Quasiconvex Optimization for Robust Geometric Reconstruction

Qifa Ke and Takeo Kanade

CMU-RI-TR-05-57

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: Geometric reconstruction, quasiconvex, convex feasibility, linear programming, second-order cone programming, optimization, reprojection error, outlier, robust

Abstract

Geometric reconstruction problems in computer vision are often solved by minimizing a cost function that combines the reprojection errors in the 2D images. In this paper, we show that, for various geometric reconstruction problems, their reprojection error functions share a *common* and *quasiconvex* formulation. Based on the quasiconvexity, we present a novel quasiconvex optimization framework in which the geometric reconstruction problems are formulated as a small number of small-scale convex programs that are ready to solve. Our final reconstruction algorithm is simple and has intuitive geometric interpretation. In contrast to existing random sampling or local minimization approaches, our algorithm is deterministic and guarantees a predefined accuracy of the minimization result. Moreover, the quasiconvexity provides an intuitive method to handle directional uncertainties and outliers in measurements. We demonstrate the effectiveness of our algorithm by experiments on both synthetic and real data.

1 Introduction

Given measurements in 2D images, the goal of geometric reconstruction in computer vision is to estimate the three-dimensional information about the scene and the camera motions. Classical examples include triangulation [11], camera resectioning [4, 10], and structure from motion (see [7] for a review). The Gold standard for these estimation problems is minimizing F_s, the *average* of squared reprojection errors (model-fitting errors measured in 2D image domain). Minimizing F_s leads to maximum likelihood estimation when measurement noises follow Gaussian distribution.

Due to the camera perspective effect, the cost function F_s is highly nonlinear and often contains multiple local minima. Minimizing F_s is therefore difficult. Hartley and Schaffalitzky [6] proposed using the *pointwise maximum* of the squared reprojection errors as the cost function, which we denote as F_∞. In contrast to F_s, it was shown that F_∞ contains only one single minimum value in its feasible domain. An approach using random line search in the parameter space was used in [6] to minimize F_∞. The convergence behavior of random line search remains unclear. As pointed out in [6], it is difficult to perform random line search when the parameter space is high dimensional. Constrained minimization is also proposed in [6] for minimizing F_∞. However, the constraints are nonlinear and nonconvex, making such constrained minimization a difficult problem by itself.

We can consider the model-fitting error as a function of the unknown parameters, which is termed *reprojection error function* in this paper. We show that the reprojection error functions share a *common* and *quasiconvex* formulation for the geometric reconstruction problems under our consideration. As a result, F_∞, the pointwise maximum of a family of quasiconvex functions, is also a quasiconvex function. We then present an one-dimensional bisection algorithm to minimize the quasiconvex function F_∞. Our algorithm consists of a small number of small-scale convex programs, specifically linear programs (LP) or second-order cone programs (SOCP). Both LP and SOCP are well-studied and existing efficient algorithms and implementations are ready to use. Compared to random line search in parameter space or local minimization approaches, our minimization approach is efficient, even when the unknowns are high dimensional. More importantly, our approach is deterministic and guarantees a predefined accuracy of the minimization result.

It has been pointed out in [6] that F_∞ is sensitive to outliers. We present two approaches to handling outliers. In the first approach, we use F_m, the pointwise m-th smallest reprojection error, as the cost function. In contrast to F_∞ or F_s, the cost function F_m is highly robust to outliers [18]. In spite of its complex formulation, in our cases F_m is still a pointwise operator of a family of quasiconvex functions. As a result, our algorithm to minimize F_∞ can be extended to efficiently minimize F_m, again by solving small-scale convex programs (LP or

SOCP). In the second approach, we show that the active constraints in the convex programs must contain outliers (if they exist). As a result, we can remove outliers by simply removing active constraints.

1.1 Background: geometric reconstruction problems

We present four classical examples of geometric reconstruction problems in computer vision.

1.1.1 Multi-view triangulation

We are given projection matrices of N cameras, denoted by $\{\mathtt{P}_i, i = 1, ..., N\}$, and the images of the unknown 3D point $\mathbf{Z}$ in these N cameras, denoted by $\{\mathbf{x}_i, i = 1, ..., N\}$. The task of triangulation is to estimate $\mathbf{Z}$ from $\{\mathtt{P}_i\}$ and $\{\mathbf{x}_i\}$. Triangulation is a necessary step in two- or multi-view 3D reconstruction, and in structure from motion.

Note that optimal triangulation algorithms [11, 9] for two-view case are not generalizable to multi-view case.

1.1.2 Camera resectioning

We are given 3D points $\{\mathbf{Z}_i, i = 1, ..., N\}$ and their images $\{\mathbf{x}_i, i = 1, ..., N\}$ in one camera. The task is to estimate the camera projection matrix $\mathtt{P}$ from these N corresponding pairs $\{\mathbf{x}_i \leftrightarrow \mathbf{Z}_i\}$. Camera resectioning is used in camera calibration and in structure from motion.

1.1.3 Multi-view reconstruction with known rotations

In some cases the camera rotations are known, leaving only the camera positions and the 3D of the scene to be estimated [6]. For example, in vision-aided inertial navigation, accurate camera pose is available from modern gyroscopes, while the camera position information from accelerometers is still noisy [3]. Another example is that there are reconstruction methods in which the camera rotation for each frame is estimated in a first step [17].

Denote the N intrinsically calibrated cameras as $\{\mathtt{P}_i = (\mathtt{R}_i, -\mathtt{R}_i\mathbf{C}_i), i = 1, ..., N\}$, where for each camera the rotation $\mathtt{R}_i$ is known, but its 3D position $\mathbf{C}_i$ is unknown. We are given 2D feature points $\{\mathbf{x}_{ij}\}$ over the N cameras. Here $\mathbf{x}_{ij}$ denotes the projection of j-th 3D point $\mathbf{Z}_j$ onto the i-th camera. The task is to estimate $\{\mathbf{Z}_j\}$ and $\{\mathbf{C}_i\}$ from the 2D points $\{\mathbf{x}_{ij}\}$ and the camera poses $\{\mathtt{R}_i\}$.

1.1.4 Multi-view reconstruction using a reference plane

When a reference plane is visible in all images, the inter-image planar homography from the given reference plane can be used to compensate the relative camera rotations. Then the problem essentially reduces to the above case of reconstruction with known rotations.

1.1.5 Planar homography estimation

Two images of points on a 3D scene plane are related by a planar homography $\mathtt{H}$, a 3×3 non-singular matrix. Given N correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i, i = 1, ..., N\}$, the task is to estimate $\mathtt{H}$ such that $\mathbf{x}'_i = \mathtt{H}\mathbf{x}$.

2 The cost function

In this section, we define the reconstruction error metric at each individual 2D measurement, and the cost functions that combine reconstruction errors from individual 2D measurements.

2.1 Error metric for one 2D measurment

We use Triangulation as an example to illustrate three often-used error metrics for an individual 2D measurement.

2.1.1 Algebraic distance

Denote $\tilde{\mathbf{x}}_i = (\mathbf{x}_i; 1)$ the homogeneous coordinates of the 2D measurement $\mathbf{x}_i$, we have the following linear equation:

$$k_i \tilde{\mathbf{x}}_i = \mathtt{P}_i \mathbf{Z} \tag{1}$$

Here $\mathbf{Z}$ is also expressed in homogeneous coordinates. The algebraic distance for $\mathbf{x}_i$ is then defined by:

$$f_i(\mathbf{Z}) = \|k_i \tilde{\mathbf{x}}_i - \mathtt{P}_i \mathbf{Z}\|_2 \tag{2}$$

Linear least-squares can be applied to estimate $\mathbf{Z}$ by minimizing the sum of squared algebraic distances. Since the algebraic distance is not geometrically or statistically meaningful, the algebraic reconstruction is not reliable (see [18, 6]).

2.1.2 Distance in 3D space

In the case of calibrated cameras, $f_i(\mathbf{Z})$ can be defined as the distance from the 3D point $\mathbf{Z}$ to the ray back-projected from $\mathbf{x}_i$. In the case of two views, this distance function leads to

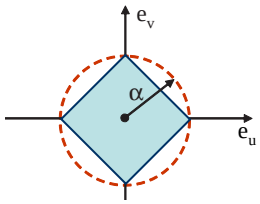


Figure 1: *Distance between* $\mathbf{x} = (u, v)$ *and* $\hat{\mathbf{x}} = (\hat{u}, \hat{v})$. *The solid square shows the contour on which the* L_1 *norm error* $e_1 = |e_u| + |e_v| = \alpha$, *while the dash line shows the contour on which the* L_2 *norm error* $e_2 = \sqrt{e_u^2 + e_v^2} = \alpha$. *Here* $e_u = (u - \hat{u})$, *and* $e_v = (v - \hat{v})$.

the midpoint estimation method where $\mathbf{Z}$ is given by midpoint of the perpendicular between the two rays.

When a camera is further away from the 3D point $\mathbf{Z}$, the camera has larger uncertainty on $\mathbf{Z}$. Distance metric in 3D space cannot take such uncertainty into account. As a result, the reconstruction result is unstable when the 3D point is far away from cameras.

2.1.3 Reprojection error in the image

The reprojection error is defined as the distance in the 2D image domain between $\mathbf{x}_i$ and its reprojection $\hat{\mathbf{x}}_i = \pi_i(\mathbf{Z})$:

$$f_i(\mathbf{Z}) = \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_l = \|\mathbf{x}_i - \pi_i(\mathbf{Z})\|_l \tag{3}$$

where $\hat{\mathbf{x}}_i = \pi_i(\mathbf{Z})$ is the reprojection of $\mathbf{Z}$ in the image of camera $\mathtt{P}_i$, and $\|\cdot\|_l$ denotes some vector norm. Both $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are in 2D Cartesian coordinates.

We choose reprojection error metric since it has a well-defined geometric meaning and it leads to maximum likelihood estimation. For example, when L_2 norm is used in Eq. (3), the reprojection error f_i is the *Euclidean distance* between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$. We can also use L_1 norm. Its geometric meaning is shown in Fig. 1.

2.2 Generalized reprojection error function

Definition 1. *The general formulation of reprojection error function:*

$$f(\mathbf{X}) = \frac{p(\mathbf{X})}{q(\mathbf{X})} \tag{4}$$

where

- $\mathbf{X} \in \mathbb{R}^n$ *is the unknown vector to be estimated;*
- $p(\mathbf{X})$ *is a convex function, and* $p(\mathbf{X}) \geq 0$.
- $q(\mathbf{X})$ *is a linear function, and* $q(\mathbf{X}) > 0$*;*

In the following we show that most reprojection error functions are special cases of the above general formulation. For a geometric reconstruction problem, if its reprojection error function conforms to the general formulation in Eq. (4), the algorithms we present in this paper can be applied to solve such reconstruction problem.

2.2.1 Reprojection error function in the image plane

Result 1. *For the reconstruction problems in Section 1.1, the reprojection error function defined in the image domain conforms to the general formulation in Definition 1.*

Proof. For the problems in Section 1.1, the reprojection of $\mathbf{x} = (u, v)$ in the image can be written as:

$$\hat{\mathbf{x}} = \left(\frac{\mathbf{a}^\top \mathbf{X}}{\mathbf{c}^\top \mathbf{X}}, \frac{\mathbf{b}^\top \mathbf{X}}{\mathbf{c}^\top \mathbf{X}} \right)^\top \tag{5}$$

Here $\mathbf{X}$ is the vector to be estimated. $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ are known vectors. For example, in the triangulation problem, they are the three rows of the camera matrix P, respectively.

The reprojection error function is:

$$f(\mathbf{X}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_l = \| \frac{1}{q(\mathbf{X})} \left(p_u(\mathbf{X}), p_v(\mathbf{X})\right) \|_l, \tag{6}$$

where $\| \cdot \|_l$ is the vector norm, and

$$\begin{aligned} p_u(\mathbf{X}) &= (u\mathbf{c}^\top - \mathbf{a}^\top)\mathbf{X}, \\ p_v(\mathbf{X}) &= (v\mathbf{c}^\top - \mathbf{b}^\top)\mathbf{X}, \\ q(\mathbf{X}) &= \mathbf{c}^\top \mathbf{X}. \end{aligned} \tag{7}$$

It is obvious that $q(\mathbf{X})$ is a linear function of $\mathbf{X}$.

In this paper, we consider affine or Euclidean reconstruction [1]. The cheirality constraint (see [7] and appendix), which states that the 3D points visible in the image must be in front of the camera, can then be expressed as $\mathbf{c}^\top \mathbf{X} > 0$. Therefore, we have $q(\mathbf{X}) > 0$. The reprojection error function in Eq. (6) can then be rewritten as:

$$f(\mathbf{X}) = \frac{1}{q(\mathbf{X})} \| \left(p_u(\mathbf{X}), p_v(\mathbf{X})\right) \|_l = \frac{p(\mathbf{X})}{q(\mathbf{X})} \tag{8}$$

Any norm function $g(\mathbf{y}) = \|\mathbf{y}\|_l$ is a convex function of $\mathbf{y}$. The function $h(\mathbf{X}) = (p_u(\mathbf{X}), p_v(\mathbf{X}))$ is an affine function of $\mathbf{X}$. The composition of a convex function g and an affine function h, denoted by $g \circ h$, is a convex function. Therefore, $p(\mathbf{X}) = (g \circ h)(\mathbf{X})$ is a convex function of $\mathbf{X}$. It is obvious that $p(\mathbf{X}) \geq 0$. □

[1] In a way similar to the method briefed in [6], our algorithm in this paper can be extended to projective reconstruction.

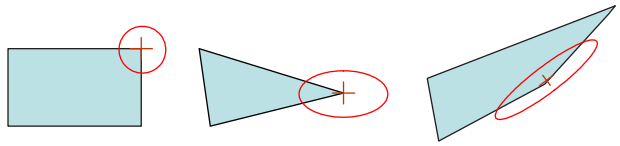


Figure 2: *Uncertainty in feature point locations. (a): Scalar uncertainty with covariance matrix* $\mathtt{Q} = diag(\sigma, \sigma)$*; (b): Directional but uncorrelated noises, with covariance matrix* $\mathtt{Q} = diag(\sigma_1, \sigma_2)$*; (c): Directional and correlated noises, with covariance matrix* $\mathtt{Q}$ *a full* 2×2 *matrix.*

2.2.2 Uncertainty-weighted reprojection error function

When uncertainty on the location of each 2D feature point is available, it can be shown that the uncertainty-weighted reprojection error function still conforms to the general formulation in Definition 1.

Uncertainty of feature position

The accuracy of feature matching depends on the intensity pattern around each feature, which often has strong directionality and is location-dependent. Such directional uncertainty can be characterized by the following inverse covariance matrix (cf. [12, 13]:

$$\mathtt{Q}^{-1} = \frac{1}{s} \sum_{(u,v) \in w} \begin{pmatrix} I_u I_u, & I_u I_v \\ I_u I_v, & I_v I_v \end{pmatrix}, \tag{9}$$

where w is a small window centered around the feature point in the image I of the i-th camera, s is determined by the intensity pattern inside w, and I_u and I_v are image gradients along u and v direction, respectively. A more accurate method to estimate feature position uncertainty is presented in [13], which takes into account not only the image pattern but also the image pixel noises.

Figure 2 shows the three different types feature uncertainty:

- $\mathtt{Q} = diag(\sigma, \sigma)$: scalar uncertainty that is feature-dependent, but is isotropic and therefore uncorrelated in u and v direction;
- $\mathtt{Q} = diag(\sigma_1, \sigma_2)$: directional ($\sigma_1 \neq \sigma_2$) but uncorrelated in u and v direction;
- $\mathtt{Q} = full\ 2 \times 2\ matrix$: directional and correlated in u and v direction.

Covariance-weighted reprojection error function

The uncertainty in the location of each 2D feature point $\mathbf{x}$ can be taken into account by weighting the reprojection error appropriately using the covariance matrix. The covariance

matrix $\mathtt{Q}$ for the 2D point $\mathbf{x}$ is a symmetric positive semi-definite matrix, and can be decomposed into the following form by the Singular Value Decomposition: $\mathtt{Q} = \mathtt{U}\Sigma\mathtt{U}^\top$, where $\Sigma = diag(\sigma_1, \sigma_2)$, and $\mathtt{U}$ is a 2×2 orthonormal matrix. The inverse covariance matrix takes the form of

$$\mathtt{Q}^{-1} = \mathtt{U}\Sigma^{-1}\mathtt{U}^\top \tag{10}$$

Denote $\mathtt{B} = \Sigma^{-1/2}\mathtt{U}^\top$, which is an affine transformation that transforms the input data into covariance-weighted data space where the noises at each feature become isotropic and $i.i.d.$. The transformed coordinates (in Euclidean) of $\mathbf{x}$ and $\hat{\mathbf{x}}$ in the image plane are :

$$\mathbf{x}' = (u', v')^\top = \mathtt{B}(u, v)^\top \tag{11}$$

$$\hat{\mathbf{x}}' = \mathtt{B}\hat{\mathbf{x}} = \frac{1}{\mathbf{c}^\top\mathbf{X}}\mathtt{B}\begin{pmatrix}\mathbf{a}^\top \\ \mathbf{b}^\top\end{pmatrix}\mathbf{X} \tag{12}$$

where $\mathbf{a}^\top$ and $\mathbf{b}^\top$ follow the notation in Eq. (5). The covariance matrix of the noise in the covariance-weighted data space now becomes isotropic and takes the form of $diag(1, 1)$. Denote

$$\mathtt{A} = \left[\mathtt{B}\begin{pmatrix}u \\ v\end{pmatrix}\mathbf{c}^\top - \mathtt{B}\begin{pmatrix}\mathbf{a}^\top \\ \mathbf{b}^\top\end{pmatrix}\right]. \tag{13}$$

The covariance-weighted reprojection error function is:

$$f^w(\mathbf{X}) = \|\mathbf{x}' - \hat{\mathbf{x}}'\| = \left\|\frac{\mathtt{A}\mathbf{X}}{\mathbf{c}^\top\mathbf{X}}\right\| \tag{14}$$

Again, the cheirality constraint [5], which states that the 3D points visible in the image must be in front of the camera, can then be expressed as $\mathbf{c}^\top\mathbf{X} > 0$. Therefore, Eq. (14) can then be written as:

$$f^w(\mathbf{X}) = \frac{1}{\mathbf{c}^\top\mathbf{X}}\|\mathtt{A}\mathbf{X}\| \tag{15}$$

The norm function $p(\mathbf{X}) = \|\mathtt{A}\mathbf{X}\|$ is convex, and the function in Eq. (15) conforms to the general formulation in Definition 1.

When $\|\cdot\|$ in Eq. (15) is L_2-norm, $f^w(\mathbf{X})$ is the Mahalanobis distance between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

2.2.3 Angular reprojection error function

When the camera is calibrated, the angle θ between the observed ray $\mathbf{x}$ and the reprojection ray $\mathbf{r} = (\mathbf{a}, \mathbf{b}, \mathbf{c})^\top\mathbf{X}$ can be used to define the reprojection error [9, 6]:

$$f(\mathbf{X}) = |\tan(\theta)| = \left|\frac{\mathbf{x} \times \mathbf{r}}{\mathbf{x}^\top\mathbf{r}}\right| \tag{16}$$

where $\times$ denotes cross-product. We choose $\tan(\theta)$ since it is a monotonically-increasing function of θ when $\theta \in [0, \pi/2)$. The cheirality constraint can be enforced by $|\theta| < \pi/2$,

which leads to $q(\mathbf{X}) = \mathbf{x}^\top \mathbf{r} > 0$. It is easy to verify that $q(\mathbf{X})$ is a linear function of $\mathbf{X}$, and $p(\mathbf{X}) = |\mathbf{x} \times \mathbf{r}|$ is convex in $\mathbf{X}$. Therefore, the angular reprojection error function $f(\mathbf{X}) = \frac{p(\mathbf{X})}{q(\mathbf{X})}$ conforms to the general form in Definition 1.

2.3 Combining reprojection errors into cost function

The often used cost function F_s in geometric reconstruction is defined as the average of the squared L_2-norm reprojection errors:

$$F_s = \frac{1}{M} \sum_i f_i^2(\mathbf{X}) \tag{17}$$

where M is the total number of 2D measurements (points). F_s is difficult to minimize as it is highly nonlinear and contains multiple local minima [6].

Hartley and Schaffalitzky [6] proposed using the pointwise maximum of the reprojection errors as the cost function:

$$F_\infty(\mathbf{X}) = \max_i f_i(\mathbf{X}) \tag{18}$$

When the measurement uncertainties are available, the uncertainty-weighted cost function is denoted by:

$$F_\infty^w(\mathbf{X}) = \max_i f_i^w(\mathbf{X}) \tag{19}$$

It was shown in [6] that $F_\infty(\mathbf{X})$ contains only one single minimum value in its domain, and is therefore easier to minimize than $F_s(\mathbf{X})$. But as is also pointed out in [6], $F_\infty(\mathbf{X})$ is sensitive to outliers.

To deal with the outliers, we propose using the *pointwise* m-th smallest reprojection errors as the cost function:

$$F_m(\mathbf{X}) = \operatorname*{mth}_i f_i(\mathbf{X}) \tag{20}$$

It is obvious that F_∞ is a special case of F_m when $m = N$. F_m is a highly robust function. For example, when $m = \lfloor N/2 \rfloor$, it is the median operator. Minimizing F_m leads to least-median optimization [18], which can handle noisy measurements with up to 50% of outliers.

3 Minimizing the cost function

Both F_∞ and F_m are constructed from pointwise operations on a family of functions. They are not differentiable at many points. As a result, classical gradient-based approaches are not applicable to minimizing them. Random line search in the parameter space was proposed in [6] to minimize F_∞, and random sampling [18, 15] is often used to detect outliers and

to minimize F_m. These randomized approaches are not scalable when the unknowns are high-dimensional. They do not guarantee convergence either.

In this section, we show that the general reprojection error function (Definition 1) is quasiconvex. Such quasiconvexity enables us to design a deterministic and efficient algorithm to minimizing F_∞ and F_m.

3.1 Minimization by feasibility

Instead of random search or sampling, let us look at a minimization approach that uses the classic bisection search in the range domain of F_∞ and F_m.

For the vision problems in which we are interested, the image size is bounded. Therefore, it is realistic to assume that $l \leq F(\mathbf{X}) \leq h$, where $F(\mathbf{X})$ is the cost function. For $\alpha \in [l, h]$, denote S_α the α-sublevel set of $F(\mathbf{X})$:

$$S_\alpha = \{\mathbf{X} \mid F(\mathbf{X}) \leq \alpha\} \tag{21}$$

If S_α is non-empty, then we know that F^*, the minimum value of $F(\mathbf{X})$, satisfies $F^* \leq \alpha$. Otherwise, we have $F^* > \alpha$. Determining whether S_α is empty or not can be achieved by solving the following feasibility problem:

$$\begin{aligned} &\text{find } \mathbf{X} \\ &s.t.\ \mathbf{X} \in S_\alpha \end{aligned} \tag{22}$$

Based on the above observation, we can use the bisection algorithm (see [2]) to pin down the optimal value of $F(\mathbf{X})$ by solving a sequence of feasibility problems. Fig. 3 shows the basic procedure of the algorithm. It starts with a range $[l, h]$ that is known to contain F^*. Then we solve the feasibility problem at its mid-point $\alpha = (l + h)/2$. If it is feasible, then the optimal value F^* is in the lower half of the interval and we can shrink $[l, h]$ to $[l, \alpha]$. Otherwise, F^* must be in the upper half of the interval and we shrink $[l, h]$ to $[\alpha, h]$. The algorithm then continues on the identified half of the interval.

As we can see, at each iteration the range is shrunk by half, and the bisection algorithm is guaranteed to converge in $\lceil \log_2((h - l)/\varepsilon) \rceil$ iterations. For example, $[0, 100]$ allows the re-projection error to be as many as 100 pixels, which is guaranteed to contain the optimal value F^*. If we choose $\varepsilon = 0.5$ pixel, the algorithm will converge in only $\lceil \log_2 200 \rceil = 8$ iterations. Note that the number of iterations is independent of the dimension of the unknown $\mathbf{X}$, indicating that the algorithm is suitable for solving high dimensional problems. More importantly, the optimal value we derive is guaranteed to be less than $\varepsilon = 0.5$ pixel away from the true minimum value.

Algorithm: $\min_{\mathbf{X}} F(\mathbf{X})$.

1: Given $l \leq F^*$, $h \geq F^*$, and the tolerance $\varepsilon > 0$.
2: **while** $(h - l) > \varepsilon$ **do**
3: $\alpha = (h + l)/2$.
4: Solve the feasibility problem (22).
5: **if** (22) is feasible, **then** $h = \alpha$;
6: **else** $l = \alpha$.
7: **end while**

Figure 3: *Classical bisection algorithm to pin down the optimal value by searching in the one-dimensional range domain.*

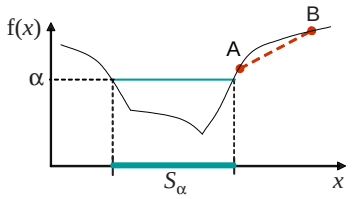


Figure 4: *A quasiconvex function. All of its α-sublevel sets $\{S_\alpha\}$ are convex. But this quasiconvex function is not convex, as can be seen from the line segment $\overline{AB}$ that lies below the function.*

3.2 Quasiconvex functions

The bisection algorithm in Fig. 3 is simple, deterministic, and it converges in a small number of iterations. It can even be applied to minimizing cost functions with multiple minima. The *critical step* in the algorithm is solving the feasibility problem in Eq. (22), which could be a hard problem by itself if the α-sublevel set of $F(\mathbf{X})$ is complicated. However, if S_α is convex, then Eq. (22) is a convex feasibility problem [2] that can be solved efficiently. A function with such convex α-sublevel set is called a *quasiconvex* function:

Definition 2. (see [2]) *A function $f : \mathbb{R}^n \to \mathbb{R}$ is* **quasiconvex** *if its domain $dom(f)$ and all its sublevel sets*

$$S_\alpha = \{x \in dom(f) \mid f(x) \leq \alpha\},$$

for $\alpha \in \mathbb{R}$, are convex.

A convex function has convex sublevel sets, and therefore, is quasiconvex. The reverse is not true in general. Fig. 4 shows an example of quasiconvex function that is not convex. The dash-line segment that lies below the function indicates the non-convexity of the function.

The reprojection error functions are not convex due to camera perspective effect, but they are quasiconvex:

Result 2. *A reprojection error function that conforms to the general form defined in Eq.* (4) *is a quasiconvex function.*

Proof. For any $\alpha > 0$, the α-sublevel set of $f(\mathbf{X})$ is:

$$\begin{aligned} S_\alpha &= \{\mathbf{X} \mid f(\mathbf{X}) \leq \alpha\} \\ &= \{\mathbf{X} \mid p(\mathbf{X}) - \alpha q(\mathbf{X}) \leq 0,\ q(\mathbf{X}) > 0\} \end{aligned}$$

From the definition of the general reprojection error function (Definition 1), we know that $p(\mathbf{X})$ is a convex function, and $-\alpha q(\mathbf{X})$ is a linear function and, therefore, a convex function. The sum of these two convex functions $\phi(\mathbf{X}) = p(\mathbf{X}) - \alpha q(\mathbf{X})$ is still a convex function. A sublevel set of a convex function is a convex set. As a result, S_α is a convex set since it is the intersection of two convex sets: the zero sublevel set of $\phi(\mathbf{X})$, and the half space defined by $q(\mathbf{X}) > 0$. Since $dom(f) = \mathbb{R}^n$ and S_α are all convex, we conclude that $f(\mathbf{X})$ is quasiconvex. □

3.3 Minimizing cost function F_∞

Result 3. $F_\infty(\mathbf{X})$*, the pointwise maximum of quasiconvex reprojection error functions* $f_i(\mathbf{X})$*, is also quasiconvex.*

Proof. The α-sublevel set S_α of $F_\infty(\mathbf{X})$ is:

$$\begin{aligned} S_\alpha &= \{\mathbf{X} \mid \max_i f_i(\mathbf{X}) \leq \alpha\} \\ &= \{\mathbf{X} \mid f_i(\mathbf{X}) \leq \alpha,\ \ i = 1, 2, \cdots, N\} \\ &= \bigcap_{i=1}^{N} S_\alpha^i \end{aligned}$$

Here S_α^i is the α-sublevel set of the reprojection error function $f_i(\mathbf{X})$. From Result 2, we know that $\{S_\alpha^i\}$ are all convex sets. As a result, their intersection S_α is also a convex set. Therefore, $F_\infty(\mathbf{X})$ is a quasiconvex function. □

Due to its quasiconvexity, F_∞ can be efficiently minimized by the bisection algorithm in Fig. 3. The convex set $S_\alpha = \bigcap_{i=1}^{N} S_\alpha^i$ can be expressed as:

$$S_\alpha = \{\mathbf{X} \mid q_i(\mathbf{X}) > 0; p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq 0;\ i = 1, \cdots, N\}$$

The feasibility problem of the bisection algorithm in Eq. (22) can now be solved by the following *convex* program:

$$\begin{aligned} \min_{\mathbf{X},\gamma} \ & \gamma \\ s.t. \ & -q_i(\mathbf{X}) + \epsilon \leq \gamma, \\ & p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma, \\ & i = 1, ..., N. \end{aligned} \tag{23}$$

Here ϵ is a small positive number. Denote γ^* the optimal value of (23). If $\gamma^* \leq 0$, then S_α of $F_\infty(\mathbf{X})$ is nonempty, and the problem in (22) is feasible; otherwise (22) is infeasible. Note that we do not need to solve (23) with high accuracy. The algorithm terminates whenever $\gamma \leq 0$ is satisfied, or whenever a dual feasible point is found with positive dual objective (which means $\gamma^* > 0$).

3.4 Minimizing robust cost function F_m

$F_\infty(\mathbf{X})$ is sensitive to outliers [6]. To deal with outliers, we use the robust cost function $F_m(\mathbf{X})$, which is defined as the m-th smallest reprojection error (see Eq. (20)). $F_m(\mathbf{X})$ is not a quasiconvex function, except for $m = N$, in which case F_m becomes F_∞.

However, since $F_m(\mathbf{X})$ is a pointwise function of a family of quasiconvex functions $\{f_i(\mathbf{X})\}$, its α-sublevel set can still be represented by the convex sublevel sets of these quasiconvex functions. As a result, we are able to extend the bisection algorithm to efficiently minimize F_m.

3.4.1 The α-sublevel set of F_m

A point $\mathbf{X}_0$ belongs to the α-sublevel set of $F_m(\mathbf{X})$ if and only if there exists a group of m α-sublevel sets whose intersection contains the point $\mathbf{X}_0$.

Result 4. *Denote S_α the α-sublevel set of $F_m(\mathbf{X})$. For any $\mathbf{X}_0$, $\mathbf{X}_0 \in S_\alpha$ if and only if $\mathbf{X}_0 \in_m \{S_\alpha^1, S_\alpha^2, \cdots, S_\alpha^N\}$. Here S_α^i is the α-sublevel set of $f_i(\mathbf{X})$. The symbol $\in_m$ means that there exist m sublevel sets in $\{S_\alpha^1, S_\alpha^2, \cdots, S_\alpha^N\}$ such that $\mathbf{X}_0$ is inside the intersection of these m sublevel sets.*

Proof. For any $\mathbf{X}_0$, we sort the N reprojection errors

$$f_1(\mathbf{X}_0), f_2(\mathbf{X}_0), \cdots, f_N(\mathbf{X}_0)$$

into the nondecreasing order

$$f_{(1)}(\mathbf{X}_0) \leq \cdots \leq f_{(m)}(\mathbf{X}_0) \leq \cdots \leq f_{(N)}(\mathbf{X}_0) \tag{24}$$

For the necessary condition, if $\mathbf{X}_0 \in S_\alpha$, then we have $F_m(\mathbf{X}_0) = f_{(m)}(\mathbf{X}_0) \leq \alpha$. The first m smallest reprojection errors $\{f_{(i)}(\mathbf{X}_0), i = 1, \cdots, m\}$ in Eq. (24) must therefore satisfy $f_{(i)}(\mathbf{X}_0) \leq \alpha$. As a result, $\mathbf{X}_0$ belongs to the intersection of the m α-sublevel sets of the first m functions in Eq. (24).

For the sufficient condition, suppose $\mathbf{X}_0$ is in the intersection of the following m sublevel sets: $\{S_\alpha^{(i)}, i = 1, \cdots, m\}$, where $S_\alpha^{(i)}$ is the α-sublevel set of $f^{(i)}$. We must have:

$$f^{(i)}(\mathbf{X}_0) \leq \alpha, \quad i = 1, \cdots, m \tag{25}$$

Now if $F_m(\mathbf{X}_0) = f_{(m)}(\mathbf{X}_0) > \alpha$, then from the sorted sequence in Eq. (24) we know that the number of less-than-α reprojection errors is less than m. This contradicts Eq. (25) where there are m less-than-α reprojection errors. Therefore we have $F_m(\mathbf{X}_0) \leq \alpha$, i.e., $\mathbf{X}_0 \in S_\alpha$. □

3.4.2 Feasibility by convex program

From Result 4, the feasibility problem in the bisection algorithm to minimizing F_m can be rewritten as:

$$\begin{aligned} &\text{find } \mathbf{X} \\ &s.t.\ \mathbf{X} \in_m \{S_\alpha^1, S_\alpha^2, \cdots, S_\alpha^N\} \end{aligned} \tag{26}$$

In other words, we need to determine if there exist m α-sublevel sets whose common intersection is non-empty. A straightforward approach is to check the feasibility of every possible group of m sublevel sets, where for each group its feasibility can be exactly determined by the convex program of Eq. (23). In worst case, this requires $\binom{N}{m}$ convex programs to solve Eq. (26), which is good for small N. When N is large, we can use either integer program or its convex approximation.

Integer program: minimizing number of infeasibility

Result 5. *The feasibility problem in Equation* (26) *can be formulated exactly by integer programming:*

$$\begin{aligned} &\min_{\mathbf{X},\boldsymbol{\gamma}} \ \gamma_1 + \gamma_2 + \cdots + \gamma_N \\ &s.t.\ \ -q_i(\mathbf{X}) < \gamma_i, \\ &\quad p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ &\quad \gamma_i = \{0, v\}, \\ &\quad i = 1, ..., N. \end{aligned} \tag{27}$$

Here $v > 0$ is a large positive integer.

Given optimal values $(\mathbf{X}^*, \boldsymbol{\gamma}^*)$ of the above minimization problem, if $\gamma_i = 0$, then the i-th reprojection error $f_i(\mathbf{X}) \leq \alpha$, i.e., $\mathbf{X}$ is inside the i-th α-sublevel set. On the other hand, if $\gamma_i = v$, then vX is outside the i-th α-sublevel set. Since the number of infeasibility $(\gamma_1 + \gamma_2 + \cdots + \gamma_N)$ is minimized, we have a solution where the number feasible constraints is maximized. In other words, we seek a solution which maximizes the number of α-sublevel sets who have non-empty common intersection. If this number is large than m, then Equation (26) feasible, otherwise, it is infeasible.

Convex program: minimizing sum of infeasibility

Although integer programming is well-studied and existing efficient package is available, it is in general more complicate than LP or SOCP.

Result 6. *We can use sum of infeasibility as an approximation to estimate the number of infeasible constraints:*

$$\begin{aligned} &\min_{\mathbf{X},\boldsymbol{\gamma}} \gamma_1 + \gamma_2 + \cdots + \gamma_N \\ &s.t.\ -q_i(\mathbf{X}) < \gamma_i, \\ &\quad p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ &\quad \gamma_i \geq 0, \\ &\quad i = 1, ..., N. \end{aligned} \tag{28}$$

Denote $\boldsymbol{\gamma}^* = (\gamma_1^*, \gamma_2^*, \cdots, \gamma_N^*)$ the optimal value of the above convex program achieving at $\mathbf{X}^*$. Denote g the number of zero elements in $\boldsymbol{\gamma}^*$. If $g \geq m$, then the problem defined by Eq. (26) *must be feasible; otherwise we consider Eq.* (26) *infeasible.*

γ_i^* is called the infeasibility of $f_i(\mathbf{X}^*)$. For any sublevel set S_α^i, if its corresponding infeasibility $\gamma_i^* = 0$, then $\mathbf{X}^*$ is inside S_α^i. As a result, the condition $g \geq m$ is sufficient for Eq. (26) to be feasible, since these g sublevel sets contain the common point $\mathbf{X}^*$.

While $g \geq m$ is a sufficient condition, it is an approximated necessary condition for Eq. (26) to be feasible. The exact conclusion about the infeasibility of Eq. (26) requires checking the feasibility of $\binom{N}{m}$ groups of m sublevel sets, or using integer programming to find the optimal point $\mathbf{X}^*$ that minimizes the number of infeasibilities (the number of nonzero components in $\boldsymbol{\gamma}^*$). Result 6 finds the minimum sum-of-infeasibilities $\sum_i \gamma_i^*$, and uses it to approximate the minimum number of infeasibilities. With such approximation, the bisection algorithm gives an upper bound on the true minimum value of F_m. The sum of infeasibility $\|\boldsymbol{\gamma}\|_1 = \sum_i \gamma_i$ is by itself a robust metric (L_1 norm is a robust metric), especially in our cases where the magnitude of outliers in the 2D measurements is bound by the image size. As a result, the bisection algorithm using Result 6 can usually achieve a tight upper bound on the true minimum value of F_m.

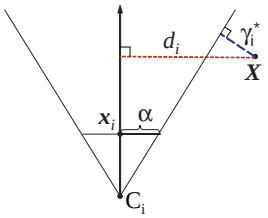


Figure 5: *When the camera matrix is normalized appropriately, the infeasibility γ_i^* is the distance from $\mathbf{X}^*$ to the α-convex cone. For comparison purpose, d_i is the distance to the ray back-projected from 2D measurement $\mathbf{x}_i$.*

Fig. 5 illustrates the meaning of γ_i^*. When the camera matrix is *normalized* appropriately, the infeasibility γ_i^* is the distance from $\mathbf{X}^*$ to the convex cone if $\mathbf{X}^*$ is outside the cone. If $\mathbf{X}^*$ is inside the cone S_α^i, then $\gamma_i^* = 0$. The sum-of-infeasibilities $\sum_i \gamma_i^*$ is therefore the sum of distances from $\mathbf{X}^*$ to the convex cones that do not contain $\mathbf{X}^*$. When $\mathbf{X}^*$ goes further away from the camera C_i, the camera C_i has larger uncertainty on $\mathbf{X}^*$. Such varying uncertainty is taken into account by γ_i^* as it is the distance to the cone, and the cone becomes larger as $\mathbf{X}^*$ goes further away from the camera C_i. This is in contrast to the distance to back-projected ray in 3D space (see Fig. 5).

Minimizing sum of weighted infeasibility

We can further improve the result by using weighted sum of infeasibility $\mathbf{w}^\top \boldsymbol{\gamma}$ in Eq. (28), where $\mathbf{w} = (w_1, \cdots, w_N)$ is the weight for each measurement, and $w_i \in [0, 1]$ can be set according to its corresponding reprojection error to down-weight outliers.

$$
\begin{aligned}
\min_{\mathbf{X},\boldsymbol{\gamma}} \; & w_1\gamma_1 + w_2\gamma_2 + \cdots + w_N\gamma_N \\
s.t. \; & -q_i(\mathbf{X}) < \gamma_i, \\
& p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\
& \gamma_i \geq 0, \\
& i = 1, ..., N.
\end{aligned}
$$

A simple weighting scheme is the $\{0, 1\}$ weighting where the weights w_i corresponding to active constraints are set to 0. Another scheme is to determine the weight w_i according to reprojection error $r_i = \left|\mathbf{x}_i - \frac{p_i}{q_i}\right|$.

3.5 Feasibility by LP or SOCP

When L_1- or L_2-norm error metric is used in defining the reprojection error function, the convex program for feasibility becomes small-scale linear programs (LP) or second-order

convex programs (SOCP), respectively.

3.5.1 L_1-norm error metric leads to LP

When L_1-norm error metric is used, the convex program in Eq. (28) becomes the following linear program:

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{\gamma}} \ & \gamma_1 + \gamma_1 + \cdots + \gamma_N \\ s.t. \ & -q_i(\mathbf{X}) + \epsilon \le \gamma_i, \\ & -\alpha q_i(\mathbf{X}) + p_{ui}(\mathbf{X}) - p_{vi}(\mathbf{X}) \le \gamma_i, \\ & -\alpha q_i(\mathbf{X}) + p_{ui}(\mathbf{X}) + p_{vi}(\mathbf{X}) \le \gamma_i, \\ & -\alpha q_i(\mathbf{X}) - p_{ui}(\mathbf{X}) - p_{vi}(\mathbf{X}) \le \gamma_i, \\ & -\alpha q_i(\mathbf{X}) - p_{ui}(\mathbf{X}) + p_{vi}(\mathbf{X}) \le \gamma_i, \\ & \gamma_i \ge 0, \quad i = 1, ..., N. \end{aligned} \tag{29}$$

Here p_{ui}, p_{vi}, and q_i are all linear functions of $\mathbf{X}$ (see Eq. (7) for the definition).

3.5.2 L_2-norm error metric leads to SOCP

When L_2-norm error metric is used, Eq. (28) becomes:

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{\gamma}} \ & \gamma_1 + \gamma_1 + \cdots + \gamma_N \\ s.t. \ & -q_i(\mathbf{X}) + \epsilon \le \gamma_i, \\ & \|\mathtt{A}_i\mathbf{X}\|_2 \le \alpha q_i(\mathbf{X}) + \gamma_i, \\ & \gamma_i \ge 0, \quad i = 1, ..., N. \end{aligned} \tag{30}$$

Here

$$\mathtt{A}_i = \begin{pmatrix} u_i\mathbf{c}_i^\top - \mathbf{a}_i^\top \\ v_i\mathbf{c}_i^\top - \mathbf{b}_i^\top \end{pmatrix}$$

is a 2×3 matrix, and $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are known vectors (see Eq. (5) for the notation). $\alpha q_i(\mathbf{X}) + \gamma_i$ is a linear function of $\mathbf{X}$. Therefore, the inequality

$$\|\mathtt{A}_i\mathbf{X}\|_2 \le \alpha q_i(\mathbf{X}) + \gamma_i$$

defines a second order convex cone [2]. As a result, Eq. (30) is a second-order cone programming (SOCP).

3.6 Explicit outlier detection

In Section 3.4, we have discussed how to use equation (28) to minimize the cost function F_m, the m-th smallest reprojection error, for a robust geometric reconstruction. In this section, we discuss how to explicitly detect outliers by examining the activeness of the constraints in the convex program of equation (28).

3.6.1 Inliers and outliers

Denote $\mathbf{X}_T$ the ground truth value of the unknown geometric parameters we want to estimate. For an inlier, its distance to the reprojection of $\mathbf{X}_T$ should follow zero-mean Gaussian distribution. On the other hand, for an outlier, such distance is larger than some value (e.g., 3σ, where σ is the standard deviation of the Gaussian distribution of the inliers' distance to $\mathbf{X}_T$).

Definition 3. *A measurement $\mathbf{x}_i$ is an inlier if and only if its α_T-sublevel set contains $\mathbf{X}_T$ as its interior point, i.e., $\mathbf{X}_T \in \text{int}(S_{\alpha_T})$. Here α_T is some predefined value.*

From the above definition, we have:

Result 7. *The α_T-sublevel sets of all inliers have a common intersection set that contains $\mathbf{X}_T$ as an interior point.*

3.6.2 Outlier detection

For convenience, equation (28) is presented here again:

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{\gamma}} \ & \gamma_1 + \gamma_2 + \cdots + \gamma_N \\ s.t. \ & -q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ & p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ & \gamma_i \geq 0, \\ & i = 1, ..., N. \end{aligned}$$

where ϵ is a small positive value.

The constraints in the convex program of Eq. (28) can be classified into active or inactive.

Definition 4. *Denote $h_i(\mathbf{X}) = p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) - \gamma_i$. Suppose $(\mathbf{X}^*, \boldsymbol{\gamma}^*)$ is the optimal solution of Eq.* (28)*. If $h_i(\mathbf{X}^*) = 0$, then the constraint $h_i(\mathbf{X}) \leq 0$ is* active *at $\mathbf{X}^*$. If $h_i(\mathbf{X}^*) < 0$, then the constraint $h_i(\mathbf{X}) \leq 0$ is* inactive *at $\mathbf{X}^*$.*

For an active constraint $h_i(\mathbf{X})$, the optimal solution $\mathbf{X}^*$ is either at the boundary of S^i_α (when $\gamma_i = 0$), or outside of S^i_α (when $\gamma_i > 0$).

If there are outliers in the measurements, then for outlier detection we only care about the case of $\alpha > \alpha_T$. Since if we can achieve $\alpha \leq \alpha_T$ in the bisection algorithm, we know the measurements are all inliers according to Definition 3.

Result 8. *If there are outliers in the measurement $\{\mathbf{x}_i\}$, then the convex program* (28) *achieves some positive optimal value $\mathbf{1}^\top\boldsymbol{\gamma}^*$ at $(\mathbf{X}^*, \boldsymbol{\gamma}^*)$, given $\alpha = \alpha_T - \epsilon$ (Here ϵ is a very small positive value). Moreover, at least one active constraint comes from some outlier.*

Proof. From the definition, it is obvious that (28) will achieve some positive optimal value, denoted by $\mathbf{X}^*$. From Result 7, we know the α-sublevel sets of all inliers have a common intersection that contains $\mathbf{X}_T$. *Suppose* all active constraints are from inliers. Denote S_I the common intersection of the α-sublevel sets of these (active) inliers. Then from Result 7, we know S_I is a non-empty set that contains $\mathbf{X}_T$ as an interior point. By choosing any point $\mathbf{X}_0 \in S_I$, we achieve a better objective value of *zero* for the convex program (28). This is in contradiction to the condition that the convex program (28) achieves a positive optimal value. Therefore, some active constraints must be from outliers. □

Result 8 suggest a simple mechanism to detect outliers, i.e., in each iteration of the bisection algorithm (see figure 3), discard the measurements corresponding to active constraints, until $\alpha < \alpha_T$. But one should be aware that some of the inliers may be discarded as they may be active. In some applications, the number of constraints is small (e.g., 3D triangulation given only 3 or 4 views), minimizing the F_m may be favorable to removing active constraints.

3.7 Geometric interpretation

The minimization algorithm we presented in this section has intuitive geometric interpretation. We use multi-view triangulation as an example to illustrate. For each 2D measurement,

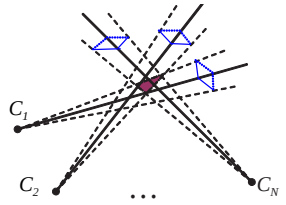
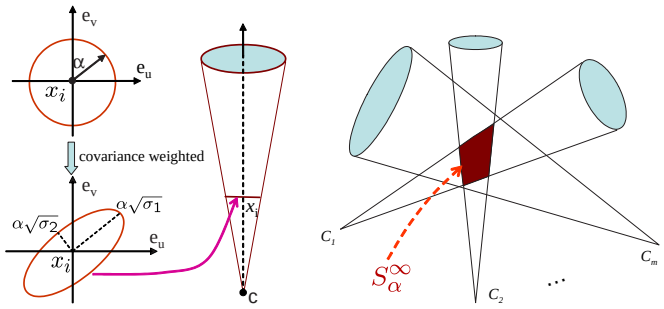


Figure 6: *Geometric illustration of 3D reconstruction using quasiconvex optimization to minimize F_m. Here L_1 norm is used in the reprojection error function. The algorithm seeks the minimum cone size with which at least m cones have non-empty intersection.*



(a) (b)

Figure 7: *Geometric illustration of 3D reconstruction using quasiconvex optimization. Here, L_2 norm is used in the reprojection error function. (a): The shape and size of each cone are determined by α and covariance matrix $\mathtt{Q}_i$; (b): The algorithm seeks minimum α such that the cones have non-empty* **common** *intersection S_α^∞. Note that cones have different shapes and sizes. The last cone C_m has large directional uncertainty and has little constraint on determining S_α^∞ in the direction that has large uncertainty.*

the camera optical center and the six linear inequalities in Eq.(29) form a convex cone S_α^i in front of the camera in the 3D space, as shown in Fig. 6. The cone size is determined by α. For any point inside the convex cone S_α^i, its reprojection error must be less than α. If the common intersection of at least m convex cones is not empty, then we conclude that there exists at least one point $\mathbf{X}_0$ in the 3D space such that the cost function $F_m(\mathbf{X}_0) \leq \alpha$. Minimizing $F_m(\mathbf{X})$ is therefore equivalent to adjusting α, the size of the convex cone, until we find the minimum α with which the intersection of at least m convex cones is non-empty.

Note that as a camera is further away from the 3D point $\mathbf{X}_0$, it has weaker constraint, since the convex cone size at $\mathbf{X}_0$ becomes larger. This is a nice property since the further away from the camera, the larger uncertainty about the 3D position the camera has.

3.7.1 Geometric interpretation: covariance-weighted quasiconvex minimization

The covariance-weighted quasiconvex minimization algorithm also has an intuitive geometric interpretation. We use triangulation as an example to illustrate.

For a feature point $\mathbf{x}_i$, the α-sublevel set S_i^w of the covariance-weighted reprojection error function $f_i^w(\mathbf{X})$ is a second order convex cone in the 3D space in front of the camera. The shape and size of such convex cone are determined by α and the covariance matrix $\mathtt{Q}_i = \mathtt{U}\,\mathrm{diag}(\sigma_1, \sigma_2)\mathtt{U}^\top$, as shown in Fig. 7(a). The construction of convex cone S_i^w is the following. First a circle in the image plane with radius α is scaled by $\sqrt{\sigma_1}$ and $\sqrt{\sigma_2}$ in u and v direction, respectively. This results in an ellipse with axes of $\alpha\sqrt{\sigma_1}$ and $\alpha\sqrt{\sigma_2}$ respectively. The ellipse is then rotated by the rotation matrix $\mathtt{U}$. The final convex cone, dubbed as *elliptical cone*, is

formed by connecting the camera optical center and the rotated ellipse.

The α-sublevel set S_α^∞ of the cost function $F_\infty^w(\mathbf{X})$ is the intersection of all α-sublevel sets $\{S_i^w\}$. Minimizing the covariance-weighted cost $F_\infty^w(\mathbf{X})$ is therefore equivalent to determining if there exists common intersection of the convex elliptical cones $\{S_i^w\}$, as shown in Fig. 7(b). The bisection algorithm then seeks a minimum value α such that the common intersection of the convex elliptical cones $\{S_i^w\}$ is non-empty.

Since α is common for all convex cones, the relative shape and size of each individual cone is actually determined by its associated covariance matrix. Therefore the effect of each convex cone constraint on the final estimation $\mathbf{X}$ is weighted by the inverse covariance matrix. Consider an extreme case when uncertainty goes to infinity. In such case, the corresponding convex cone is scaled to infinite size and does not have any constraint on the estimate of $\mathbf{X}$. On the other hand, if the uncertainty is zero, the convex cone becomes a ray, and we must constraint $\mathbf{X}$ on the ray, which is a strong constraint. When directional uncertainty is presented, the effect of constraints from different directions are determined by σ_1 and σ_2, respectively.

3.8 Differences from algebraic method

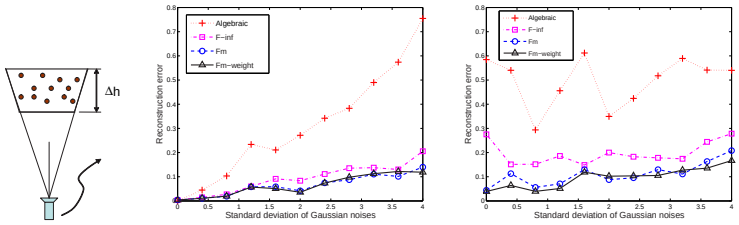
Both the linear program in Eq. (29) and the algebraic approach in Section 2.1.1 are linear. However, they have important differences.

The algebraic approach directly computes an *approximated* estimation of the *unknown parameters*, while the linear program only determines the *feasibility* in the bisection algorithm. It is the overall bisection algorithm that produces the final estimation of the unknown parameters.

The algebraic approach outputs an *approximated* estimation of the unknown parameters from an over-constrained linear equation set. It minimizes the sum of squared algebraic distance that is neither geometrically nor statistically meaningful. The noise in every linear equation affects the final estimation of the unknown parameters. In our approach, the noise has been taken into account by the cone size α. Once α is given, the resulting linear program is considered to be noiseless in a sense that it gives *exact* solution to determining the feasibility for bisection algorithm. The redundant constraints in the linear program will be inactive and, therefore, do not affect its final solution.

4 Experiments

We apply our quasiconvex optimization algorithm to multi-view triangulation and sequential structure from motion (SFM)(see [1]), and evaluate the performance using both synthetic and



(a) (b) (c)

Figure 8: *Multi-view triangulation: synthetic data. (a): The camera is rotating and translating, imaging a 3D scene consists of 40 points; (b): Reconstruction errors (normalized by Eq.* (31)*) with zero-mean Gaussian noises added to 2D point coordinates; (c): Reconstruction errors under both Gaussian noises and outliers.*

real data. We also show that using covariance-weighted error functions can greatly improve the reconstruction results.

4.1 Multi-view triangulation: synthetic data

The synthetic scene contains forty 3D points, distributed at different depth, that are imaged by a moving synthetic camera, as shown in Fig. 8(a). We use 10 consecutive views in the triangulation. Controlled zero-mean Gaussian noises and outliers are added to the 2D points. We apply our algorithm to minimize three cost functions F_∞, F_m, and F_m^w. Here F_m^w denotes F_m with weighted sum-of-infeasibilities used in Eq. (28). The reconstruction results from the algebraic approach (see Section 2.1.1) are included for comparison purpose.

Fig. 8 shows the average reconstruction errors, where (b) shows results when Gaussian noises are added to the 2D positions at increasing variances, and (c) shows the results with both Gaussian noises and 50% of outliers. The reconstruction error is normalized by

$$err = \frac{\|\mathbf{Z} - \mathbf{Z}_T\|_2}{\|\mathbf{Z}_T\|_2} \tag{31}$$

where $\mathbf{Z}_T$ is the known ground truth of 3D position, and $\mathbf{Z}$ is the triangulation result. As we can see, the algebraic approach has poor performance when there are noises or outliers, while our quasiconvex optimization successfully minimizes F_∞, F_m, and F_m^w. Without outliers, F_∞, F_m, and F_m^w have similar performance, with F_m and F_m^w better than F_∞ when the noises become larger. When there are outliers, the performance of F_∞ degrades quickly.

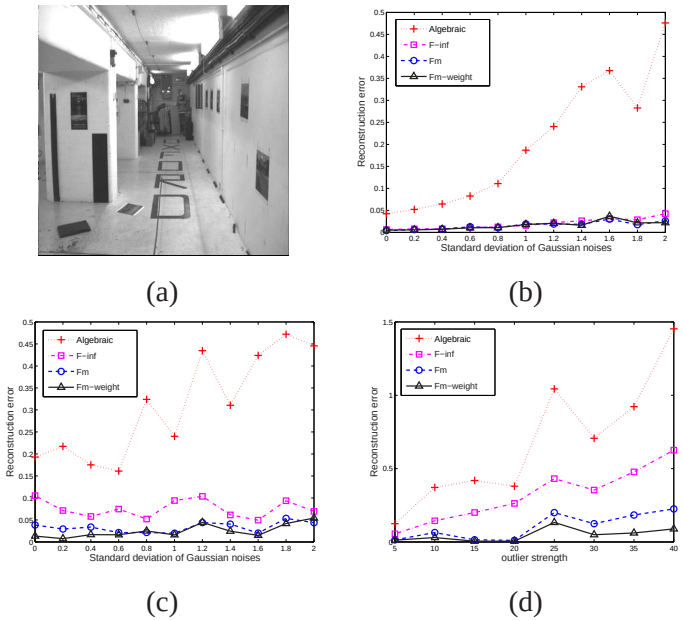


Figure 9: *Multi-view triangulation: corridor sequence. (a): The first image of this 11-frame sequence; (b): Reconstruction errors (normalized by Eq.* (31)*) with added zero-mean Gaussian noises; (c): Reconstruction errors with both Gaussian noises and outliers. For each feature track, the outliers are added to 1 to 3 views, depending on the number of views in which the corresponding 3D point is visible. (d): Reconstruction errors with increased strength of outliers (ranged from 5 to 40 pixels).*

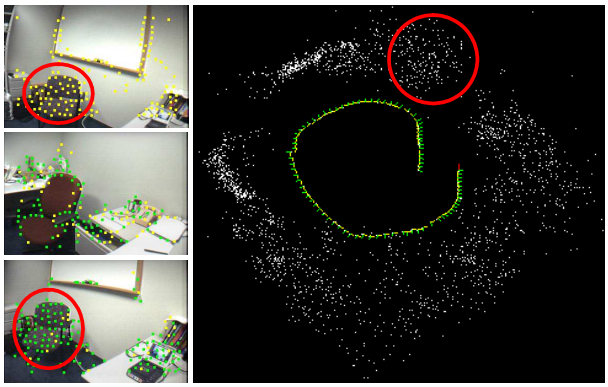
4.2 Multi-view triangulation: real data with "ground truth"

We use the *corridor* sequence [2] in which the camera is moving forward along the corridor. Fig. 9(a) shows the first frame of this 11-frame sequence. Along with the sequence, the 2D feature tracks, camera projection matrices, and 3D points are also provided. We use 2D feature tracks and camera matrices for triangulation, and compare the recovered 3D against the provided "ground truth".

Controlled zero-mean Gaussian and/or outliers are added to the 2D feature coordinates. Fig. 9(b) and (c) show the reconstruction errors. The results are consistent with those from the synthetic data experiment. Again, our quasiconvex optimization successfully minimizes F_∞, F_m, and F_m^w.

We observed that F_∞ is determined by outliers. Its performance depends on the "strength" of the outliers. Fig 9(d) shows the results where the strength of one outlier is increased. As we can see, the performance from F_∞ degrades quickly when outlier strength is increased. F_m^w performs better than F_m when outlier strength is large. When the 2D feature tracking error is less than 25 pixels, F_m performs as well as F_m^w, indicating that in real scenarios F_m is usually good enough.

[2] `http://www.robots.ox.ac.uk/~vgg/data1.html`



(a) (b)

Figure 10: *Multi-view triangulation in sequential SFM. The camera is moved (largely forward motion) around inside the office. (a): The first, middle, and last frame of the 450-frame sequence (image size* 360×240*), with tracked points superimposed. (b): Top-down view of the reconstruction results of camera trajectory and 3D points. The yellow lines show the optical axis of the recovered cameras. The red circle indicates the 3D points corresponding to the chair.*

4.3 Application: sequential structure from motion

Our target application is vision-aided small and micro aerial vehicle navigation, in which sequential SFM is applied to estimate both the camera motions and the 3D. We apply our multi-view triangulation using F_m minimization to the sequential SFM.

A 450-frame image sequence is taken by a mini camera that was moved around by hand in an office. Fig. 10(a) shows the first, middle, and last frames in this sequence. The camera is mostly moving forward, which is typical for a micro aerial vehicle. The forward motion makes the 3D estimation very challenging. Moreover, the images captured by the mini camera have low quality, resulting in noisy 2D feature tracking. We therefore seek to use as many frames as possible in triangulating a 3D point.

Fig. 10(b) shows the final reconstruction result (without global bundle adjustment). The red circle indicates the points from the chair visible both in the first and the last image. In the 3D view, the reconstruction of those points at the end of the sequence aligns very well with their reconstruction at the beginning of the sequence, indicating a good estimation of both the 3D and the camera motions.

4.4 Experiments: quasiconvex minimization with uncertainty

We use planar homography estimation and multi-view triangulation as two applications of our algorithm, and use synthetic and real data to evaluate its performance. We compare the performances of three algorithms: the quasiconvex minimization of F_∞^w with uncertainty

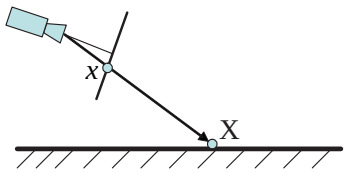


Figure 11: *The set up for synthetic data generation for homography estimation. The camera is looking at points on the ground plane. The image coordinates are normalized such that the camera focal length* $f = 1$.

model, the quasiconvex minimization of F_∞ without utilizing feature uncertainty, and the normalized linear algorithm.

4.4.1 Homography estimation with uncertainty: synthetic data

Fig. 11 shows the setup to generate the synthetic data for homography estimation, where the camera images the points on the "ground plane". This simulates the case where a camera mounted on a vehicle is looking at the ground plane at some angle. Note that in this case h_{33} in homography $\mathtt{H}_T$ may become very small. In all the algorithms being compared, we do not assume $h_{33} = 1$.

We randomly generate twenty 3D points $\mathbf{X}_i$ on the ground plane, and compute the 2D images $\mathbf{x}_i$ of these 3D points. We then add elliptical Gaussian noise to $\mathbf{x}_i$. The noise perturbed points are denoted by $\tilde{\mathbf{x}}_i$. The ellipticity of the noise is measured by $r = \sqrt{\sigma_{max}/\sigma_{min}}$, where σ_{max} and σ_{min} are the major and minor axes of the uncertainty ellipse, respectively. The orientation of the ellipse is randomly selected for each point.

We compare the performances using four criteria:

- Maximum reprojection error

$$F_\infty(\mathbf{X}) = \max_i d(\tilde{\mathbf{x}}_i, \mathtt{H}\mathbf{X}_i) \tag{32}$$

  where $d(\cdot, \cdot)$ denotes the Euclidean distance;

- Root of Mean Squares (RMS) of reprojection errors;

- Maximum covariance-weighted reprojection error $F_\infty^w(\mathbf{X})$ as defined in Eq. (19);

- Error in $\mathtt{H}$ defined as:

$$e_H = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d^2(\mathbf{x}_i, \mathtt{H}\mathbf{X}_i)} \tag{33}$$

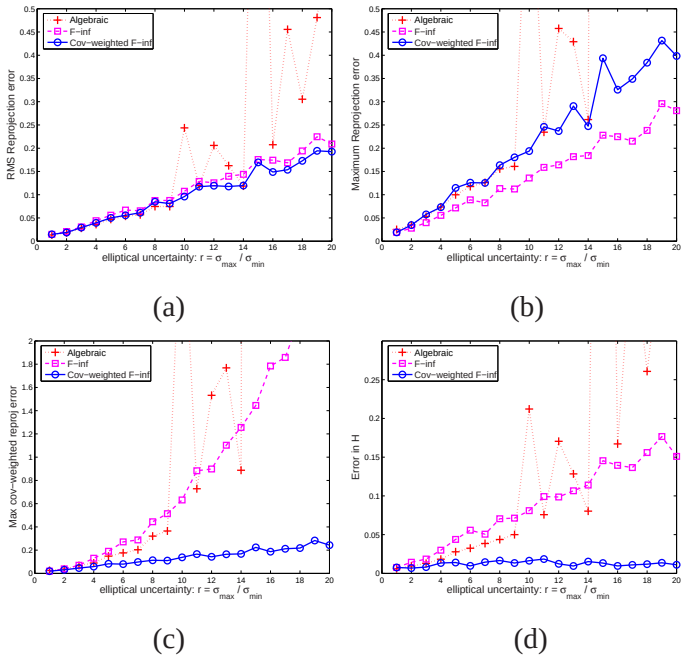


Figure 12: *Results from planar homography estimation. (a): RMS error; (b): Maximum of reprojection error; (c): Maximum of covariance-weighted reprojection error; (d): Error in* $\mathtt{H}$.

where $d(\mathbf{x}_i, \mathtt{H}\mathbf{X}_i)$ is the Euclidean distance between the ground truth 2D point $\mathbf{x}_i$ and the reprojection point $\mathtt{H}\mathbf{X}_i$. This error metric compares the estimated $\mathtt{H}$ to the ground truth $\mathtt{H}_T$ using ground truth points. If there is not error in the estimated $\mathtt{H}$, then $e_H = 0$.

We apply our algorithm to estimate $\mathtt{H}$ from this synthetic data. We repeat the experiments for 20 times, and report the above error measures using mean of these 20 runs. Fig. 12 shows the results, where $\sqrt{\sigma_{min}} = 0.01$ and $\sqrt{\sigma_{max}}$ varies from 0.01 to 0.2, i.e., the ellipticity r varies from 1 (isotropic) to 20.

As we can see from (a), both F_∞ and F_∞^w have similar RMS error. Normalized linear algorithm has similar RMS error when r is small, but becomes unreliable when $r \geq 10$.

Fig. 12(b) shows that minimizing F_∞ gives lowest maximum reprojection error, while Fig. 12(c) shows that minimizing F_∞^w gives lowest maximum covariance-weighted reprojection error. This indicates that quasiconvex minimization indeed achieves the global minimum of F_∞ and F_∞^w, respectively.

From Fig. 12(d), which compares the estimated $\mathtt{H}$ against ground truth $\mathtt{H}_T$ using the metric e_H, we can see that F_∞^w performs the best, and its performance does not degrade at all with the increase of r. This indicates that F_∞^w is the proper metric to minimize. We also find that normalized linear algorithm performs better than F_∞ when r is small, but when r is large, its performance becomes unreliable.

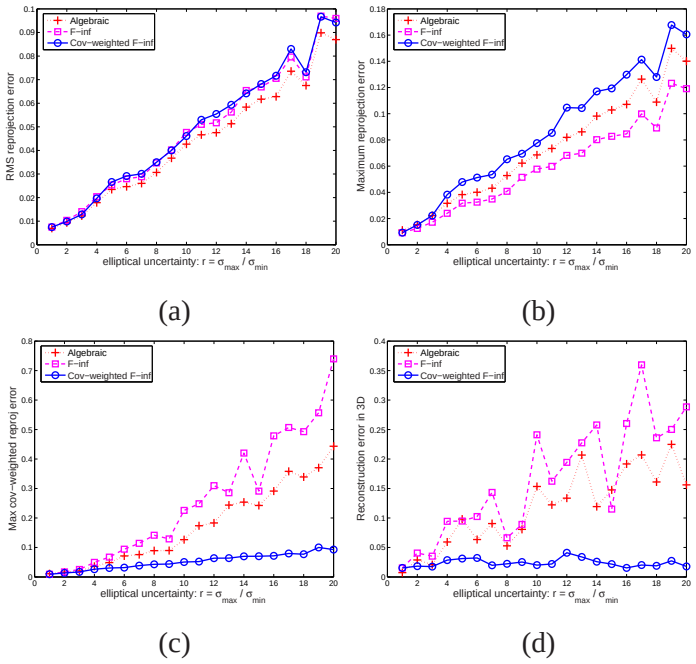


Figure 13: *Results from multi-view triangulation. (a): RMS error; (b): Maximum of reprojection error; (c): Maximum of covariance-weighted reprojection error; (d): Reconstruction error compared against ground truth 3D.*

4.4.2 Mulit-view triangulation with uncertainty: synthetic data

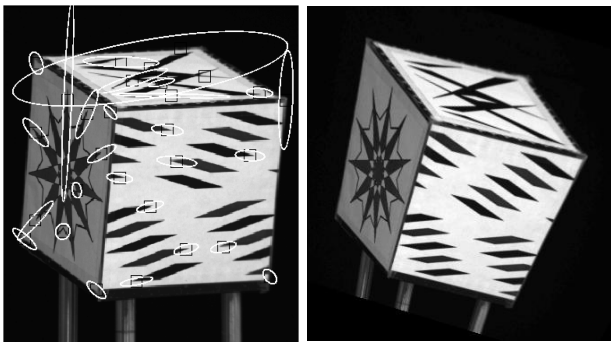
We use the set up in Fig. 8(a) to generate the synthetic data for multi-view triangulation with uncertainty. The camera is rotating and translating, and takes ten consecutive views of twenty 3D points located at different depth. Directional Gaussian noises are added to the locations of image points.

Fig. 13 shows the average results of 20 runs. Again it shows that the covariance-weighted reprojection error is the right metric to use, as can be seen by the fact that F_{∞}^{w} gives the best 3D estimation $\mathbf{Z}$ when compared to the ground truth $\mathbf{Z}_T$ using the following metric:

$$e_{3D} = \frac{\|\mathbf{Z} - \mathbf{Z}_T\|_2}{\|\mathbf{Z}_T\|_2} \tag{34}$$

4.5 Infinite elliptical uncertainty

The performance of F_{∞}^{w} does not degrade even when the ellipticity of noises r goes to essentially infinity, as can be seen from Table 1. This fact indicates that the normal optical flow can be modelled by directional (infinity) uncertainty. As a result, the point feature and line feature can be simultaneously used in the quasiconvex optimization for many geometric reconstruction problems.



(a) (b)

Figure 14: *Two input images. (a): First image, where elliptical uncertainties for some feature points are shown; (b): Last image.*

4.5.1 Homography estimation with uncertainty: real data

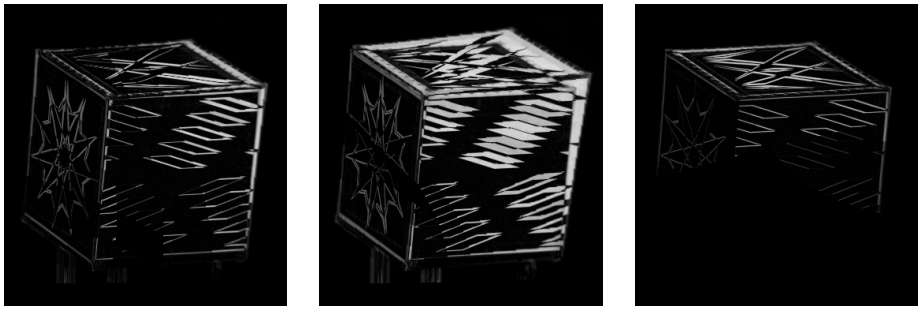
We apply our algorithm to estimate the inter-image homography using real image data. Fig. 14 shows two input images. Here Fig. 14(a) shows the first image. The other images are obtained by applying a known and gradually-changed planar homography to image (a). This way we have the ground truth of the planar homography for evaluation purpose.

The features in the first image are tracked through the sequence, and the inverse covariance matrix for each feature is computed by the Hessian matrix (Eq. (9)). The elliptical uncertainties for some feature points are shown in Fig. 14(a).

Fig. 15 shows the image residual by applying the inverse-warping using the estimated homographies. As we can see, minimizing F_∞^w gives the homography that has the lowest intensity residual. It correctly down weight the features with large directional uncertainty on the top of the box to produce a correct estimate of `H`. On the other hand, both normalized linear algorithm and the minimization of F_∞ give worse results, as can be seen by the large residuals on the top of the box, where there exist features with large directional uncertainty. Normalized linear algorithm performs better than minimizing F_∞ in this case.

	Homography	Triangulation
Algebraic	833.8092	1.4380
F_∞	710.3891	3.1084
F_∞^w	0.0093	0.0146

Table 1: *Results under infinite elliptical uncertainty $r = 10^5$. The table shows e_H for homography estimation and e_{3D} for triangulation.*



(a) (b) (c)

Figure 15: *Residual image by applying the estimated homography. Note that residuals are scaled up for visibility. The average pixel residual is shown as e for each case too. (a): Normalized linear algorithm, $e = 15.0798$; (b): Minimizing F_∞, $e = 28.2082$; (c): Minimizing F_∞^w, $e = 5.8896$.*

5 Conclusion

We have presented a novel quasiconvex optimization framework to geometric reconstruction problems. Our algorithm is an efficient bisection search in the *one*-dimensional range domain, with each search step accomplished by a small-scale convex program that can be efficiently solved. We derived the algorithm based on sound mathematical grounds, and the algorithm is essentially free of parameter tuning. The final algorithm is simple, robust, deterministic, and has very intuitive geometric interpretation. We have demonstrated the effectiveness of our approach, using both synthetic and real data.

We identified the general quasiconvex formulation of the reprojection error functions, therefore our quasiconvex optimization framework can be potentially applied to many other estimation problems. We are investigating the applications of our approach to space carving [8], multi-baseline stereo reconstruction, and efficient bundle adjustment [16] in structure from motion.

Appendix: Chierality constraint for homography estimation

The camera matrix appears in the first three problems in Section 1.1 and the Chierality constraint is straightforward to be represented, for example, as $\mathbf{p}_3^\top \mathbf{X}$, where $\mathbf{p}_3^\top$ is the third row of the camera matrix `P` and $\mathbf{X}$ is the point in 3D space. But the camera matrix does not appear in the problem of planar homography estimation. In such case we need to represent the Chierality constraint in terms of the planar homography `H` in stead of the camera matrix `P`.

Result 9. *For a 2D point $\mathbf{u} = [u, v, 1]^\top$ in the first image, the chierality constraint for planar homography estimation can be written as $\mathbf{h}_3^\top \mathbf{u} > 0$, where $\mathbf{h}_3^\top$ is the third row of* `H`.

Proof. Assume that a plane $\pi^\top \mathbf{X} = 0$, with $\pi = [\mathbf{v}^\top, 1]^\top$, is imaged by two cameras $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{A} \mid \mathbf{b}]$, where the i-th row of $\mathtt{A}$ is denoted as $\mathbf{a}_i^\top$, and $\mathbf{b} = [b_1, b_2, b_3]^\top$. The planar homography can then be written as [7]:

$$\mathtt{H} = \mathtt{A} - \mathbf{b}\mathbf{v}^\top \tag{35}$$

Suppose $\mathbf{u} = [u, v, 1]^\top$ is the image (in the first camera) of the 3D point

$$\mathbf{X} = [\mathbf{x}^\top, 1]^\top = [X, Y, Z, 1]^\top.$$

Then we have

$$Z\mathbf{u} = \mathtt{P}\mathbf{X} = \mathbf{x}.$$

Since $\mathbf{X}$ is on the plane π. We have

$$\mathbf{X} = [\mathbf{x}^\top, -\mathbf{v}^\top \mathbf{x}]^\top.$$

The chierality constraint is:

$$\begin{aligned} \mathbf{p}_3'^\top \mathbf{X} &= [\mathbf{a}_3^\top, b_3][\mathbf{x}^\top, -\mathbf{v}^\top \mathbf{x}]^\top \\ &= (\mathbf{a}_3^\top - b_3 \mathbf{v}^\top)\mathbf{x} \\ &= \mathbf{h}_3^\top \mathbf{x} \\ &= Z\ \mathbf{h}_3^\top \mathbf{u} \\ &> 0 \end{aligned}$$

where Z is the depth of the 3D point $\mathbf{X}$. From the chierality constraint for the first camera $\mathtt{P}$, we have $Z > 0$. As a result, $\mathbf{h}_3^\top \mathbf{u} > 0$. □

References

[1] P. A. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *IJCV*, 23:235–259, June 1997.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] D.D. Diel, P. DeBitetto, and S. Teller. Epipolar constraints for vision-aided inertial navigation. In *EEE Workshop on Motion and Video Computing*, 2005.

[4] R.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the 3-point perspective pose estimation problem. *IJCV*, 13(3):331–356, 1994.

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

[6] R. I. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *CVPR 2004*.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000.

[9] J. Oliensis. Exact two-image structure from motion. *PAMI*, 24(12):1618–1633, 2002.

[10] Long Quan and Zhongdan Lan. Linear n $\geq$ 4-point pose determination. In *ICCV 1998*.

[11] R.Hartley and P.Sturn. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[12] Jianbo Shi and Carlo Tomasi. Good features to track. In *CVPR'94*.

[13] R. Matt Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. In *CVPR*, 2005.

[14] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[15] Bill Triggs, P. McLauchlan, Richard Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. *Springer Lecture Notes in Computer Science*, 1883:298–372, 2000.

[16] M. Uyttendaele, A. Criminisi, S.B. Kang, S. Winder, R. Hartley, and Richard Szeliski. High-quality image-based interactive exploration of real-world environments. In *IEEE Computer Graphics and Applications*, 2004.

[17] Z.Y. Zhang. Determining the epipolar geometry and its uncertainty: A review. *IJCV*, 27(2):161–195, 1998.