problem. Despite the low-dimensionality of $\mathbf{x}$, the exact algorithms are computationally costly, and are practical only for small instances.

Our contributions We propose a novel *locally optimal* algorithm for LMS triangulation (2). At each iteration, our approach calculates the update via a 1D quasiconvex LMS problem, which can be solved efficiently via plane sweep. We develop the necessary theory and algorithm for quasiconvex plane sweep, and establish the convergence of the overall algorithm to a local minimum.

Experimentally, we show that our method consistently yields better solutions than random sampling with comparable runtimes. Further, our technique is much faster and practical than the globally optimal methods.

Differentiation against RANSAC Another popular robust technique in computer vision is RANSAC [12]. Unlike LMS, the aim of RANSAC is to maximize the number of inliers, given a threshold. Whether LMS or inlier maximization is the "better" criterion is debatable—certainly for robust triangulation, both are valid and widely used.

The optimization machinery in RANSAC is random sampling, thus, it shares the disadvantages of the randomized heuristic for LMS mentioned above. Of course, there are alternative methods for inlier maximization, e.g., RANSAC variants [6], branch-and-bound [18] and subset search [5]. We stress, however, that from an optimization viewpoint, these methods solve a different problem to LMS and are not strictly comparable to our algorithm (not to mention that global methods for inlier maximization would also be costly, similar to global methods for LMS).

2. Background

First, define and rewrite the i-th reprojection error as

$$r_i(\mathbf{x}) = \left\| \mathbf{u}_i - \frac{\mathbf{P}_i^{1:2}\tilde{\mathbf{x}}}{\mathbf{P}_i^3\tilde{\mathbf{x}}} \right\|_p = \frac{\|\mathbf{A}_i\mathbf{x} + \mathbf{b}_i\|_p}{\mathbf{c}_i^T\mathbf{x} + d_i}, \quad (3)$$

$$\text{where } \mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{i,1}^T \\ \mathbf{a}_{i,2}^T \end{bmatrix} \in \mathbb{R}^{2\times 3}, \;\; \mathbf{b}_i = \begin{bmatrix} b_{i,1} \\ b_{i,2} \end{bmatrix} \in \mathbb{R}^2, \quad (4)$$

$\mathbf{c}_i \in \mathbb{R}^3$ and d_i are constants calculated from the data $\mathbf{P}_i$ and $\mathbf{u}_i$. For $p \geq 1$, $r_i(\mathbf{x})$ is quasiconvex [11].

To enable direct comparison with the state-of-the-art polyhedron collapse method, we also base our method on the same $p = \infty$. The reprojection error thus becomes

$$r_i(\mathbf{x}) = \max\left(\frac{|\mathbf{a}_{i,1}^T\mathbf{x} + b_{i,1}|}{\mathbf{c}_i^T\mathbf{x} + d_i}, \frac{|\mathbf{a}_{i,2}^T\mathbf{x} + b_{i,2}|}{\mathbf{c}_i^T\mathbf{x} + d_i} \right), \quad (5)$$

which can be further developed into

$$\begin{aligned} r_i(\mathbf{x}) &= \max\left(\frac{\mathbf{a}_{i,1}^T\mathbf{x} + b_{i,1}}{\mathbf{c}_i^T\mathbf{x} + d_i}, \frac{-\mathbf{a}_{i,1}^T\mathbf{x} - b_{i,1}}{\mathbf{c}_i^T\mathbf{x} + d_i}, \right. \\ &\qquad \left. \frac{\mathbf{a}_{i,2}^T\mathbf{x} + b_{i,2}}{\mathbf{c}_i^T\mathbf{x} + d_i}, \frac{-\mathbf{a}_{i,2}^T\mathbf{x} - b_{i,2}}{\mathbf{c}_i^T\mathbf{x} + d_i} \right) \quad (6) \\ &= \max\left(r_{i,1}(\mathbf{x}), r_{i,2}(\mathbf{x}), r_{i,3}(\mathbf{x}), r_{i,4}(\mathbf{x}) \right). \quad (7) \end{aligned}$$

For simplicity, we define $r_{i,j}(\mathbf{x})$, $j = 1, \ldots, 4$, as

$$r_{i,j}(\mathbf{x}) = \frac{\mathbf{a}_{i,j}^T\mathbf{x} + b_{i,j}}{\mathbf{c}_i^T\mathbf{x} + d_i}; \quad (8)$$

the reader should be reminded that constants $\mathbf{a}_{i,j}$ and $b_{i,j}$ should be taken with the appropriate sign from the input data. Henceforth, we call the $r_{i,j}(\mathbf{x})$'s "constraints".

Defining $r_i(\mathbf{x})$ as above, i.e., as a $\max$ over four linear fractional terms, will be crucial for clarifying the operations of our method later. For now, we re-express the LMS triangulation problem equivalently as

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \underset{i\in\{1,\ldots,N\}}{\text{median}} \; r_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{c}_i^T\mathbf{x} + d_i > 0 \;\; \forall i \in \{1, \ldots, N\}, \end{aligned} \quad (9)$$

where the input data is $\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$.

3. Locally optimal LMS triangulation

Algorithm 1 describes the proposed locally optimal method (called **Q-sweep**) to solve (9). The overall structure of Q-sweep is simple—given an initial feasible estimate $\hat{\mathbf{x}}$, find a direction $\Delta\mathbf{x}$ and step size α to adjust $\hat{\mathbf{x}}$ such that the median error decreases; stop when a valid $\Delta\mathbf{x}$ cannot be found. A similar overall structure exists in polyhedron collapse, and indeed in many techniques in the wider optimization literature [21]. Nonetheless, there are significant novelties in our work, namely, an efficient routine to compute the optimal step size α for LMS triangulation, and theoretical analyses on convergence and complexity.

Algorithm 1 Q-sweep method for LMS triangulation.

Require: Input data $\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$, initial soln. $\hat{\mathbf{x}}$.
$\Delta\mathbf{x} \leftarrow \texttt{DESCENTDIR}\big(\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N, \hat{\mathbf{x}}\big)$.
while $\Delta\mathbf{x}$ is not null **do**
  $\alpha \leftarrow \texttt{STEPSIZE}\big(\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N, \hat{\mathbf{x}}, \Delta\mathbf{x}\big)$.
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \alpha\Delta\mathbf{x}$.
  $\Delta\mathbf{x} \leftarrow \texttt{DESCENTDIR}\big(\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N, \hat{\mathbf{x}}\big)$.
end while
return $\hat{\mathbf{x}}$.

The rest of this section is devoted to fleshing out Algorithm 1 (details on initialization are postponed until Sec. 4).

3.1. Finding descent direction

Algorithm 2 describes the routine DESCENTDIR used in Q-sweep to find $\Delta\mathbf{x}$ for the current estimate $\hat{\mathbf{x}}$. The routine begins by finding the set of residuals $\mathcal{A}$ and constraints $\mathcal{J}$ that are active, i.e., has the same value as the *median* error for $\hat{\mathbf{x}}$. Given $\mathcal{J}$, the rest of the routine largely follows the procedure of polyhedron collapse to calculate $\Delta\mathbf{x}$. For brevity, we will give only high-level account of the method.

Algorithm 2 DESCENTDIR to find descent direction.

Require: Input data $\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$, an estimate $\hat{\mathbf{x}}$.
1: $\hat{r} \leftarrow \text{med}_i\ r_i(\hat{\mathbf{x}})$.
2: $\mathcal{A} \leftarrow \{p \mid r_p(\hat{\mathbf{x}}) = \hat{r}\}$.
3: $\mathcal{J} \leftarrow \{(p,q) \mid p \in \mathcal{A},\ r_{p,q}(\hat{\mathbf{x}}) = \hat{r}\}$.
4: $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \dots\} \leftarrow$ set of normals for the constraints indexed by $\mathcal{J}$; see (11) and surrounding text.
5: $\Delta\mathbf{x} \leftarrow$ null.
6: **if** $|\mathcal{N}| = 1$ **then**
7:   $\Delta\mathbf{x} \leftarrow \mathbf{n}_1$.
8: **else if** $|\mathcal{N}| = 2$ **then**
9:   $\Delta\mathbf{x} \leftarrow \mathbf{n}_1 + \mathbf{n}_2$.
10: **else**
11:   **for** each triplet $(\mathbf{n}_u, \mathbf{n}_v, \mathbf{n}_w)$ of $\mathcal{N}$ **do**
12:     $\mathbf{y} \leftarrow \mathbf{n}_u \times \mathbf{n}_v + \mathbf{n}_v \times \mathbf{n}_w + \mathbf{n}_w \times \mathbf{n}_u$.
13:     $s \leftarrow \langle \mathbf{n}_u, \mathbf{y}\rangle / |\langle \mathbf{n}_u, \mathbf{y}\rangle|$.
14:     $\mathbf{y} \leftarrow s\mathbf{y}$.
15:     **if** $\langle \mathbf{y}, \mathbf{n}\rangle > 0,\ \forall \mathbf{n} \in \mathcal{N}$ **then**
16:       $\Delta\mathbf{x} \leftarrow \mathbf{y}$.
17:       Break.
18:     **end if**
19:   **end for**
20: **end if**
21: **return** $\Delta\mathbf{x}$.

Each active constraint $r_{p,q}(\mathbf{x})$ indexed by $\mathcal{J}$ defines a 2D plane in 3D space, i.e.,

$$\begin{aligned} r_{p,q}(\mathbf{x}) &= \frac{\mathbf{a}_{p,q}^T\mathbf{x} + b_{p,q}}{\mathbf{c}_p^T\mathbf{x} + d_p} = \hat{r}, \\ \implies &(\mathbf{a}_{p,q}^T - \hat{r}\mathbf{c}_p^T)\mathbf{x} + b_{p,q} - \hat{r}d_p = 0. \end{aligned} \tag{10}$$

The normal of the plane pointing towards the negative direction is given by

$$\mathbf{n} = -(\mathbf{a}_{p,q}^T - \hat{r}\mathbf{c}_p^T). \tag{11}$$

The normal $\mathbf{n}$ is also the direction where $r_{p,q}(\mathbf{x})$ will reduce in value, starting from the point $\hat{\mathbf{x}}$.

If $\mathcal{J}$ has only one element, then the normal of that constraint is installed as $\Delta\mathbf{x}$ (Step 7). If there are more than one active constraints, then the normals of the constraints are combined: by a simple addition if there are two active constraints (Step 9), or if there are more active constraints, triplets of normals are considered. For each triplet, the vector that gives the same scalar product on the normals are computed (Step 14). The first such vector that allows all active constraints to reduce is then taken as $\Delta\mathbf{x}$ (Step 16)—if no such vector is available, the overall algorithm terminates.

Donné et al. showed that finding $\Delta\mathbf{x}$ in the manner above guarantees that $\Delta\mathbf{x}$ represents a direction from $\hat{\mathbf{x}}$ along which the active residuals (which are the median residuals in our case) decrease in value. Sec. 3.3 will establish that DESCENTDIR always find a descent direction until the convergence of Q-sweep to a local minimum.

Number of active constraints The cost of Algorithm 2 depends on the number $|\mathcal{J}|$ of active constraints. Donné et al. cited empirical evidence to support that the number of active constraints is small (3 or 4). Actually, as we prove below, there is a theoretical limit on the number of active constraints—this is another contribution of our work.

Theorem 1. *For any feasible $\hat{\mathbf{x}}$, there are at most 8 constraints $r_{i,j}(\mathbf{x})$ such that $r_{i,j}(\hat{\mathbf{x}}) = \text{median}_i\ r_i(\hat{\mathbf{x}})$.*

Proof. The combinatorial dimension of quasiconvex ℓ_∞ triangulation is 4, and assuming that the input data is non-degenerate[2], the number of active residuals (i.e., the cardinality of $\mathcal{A}$ in Algorithm 2) is at most 4 [26].

From (6), in each residual $r_i(\mathbf{x})$, the operands $r_{i,1}(\mathbf{x})$ and $r_{i,2}(\mathbf{x})$ are symmetric, such that for any $\hat{r} > 0$,

$$r_{i,1}(\hat{\mathbf{x}}) = \hat{r} \quad \text{and} \quad r_{i,2}(\hat{\mathbf{x}}) = \hat{r} \tag{12}$$

cannot be satisfied simultaneously. Likewise for $r_{i,3}(\mathbf{x})$ and $r_{i,4}(\mathbf{x})$. Thus, for each active residual $r_p(\mathbf{x})$, there are at most two operands that satisfy $r_{p,q}(\hat{\mathbf{x}}) = \hat{r}$. The total number of active constraints thus cannot be more than 8. □

[2]For the precise definition of degeneracy, see [20, Sec. 2.2]. In practical instances that are affected by noise, the data is usually non-degenerate.

3.2. Computing step size

Henceforth represents a significant departure from polyhedron collapse. From $\hat{\mathbf{x}}$, the new estimate is obtained as

$$\mathbf{x}' = \hat{\mathbf{x}} + \alpha\Delta\mathbf{x}. \quad (13)$$

Naturally, $\mathbf{x}'$ must remain feasible, but we would also like to find the α that reduces the median error the most.

Along $\Delta\mathbf{x}$ and starting from $\hat{\mathbf{x}}$, the constraints can be rewritten as a function of α:

$$r_{i,j}(\alpha) = \frac{\mathbf{a}_{i,j}^T(\hat{\mathbf{x}} + \alpha\Delta\mathbf{x}) + b_{i,j}}{\mathbf{c}_i^T(\hat{\mathbf{x}} + \alpha\Delta\mathbf{x}) + d_i} := \frac{u_{i,j}\alpha + v_{i,j}}{w_i\alpha + z_i}, \quad (14)$$

where $u_{i,j}$, $v_{i,j}$, w_i and z_i are constants calculated from the data; $r_{i,j}(\alpha)$ is again a linear fractional function, which is quasiconvex [2]. Trivially, the reprojection error $r_i(\mathbf{x})$ along direction $\Delta\mathbf{x}$ and starting from $\hat{\mathbf{x}}$ is

$$r_i(\alpha) = \max(r_{i,1}(\alpha), r_{i,2}(\alpha), r_{i,3}(\alpha), r_{i,4}(\alpha)), \quad (15)$$

with the usual condition $w_i\alpha + z_i > 0$ on the denominator. Since each of the $\max$ operands in (15) is quasiconvex, $r_i(\alpha)$ is quasiconvex; Fig. 1(a) illustrates.

The problem of determining α can be formulated as

$$\begin{aligned} \alpha^* = &\operatorname*{argmin}_{\alpha\in\mathbb{R}_+} \quad \operatorname*{median}_{i\in\{1,\dots,N\}} r_i(\alpha), \\ &\text{s.t.} \qquad w_i\alpha + z_i > 0 \;\; \forall i \in \{1,\dots,N\}, \end{aligned} \quad (16)$$

i.e., a quasiconvex LMS problem defined over α. Solving (16) exactly to find α^* remains *theoretically* intractable. Nonetheless, since we are dealing with only one dimension, there are "tricks" to do this efficiently.

Characterization of the solution Where can we expect α^* to lie? We first define several geometrical concepts.

Definition 1 (Extremity)**.** *The extremity m_i of $r_i(\alpha)$ is the point at which $r_i(\alpha)$ attains the minimum. Since $r_i(\alpha)$ is a linear fractional function, it is actually pseudoconvex (a stronger condition than quasiconvexity) [2], implying that m_i is unique; see Fig. 1(a). The extremity can be obtained analytically by intersecting all the constraints*

$$\frac{u_{i,j}\alpha + v_{i,j}}{w_i\alpha + z_i} = \frac{u_{i,j'}\alpha + v_{i,j'}}{w_i\alpha + z_i}, \;\; j, j' \in \{1,\dots,4\} \quad (17)$$

from $r_i(\alpha)$, and finding the roots. The smallest root that is not below $r_i(\alpha)$ is then installed as m_i.

Definition 2 (Intersection)**.** *An intersection between $r_i(\alpha)$ and $r_{i'}(\alpha)$ is a point where the two error functions intersect. Note that for quasiconvex functions, there are in general more than one intersection. We let $I_{i,i'}^k$ denote the k-th intersection between $r_i(\alpha)$ and $r_{i'}(\alpha)$. The intersections can also be found analytically, by pairing the constraints from $r_i(\alpha)$ and $r_{i'}(\alpha)$, and solving the quadratic equations.*

Definition 3 (Boundary)**.** *The boundary $\alpha_{\max}$ is the largest α such that $w_i\alpha + z_i > 0$ for all i.*

Definition 4 (Events)**.** *The events are a set that consists of*

- *all extremities m_i in the range $[0, \alpha_{\max}]$.*
- *all intersections $I_{i,i'}^k$ in the range $[0, \alpha_{\max}]$.*
- *the boundary point $(\alpha_{\max}, \operatorname{median}_i \; r_i(\alpha_{\max}))$.*

We call an item of $\mathcal{E}$ an event point.

The following identifies the possible locations of α^*.

Theorem 2. *The minimizer α^* of problem* (16) *is an event point of the problem.*

Proof. In the feasible range $[0, \alpha_{\max}]$, let

$$g : [0, \alpha_{\max}] \mapsto \{1, \dots, N\} \quad (18)$$

give the index of the error corresponding to the median, i.e.,

$$r_{g(\alpha)}(\alpha) = \operatorname*{median}_i \;\; r_i(\alpha). \quad (19)$$

The function g partitions the feasible range $[0, \alpha_{\max}]$ into segments $(s_1, s_2, \dots)$, where α's from the same segment s_t yield the same index, i.e.,

$$g(\alpha_1) = g(\alpha_2) \;\; \text{for} \;\; \alpha_1, \alpha_2 \in s_t; \quad (20)$$

see Fig. 1(b). In turn, the segments give rise to a sequence of error functions $(r_{g_1}, r_{g_2}, \dots)$ that correspond to the median. It is thus sufficient to examine this sequence.

If an error function $r_{g_t}(\alpha)$ in the sequence achieves its minimum (or extremity) m_{g_t} in the segment s_t, then m_{g_t} is a local minimum of the median error; see Fig. 1(b).

Any two successive error functions $r_{g_t}(\alpha)$ and $r_{g_u}(\alpha)$ in the sequence give rise to an intersection I_{g_t,g_u}^k. If the gradient of $r_{g_t}(\alpha)$ and $r_{g_u}(\alpha)$ have opposing signs at I_{g_t,g_u}^k, then I_{g_t,g_u}^k is a local minimum; see Fig. 1(b).

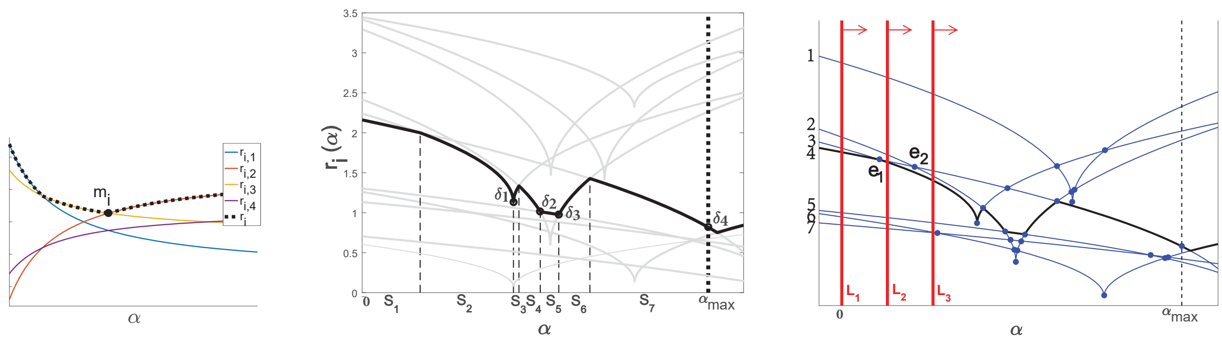
Lastly, the median error may achieve a local minimum at the boundary point; see Fig. 1(b).

The above are all the possible local minima of (16), and one of them is the global minimum. □

Based on Theorem 2, a simple approach to solve (16) would be to visit all event points, and calculate the median error at each event point. In the following, a more efficient technique that avoids recomputing the median is described.

Quasiconvex plane sweep Plane sweep is a basic technique for many geometric problems, such as Delaunay triangulation [8]. Souvaine and Steele [27] developed an LMS *line fitting* algorithm based on plane sweep. However, their method is not directly applicable to quasiconvex LMS (16), due to several critical differences:

- A pair of quasiconvex curves may have multiple intersections (see the curves in Fig. 1(b)), while two lines have at most one intersection;



(a) (b) (c)

Figure 1. (Panel a) Reprojection error as a function of α. The black dashed curve is $r_i(\alpha)$ and the other four curves are the four constraints $r_{i,j}(\alpha)$ corresponding to $r_i(\alpha)$; m_i is the extremity of $r_i(\alpha)$. (Panel b) The black solid curve is the median over 7 reprojection errors $r_i(\alpha)$. Within each segment s_t, the median is defined by one of the error functions: $r_{g_t}(\alpha)$. Here, δ_1, δ_3 and δ_4 are local minima corresponding respectively to an extremity, an intersection, and the boundary. δ_2 is an intersection, but not a local minimum. (Panel c) Demonstrating plane sweep: the events are shown as dots. The sweep line is initialized at L_1, with the ordering **List** $= [1, 2, 3, 4, 5, 6, 7]$; the center item is 4, thus $r_4(\alpha)$ is the median. As the sweep line passes through event point e_1, indices 3 and 4 swap places; at L_2, the ordering becomes **List** $= [1, 2, 4, 3, 5, 6, 7]$, and $r_3(\alpha)$ is the median. After passing through e_2, **List** $= [1, 4, 2, 3, 5, 6, 7]$, and $r_3(\alpha)$ remains the median.

- At an intersection, two quasiconvex curves may not cross (i.e., they are tangent to each other), while two lines necessarily cross at their intersection
- A quasiconvex function may achieve a minimum in the feasible range, while a line is unbounded.

Here, we develop a novel plane sweep algorithm for (16), to be used as routine `STEPSIZE` in Q-sweep.

The idea is as follows: imagine a vertical line L in the plane $[0, \alpha_{\max}] \times \mathbb{R}_{\geq 0}$ that is "swept" from $\alpha = 0$ to $\alpha = \alpha_{\max}$; see Fig. 1(c). At each position of L, the error functions $r_i(\alpha)$ can be ordered based on their height along L; the ordering is called the **List**. The median error is exactly the height of the median point in L. In plane sweep, we visit the event points incrementally and query **List**.

A crucial observation is that **List** changes only when L passes through an event point that is a non-tangential intersection. In fact, when sweeping past an non-tangential intersection $I^k_{i,i'}$, only $r_i(\alpha)$ and $r_{i'}(\alpha)$ swap orders along L; see Fig. 1(c). Thus, **List** can be maintained and updated efficiently as the sweep line passes through the event points.

Algorithm 3 describes the proposed quasiconvex plane sweep algorithm in detail. Programmatically, an event point $e \in \mathcal{E}$ is endowed with attributes α, i and i', where

- $e.\alpha$ is the α value of the event point e.
- if e is an extremity, $e.i$ returns the index of the error $r_i(\alpha)$ that gives rise to e (here, $e.i'$ is null);
- if e is an intersection, $e.i$ and $e.i'$ are the indices of the errors $r_i(\alpha)$ and $r_{i'}(\alpha)$ whose intersection forms e.

Note that, as in most plane sweep-type algorithms [8], the sweep line L is not explicitly realized.

Complexity analysis The runtime of Algorithm 3 depends on the size of $\mathcal{E}$. Since there are at most N extremities, $\mathcal{E}$ is dominated by the intersections. The following establishes a bound on the number of intersections.

Algorithm 3 `STEPSIZE` to optimize step size.

Require: Input data $\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$, current estimate $\hat{\mathbf{x}}$, and current descent direction $\Delta\mathbf{x}$.
Convert reprojection errors to 1D version (15).
$\alpha^* \leftarrow 0$. /* Current estimate of α */
$\mathcal{E} \leftarrow$ set of event points sorted ascendingly by $\{e.\alpha\}$.
List $\leftarrow$ indices of error functions $r_i(\alpha)$, $i = 1, \ldots, N$, in descending order of error value $r_i(\alpha^*)$ at α^*.
$K \leftarrow \lceil N/2 \rceil$.
$r^* \leftarrow r_{\mathbf{List}(K)}(\alpha^*)$. /* Current median error */
for each event point $e \in \mathcal{E}$ **do**
  if e is an intersection **then**
    if $r_{e.i}(\alpha)$ and $r_{e.i'}(\alpha)$ are not tangent at e **then**
      Swap the order of $e.i$ and $e.i'$ in **List**.
    end if
  end if
  if $e.i =$ **List**(K) or $e.i' =$ **List**(K) **then**
    if $r^* > r_{e.i}(e.\alpha)$ **then**
      $\alpha^* \leftarrow e.\alpha$, $r^* \leftarrow r_{e.i}(e.\alpha)$. /* Update */
    end if
  end if
end for
return α^*.

Lemma 1. *The number of intersections of all errors $r_i(\alpha)$, $i = 1, \ldots, N$, is bounded above by $16N^2 - 16N$.*

Proof. Let $r_i(\alpha)$ and $r_{i'}(\alpha)$ be two reprojection errors. The intersection of 2 constraints, respectively from $r_i(\alpha)$ and $r_{i'}(\alpha)$, gives rise to a quadratic function with at most 2 intersections. Thus the number of intersections between $r_i(\alpha)$ and $r_{i'}(\alpha)$ is limited to 32. For all $\binom{N}{2}$ pairs of reprojection errors, the total number of intersections is bounded above by $32N(N-1)/2 = 16N^2 - 16N$. $\square$

By maintaining **List** in a binary heap equipped with an auxiliary pointer array [27], looking up the median error (Step 14) and conducting swapping (Step 10) can be done in constant time at each event point—the sweep thus consumes $\mathcal{O}(N^2)$ time. The cost of Algorithm 3 is thus dominated by the sorting of $\mathcal{E}$ (Step 3), which needs $\mathcal{O}(N^2 \log N)$ time.

3.3. Convergence of Q-sweep to local minimum

In Secs. 3.1 and 3.2 we have described the details of subroutines `DESCENTDIR` and `STEPSIZE` in Q-sweep (Algorithm 1). Here, we establish the convergence of Q-sweep to a local minimum of LMS triangulation (9).

Theorem 3. *Q-sweep (Algorithm 1) converges to a local minimum of problem* (9).

Proof. Without loss of generality, define $K = \lceil N/2 \rceil$ as the median index. Given the current estimate $\hat{\mathbf{x}}$, let

$$r_{(1)}(\hat{\mathbf{x}}), \dots, r_{(K)}(\hat{\mathbf{x}}), \dots, r_{(N)}(\hat{\mathbf{x}}), \tag{21}$$

be the ordered residuals, where $r_{(p)}(\hat{\mathbf{x}}) \leq r_{(q)}(\hat{\mathbf{x}})\ \forall p < q$.

Algorithm 1 terminates when the $\Delta\mathbf{x}$ returned by Algorithm 2 is null. By [9, Supp. material], $\hat{\mathbf{x}}$ is thus the global minimizer to the ℓ_∞ triangulation problem defined by the subset of data indexed by $\{(1), \dots, (K)\}$, i.e.,

$$\begin{aligned} \hat{\mathbf{x}} = & \arg\min_{\mathbf{x}\in\mathbb{R}^3} \quad \underset{i\in\{(1),\dots,(K)\}}{\text{maximum}}\ r_i(\mathbf{x}) \\ & \text{s.t.} \qquad \mathbf{c}_i^T\mathbf{x} + d_i > 0\ \ \forall i \in \{(1), \dots, (K)\}. \end{aligned} \tag{22}$$

By assuming non-degeneracy (see proof of Theorem 1), there is an open subset $\mathcal{X}$ of $\mathbb{R}^3$ containing $\hat{\mathbf{x}}$ such that

$$r_{(K+1)}(\mathbf{x}) > r_{(i)}(\mathbf{x}),\ \ i = 1\dots,K,\ \ \forall \mathbf{x} \in \mathcal{X}, \tag{23}$$

i.e., $r_{(K+1)}(\mathbf{x})$ is always the $(K+1)$-th largest residual in $\mathcal{X}$. Thus, when the stopping criterion is achieved, Algorithm 1 terminates at a local minimum of the median error.

For $\mathbf{x}$ such that $\mathbf{c}_i^T\mathbf{x} + d_i > 0$, $r_i(\mathbf{x})$ is bounded below by 0. Thus, $\text{median}_i\ r_i(\mathbf{x})$ is also bounded below by 0 in the feasible region. Since each iteration of Q-sweep follows a descent direction and guarantees reduction in the median error, the algorithm converges to a local minimum. $\square$

4. Results

We compared Q-sweep against the following methods for the triangulation problem:

- Polyhedron collapse [9], which solves (1) and is thus non-robust—we regard it as the **control method**;
- Random sampling heuristic [1] with confidence 0.99 and outlier rate 0.5 for the stopping criterion;
- Ke & Kanade's approximate algorithm for LMS [16];
- Li's globally optimal method [17];
- The proposed Q-sweep method (Algorithm 1); and
- Q-sweep method with brute force search to solve (16) for the step size, in place of plane sweep.

Since the originators' codes were not publicly available, we implemented polyhedron collapse ourselves in Matlab—this is sufficient since it is is a very efficient algorithm. For random sampling, DLT was used as the minimal solver. For Ke & Kanade, the feasibility test was solved using Matlab's LP solver. For Li's method, polyhedron collapse was used for basis computations. For Q-sweep, the plane sweep routine (Algorithm 3) was implemented in C-mex.

All experiments were conducted on a standard machine with a 3.6GHz Intel i7 CPU and 16GB RAM. Unavoidably, differences in implementation and programming languages will affect the relative runtimes of the above methods—in Sec. 4.1, we will factor out the effects of these differences by examining asymptotic runtime on synthetic data.

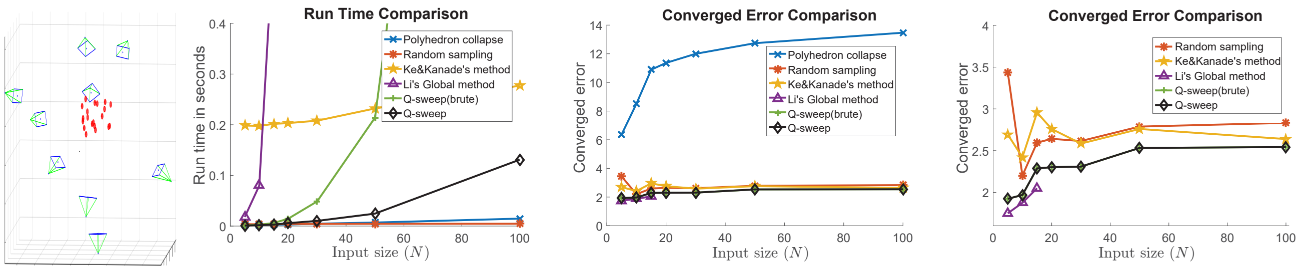
Details on initialization To initialize Ke & Kanade and Q-sweep, we used the mid-point method (a closed form solver) [13] on two randomly selected measurements to find the initial $\hat{\mathbf{x}}$, which was then tested for feasibility.

4.1. Synthetic data experiments

Synthetic datasets for triangulation were generated as follows: a dataset contained 20 random scene points in $\mathbb{R}^3$, and N cameras $\{\mathbf{P}_i\}_{i=1}^N$ created with random poses with the condition that the scene points lay in front of the cameras; see Fig. 2(a). A triangulation instance was formed by projecting a scene point onto the cameras, and adding Gaussian noise of $\sigma = 3$ pixels to the image points. To create outliers, 30% of the image points were randomly selected, and Gaussian noise of $\sigma = 9$ pixels was added to them.

Fig. 2(b) shows the runtime of all methods plotted against the size N of the outlier-contaminated triangulation instances (the runtime of each N was averaged over the 20 instances in the dataset). Expectedly, the runtime of the global method increased very rapidly. The cost of random sampling and polyhedron collapse (non-robust) remained more or less constant. Ke & Kanade and Q-sweep gave similar asymptotic behaviour—note, however, that Ke & Kanade does not guarantee local optimality, unlike Q-sweep. Finally, the runtime of Q-sweep with brute force step size search also grew rapidly, illustrating the significant computational savings due to plane sweep (Algorithm 3).

Figs. 2(c) and 2(d) show the converged reprojection error of all the triangulation methods. All LMS algorithms recover the noise level for inliers (3 pixels) while polyhedron collapse is affected by outliers (errors over 10 pixels). As is evident in Fig. 2(d), Q-sweep (and brute force variant) gave



(a) (b) (c) (d)

Figure 2. (a) A synthetic dataset with 20 scene points and $N = 9$ cameras. (b) Average runtime of all algorithms on synthetic triangulation instances plotted against input size N. (c) Average converged error for all methods. (d) Same as (c) but without polyhedron collapse.

	Temple ($\#p$=6927)		**Courtyard ($\#p$=59562)**		**University ($\#p$=19476)**		**Water Tower ($\#p$=58556)**	
Algorithm	Time (s)	OptErr	Time (s)	OptErr	Time (s)	OptErr	Time (s)	OptErr
Polyhedron collapse [9] (non-robust)	3.931	3.604	103.406	12.248	18.443	12.166	96.197	11.414
Random sampling (approx.)	8.927	0.935	109.660	1.906	26.797	4.564	101.905	3.151
Ke & Kanade [16] (approx.)	426.425	1.565	5257.441	4.072	1167.412	5.217	5084.582	4.131
Li [17] (global)	459.258	0.397	N/A	N/A	N/A	N/A	N/A	N/A
Q-sweep (locally optimal)	2.638	0.734	176.917	1.337	13.998	2.109	223.158	1.775
Q-sweep (brute force, locally optimal)	2.214	0.734	1338.820	1.337	24.586	2.109	2980.607	1.775

Table 1. Results on real datasets. $\#p$: total number of triangulation instances in the dataset. For information regarding the size N of the instances, see Panel (a) in Figs. 3 to 6. Time: *total* runtime for solving all triangulation instances (N/A if not finished by 2 hours); OptErr: the converged reprojection error, averaged over all instances, in pixels.

the lowest error among all approximate LMS algorithms, due to the ability of Q-sweep to converge to local minima.

4.2. Real data experiments

We used data from [10] (University of Washington, Alcatraz Courtyard, Alcatraz Water Tower) and [7] (Temple Ring). Olsson's SfM implementation [22] was used to estimate the camera poses and initial 3D structure (the input images were first resized to a factor 0.3). Then, SIFT [19, 29] was invoked on the original images to produce more feature correspondences, which were associated to form triangulation instances—these instances were contaminated by outliers, since they were not put through the SfM pipeline. The number of instances generated in this manner is shown as $\#p$ in Table 1, whereas Panel (a) in Figs. 3 to 6 plots the histogram of the sizes N of the instances—though most of the instances were small, a non-negligible number of them were of moderate to large sizes.

Table 1 summarizes the total runtime and average converged reprojection error for all methods.

Accuracy comparison On the smallest dataset (Temple), the global method expectedly gave the lowest error, followed by Q-sweep. However, the global method was not feasible on the other larger datasets; it was terminated after reaching the time limit of 2 hours.

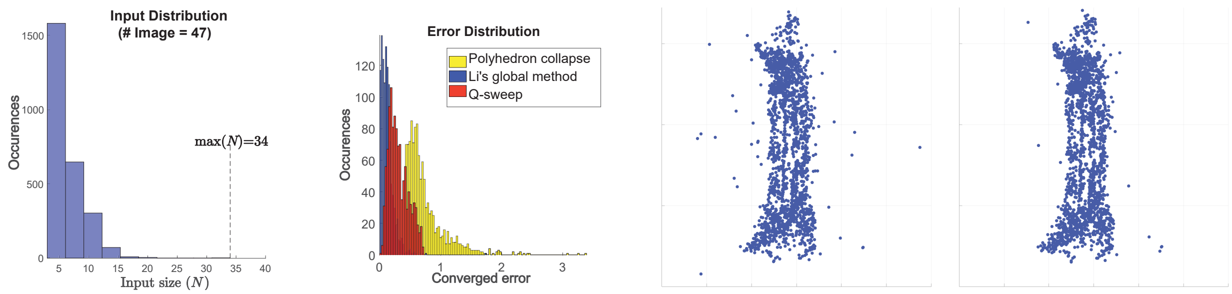
On the other datasets, Q-sweep gave the lowest error, due to its ability to converge to local minima. This was followed by random sampling, Ke & Kanade, and polyhedron collapse. In Courtyard, University and Water Tower, the average converged error of polyhedron collapse was around 10 pixels, indicating the presence of outliers.

Panel (b) in Figs. 3 to 6 plots the histogram of converged errors for Q-sweep and polyhedron collapse. Evidently a large number of instances contain outliers, and the benefit of LMS triangulation with Q-sweep is clearly exhibited. Panels (c) and (d) in the figures show the triangulated points from both methods. Observe that there are much fewer spurious points in the results of Q-sweep.

Runtime comparison The recorded runtimes comply with the trends observed in the synthetic data experiments. We note again that step size search in Q-sweep with brute force was much more expensive than plane sweep.

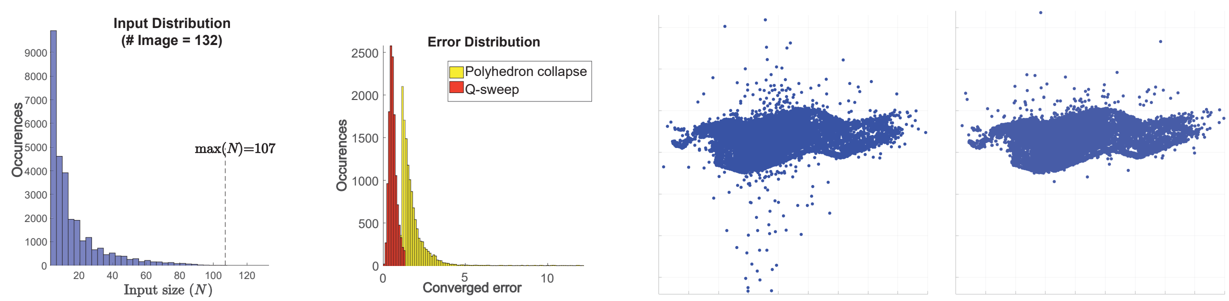
5. Conclusions

Robust triangulation is a vital computer vision problem that has not been satisfactorily solved. The proposed Q-sweep algorithm fills a gap in currently available techniques for LMS triangulation. Unlike random sampling, it guarantees convergence to local minima, thus giving higher quality outcomes. Unlike global methods, Q-sweep is much more efficient and practical. At a higher level, our work illustrates useful adaptation of geometric algorithms to vision.



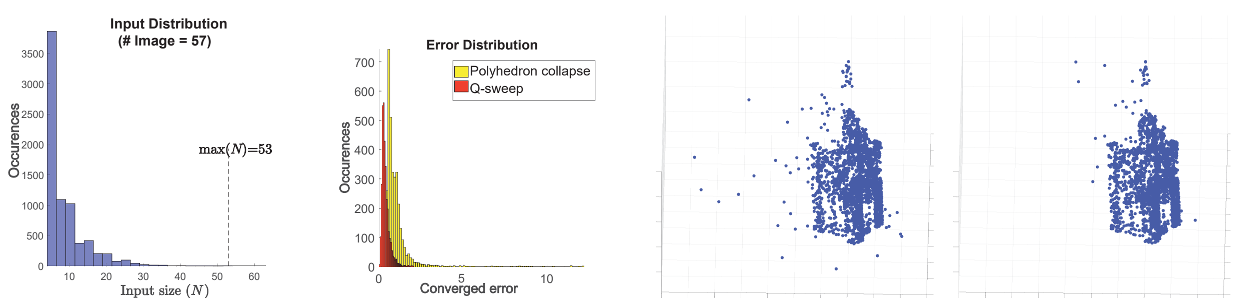
(a) (b) (c) (d)

Figure 3. Results for **Temple Ring** dataset. (a) Distribution of instance sizes N. (b) Histogram of converged reprojection errors for polyhedron collapse, global method, and Q-sweep. (c)(d) 3D structure reconstructed respectively by polyhedron collapse and Q-sweep.



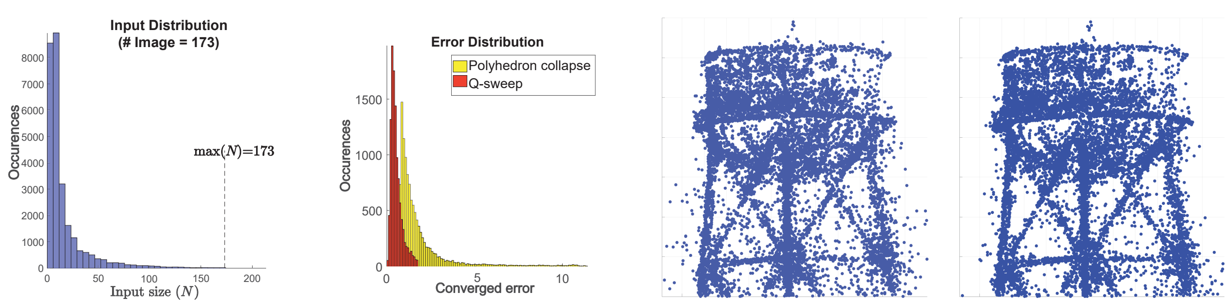
(a) (b) (c) (d)

Figure 4. Results for **Alcatraz Courtyard** dataset. (a) Distribution of instance sizes N. (b) Histogram of converged reprojection errors for polyhedron collapse and Q-sweep. (c)(d) 3D structure reconstructed respectively by polyhedron collapse and Q-sweep.



(a) (b) (c) (d)

Figure 5. Results for **University of Washington** dataset. (a) Distribution of instance sizes N. (b) Histogram of converged reprojection errors for polyhedron collapse and Q-sweep. (c)(d) 3D structure reconstructed respectively by polyhedron collapse and Q-sweep.



(a) (b) (c) (d)

Figure 6. Results for **Alcatraz Water Tower** dataset. (a) Distribution of instance sizes N. (b) Histogram of converged reprojection errors for polyhedron collapse and Q-sweep. (c)(d) 3D structure reconstructed respectively by polyhedron collapse and Q-sweep.

References

[1] https://en.wikipedia.org/wiki/Random_sample_consensus.

[2] https://en.wikipedia.org/wiki/Linear-fractional_programming.

[3] S. Agarwal, N. Snavely, and S. M. Seitz. Fast algorithms for L_∞ problems in multiview geometry. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[4] J. Agulló. Exact algorithms for computing the least median of squares estimate in multiple linear regression. *Lecture Notes-Monograph Series*, pages 133–146, 1997.

[5] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter. Efficient globally optimal consensus maximisation with tree search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2413–2421, 2015.

[6] S. Choi, T. Kim, and W. Yu. Performance evaluation of ransac family. *Journal of Computer Vision*, 24(3):271–300, 1997.

[7] T. M. C. computer vision. Multi-view stereo dataset. http://vision.middlebury.edu/mview/data/.

[8] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf. Computational geometry. In *Computational geometry*, pages 1–17. Springer, 2000.

[9] S. Donné, B. Goossens, and W. Philips. Point triangulation through polyhedron collapse using the l infinity norm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 792–800, 2015.

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

[11] 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*, pages 4066–4073. IEEE, 2014.

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[14] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[15] F. Kahl. Multiple view geometry and the L_∞-norm. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1002–1009. IEEE, 2005.

[16] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007.

[17] H. Li. A practical algorithm for L_∞ triangulation with outliers. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[18] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1074–1080. IEEE, 2009.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[20] J. Matoušek. On geometric optimization with few violated constraints. *Discrete & Computational Geometry*, 14(4):365–384, 1995.

[21] J. Nocedal and S. J. Wright. *Numerical optimization, second edition*. Springer, 2006.

[22] C. Olsson. Stable structure from motion for unordered image collections. http://www.maths.lth.se/matematiklth/personal/calle/sys_paper/sys_paper.html.

[23] C. Olsson, A. P. Eriksson, and F. Kahl. Efficient optimization for L_∞-problems using pseudoconvexity. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[24] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[25] Y. Seo and R. Hartley. A fast method to minimize L_∞ error norm for geometric vision problems. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[26] K. Sim and R. Hartley. Removing outliers using the L_∞ norm. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 485–494. IEEE, 2006.

[27] D. L. Souvaine and J. M. Steele. Time-and space-efficient algorithms for least median of squares regression. *Journal of the American Statistical Association*, 82(399):794–801, 1987.

[28] A. J. Stromberg. Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression. *SIAM Journal on Scientific Computing*, 14(6):1289–1299, 1993.

[29] A. Vedaldi. An open implementation of the sift detector and descriptor. *UCLA CSD*, 2007.

[30] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.