Quasiconvex Optimization for Robust Geometric Reconstruction

Qifa Ke, *Member, IEEE*, and Takeo Kanade, *Fellow, IEEE*

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 readily solvable. Our final reconstruction algorithm is simple and has intuitive geometric interpretation. In contrast to existing local minimization approaches, our algorithm is deterministic and guarantees a predefined accuracy of the minimization result. The quasiconvexity also provides an intuitive method to handle directional uncertainties and outliers in measurements. For a large-scale problem or in a situation where computational resources are constrained, we provide an efficient approximation that gives a good upper bound (but not global minimum) on the reconstruction error. We demonstrate the effectiveness of our algorithm by experiments on both synthetic and real data.

Index Terms—Multiview geometry, geometric reconstruction, convex programming, directional uncertainty, robust.

✦

---

1 INTRODUCTION

G IVEN measurements in 2D images, the goal of geometric reconstruction in computer vision is to estimate the three-dimensional information about the scene and/or the camera motions. Classical examples include triangulation [1], camera resectioning [2], [3], and structure from motion (see [4] 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 nonconvex and often contains multiple local minima. Minimizing F_s is therefore difficult. Hartley and Schaffalitzky [5] proposed using L_∞, the *pointwise maximum* of the squared reprojection errors, as the cost function, which we denote as F_∞. It was first shown by Hartley and Schaffalitzky in their pioneer work [5] that F_∞, in contast to F_s, contains only one single minimum value in its feasible domain. An approach using random line search in the parameter space was used in [5] to minimize F_∞. The convergence behavior of random line search remains unclear. As pointed out in [5], it is difficult to perform random line search when the parameter space is high dimensional. Constrained minimization is also proposed in [5] for minimizing F_∞. However, the constraints are nonlinear

and nonconvex, making such constrained minimization a difficult problem by itself. These properties of L_∞ has motivated two simultaneously works by Ke and Kanade [6] and Kahl [7] that offer efficient global minimization of F_∞.

We can consider the model-fitting error as a function of the unknown parameters, which is termed *reprojection error function* in this paper. The error function is quasiconvex for the geometric reconstruction problems under our consideration [6], [7]. F_∞, the pointwise maximum of a family of quasiconvex functions, is also a quasiconvex function. As a result, F_∞ can be efficiently minimized using an one-dimensional bisection algorithm [8]. 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. Previously, global optimal estimates can be achieved only in rare instances, such as two-view triangulation using L_2-norm [1] or L_∞-norm [9], three-view triangulation by solving polynomial equation set [10], and affine reconstruction using matrix factorization [11]. Minimizing F_∞ by quasiconvex optimization provides a framework to achieve global optimal estimation in many multiview geometric reconstruction problems, without the limitation on the number of views and/or the use of affine camera model.

It has been pointed out in [5] that F_∞ is sensitive to outliers. To handle outliers, we use F_m (see definition in (24)), the pointwise mth smallest reprojection error, as the cost function. In contrast to F_∞ or F_s, the cost function F_m is highly robust to outliers [12]. In spite of its complex formulation, in our cases F_m is still a pointwise operator of

---

- *Q. Ke is with Ricoh Innovations, Inc., California Research Center, 2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025-7022. E-mail: qifa@rii.ricoh.com.*
- *T. Kanade is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: tk@cs.cmu.edu.*

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). For small-scale problems, our approach can identify the global minimum of F_m. For large-scale problems, our approach can provide a good upper bound on the reconstruction error. But to locate the exact global minimum of F_m, we need to solve $\binom{N}{m}$ convex programs, where N is the number of inputs. Although this might be computational prohibitive for some applications for large-scale problems, it does provide a mechanism to locate the global minimum. However, we need to emphasize that F_m is no longer a quasiconvex function and is subject to potentially numerous local minima for large-scale problems.

Our quasiconvex minimization framework can also take directional uncertainty into account in an intuitive way. Cost functions (e.g., F_s and F_∞) are meaningful objectives to minimize only when the measurement noises are isotropic and *i.i.d.* (independent and identically distributed) at every 2D feature. In real data, this is rarely the case since the quality of feature matching depends on the image intensity pattern around the feature, which often varies at different feature points and has strong directionality to it. To account for the feature-dependent directional uncertainty, one should minimize the covariance-weighted reprojection error (the Mahalanobis distance), instead of the euclidean distance. We incorporate the directional uncertainty model into the quasiconvex optimization framework. The directional uncertainty can be characterized by the covariance matrix at each 2D feature, as has been used in matrix factorization for affine reconstruction [13], [14]. We show that the point wise maximum of covariance-weighted reprojection errors is still a quasiconvex function and, therefore, its global minimum can be obtained by the quasiconvex minimization framework. Moreover, since a line feature can be modeled as a feature point with infinite uncertainty along the line direction, point and line features can be used simultaneously for geometric reconstruction in a common quasiconvex optimization framework.

1.1 Background: Geometric Reconstruction Problems

We present some classical examples of geometric reconstruction problems in computer vision. These geometric reconstruction problems can all be solved using our method presented in this paper. We will also present, for each problem, the reprojection (in the image plane) of the geometric reconstruction result, which will be used later. The following problems have been studied by Ke and Kanade [6] and Kahl [7]. Kahl [7] showed that the reconstruction problem given a reference plane can also be solved by SOCP. Ke and Kanade [6] also showed that, in addition to SOCP, the simpler Linear Programming (LP) can be used to obtain the global minimum if robust L_1-distance is used to measure the reprojection error in the image plane. We have also investigated using convex optimization to handle directional uncertainties [15] and outliers [6].

1.1.1 Multiview Triangulation

Given projection matrices of N cameras, denoted by $\{P_i, i = 1, \ldots, N\}$ and the images of the unknown 3D point $\tilde{\mathbf{Z}} = (X, Y, Z, 1)^\top$ in these N cameras, denoted by $\{\mathbf{x}_i, i = 1, \ldots, N\}$, the task of multiview triangulation is to estimate $\tilde{\mathbf{Z}}$ from $\{P_i\}$ and $\{\mathbf{x}_i\}$. Triangulation is a necessary step in two or multiview 3D reconstruction and in structure from motion. Note that optimal triangulation algorithms [1], [16] for two-view case are not generalizable to multiview case.

The reprojection of $\tilde{\mathbf{Z}}$ on the ith image is

$$\hat{\mathbf{x}}_i^\top = \left(\frac{\mathbf{p}_i^{1\top}\tilde{\mathbf{Z}}}{\mathbf{p}_i^{3\top}\tilde{\mathbf{Z}}}, \; \frac{\mathbf{p}_i^{2\top}\tilde{\mathbf{Z}}}{\mathbf{p}_i^{3\top}\tilde{\mathbf{Z}}} \right), \tag{1}$$

where $\mathbf{p}_i^{k\top}$ is the kth row of the matrix P_i.

1.1.2 Camera Resectioning

Given 3D points $\{\tilde{\mathbf{Z}}_j, j = 1, \ldots, M\}$ and their images $\{\mathbf{x}_j, j = 1, \ldots, M\}$ in one camera, the task of camera resectioning is to estimate the camera projection matrix P from these N corresponding pairs $\{\mathbf{x}_j \leftrightarrow \tilde{\mathbf{Z}}_j\}$. Camera resectioning is used in camera calibration and in structure from motion.

The reprojection of $\tilde{\mathbf{Z}}_j$ on the image is

$$\hat{\mathbf{x}}_j^\top = \left(\frac{\tilde{\mathbf{Z}}_j^\top \mathbf{p}^1}{\tilde{\mathbf{Z}}_j^\top \mathbf{p}^3}, \; \frac{\tilde{\mathbf{Z}}_j^\top \mathbf{p}^2}{\tilde{\mathbf{Z}}_j^\top \mathbf{p}^3} \right), \tag{2}$$

where $\mathbf{p}^{k\top}$ is the kth row of matrix P.

1.1.3 Multiview 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 [5]. 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 [17]. Another example is that there are reconstruction methods in which the camera rotation for each frame is estimated in the first step [18]. Denote the N intrinsically calibrated cameras as $\{P_i = (R_i, -R_i\mathbf{C}_i), \; i = 1, \ldots, N\}$, where for each camera the rotation 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 jth 3D point $\tilde{\mathbf{Z}}_j = (X_j, Y_j, Z_j, 1)^\top$ onto the ith camera. The task is to estimate $\{\tilde{\mathbf{Z}}_j\}$ and $\{\mathbf{C}_i\}$ from the 2D points $\{\mathbf{x}_{ij}\}$ and the camera poses $\{R_i\}$.

The reprojection of $\tilde{\mathbf{Z}}_j$ onto the ith image is

$$\hat{\mathbf{x}}_{ij}^\top = \left(\frac{\mathbf{r}_i^{1\top}\mathbf{X}_{ij}}{\mathbf{r}_i^{3\top}\mathbf{X}_{ij}}, \; \frac{\mathbf{r}_i^{2\top}\mathbf{X}_{ij}}{\mathbf{r}_i^{3\top}\mathbf{X}_{ij}} \right), \tag{3}$$

where $\mathbf{r}_i^{k\top}$ is the kth row of the rotation matrix R_i and $\mathbf{X}_{ij} = (X_j, Y_j, Z_j)^\top - \mathbf{C}_i$. The scale and the origin are ambiguous in the reconstruction of the 3D points $\{\tilde{\mathbf{Z}}\}$ and the camera locations $\{\mathbf{C}\}$. We set the first point $\tilde{\mathbf{Z}}_1$ as the origin (i.e., $\tilde{\mathbf{Z}}_1 = (0, 0, 0, 1)^\top$) and fix the scale by setting $\mathbf{C}_1 = (*, *, 1)^\top$.

1.1.4 Planar Homography Estimation [6], [7]

Two images of points on a 3D scene plane are related by a planar homography H, a 3×3 nonsingular matrix. Given

M correspondences $\{\mathbf{x}_j \leftrightarrow \mathbf{x}'_j, j = 1, \ldots, M\}$, the task is to estimate $\mathtt{H}$ such that $w_j \tilde{\mathbf{x}}'_j = \mathtt{H}\tilde{\mathbf{x}}_j$. Here, $\tilde{\mathbf{x}}^\top = (\mathbf{x}^\top, 1)$ and w_j is an unknown scale.

The mapping of $\mathbf{x}_i$ onto the second image is:

$$\hat{\mathbf{x}}_j^\top = \left(\frac{\tilde{\mathbf{x}}_j^\top \mathbf{h}^1}{\tilde{\mathbf{x}}_j^\top \mathbf{h}^3}, \; \frac{\tilde{\mathbf{x}}_j^\top \mathbf{h}^2}{\tilde{\mathbf{x}}_j^\top \mathbf{h}^3} \right), \tag{4}$$

where $\mathbf{h}^{k\top}$ is the kth row of $\mathtt{H}$.

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 Measurement

Using triangulation as an example, let us illustrate three error metrics for an individual 2D measurement. We denote the homogeneous coordinate $\tilde{\mathbf{Z}} = (\mathbf{Z}; 1) = (X, Y, Z, 1)^\top$.

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:

$$k_i \tilde{\mathbf{x}}_i = \mathtt{P}_i \tilde{\mathbf{Z}}. \tag{5}$$

The algebraic error distance for $\mathbf{x}_i$ is defined by

$$f_i(\tilde{\mathbf{Z}}) = \| k_i \tilde{\mathbf{x}}_i - \mathtt{P}_i \tilde{\mathbf{Z}} \|_2. \tag{6}$$

Linear least-squares can be applied to estimate $\tilde{\mathbf{Z}}$ by minimizing the sum of squared algebraic distances. Since the algebraic distance is not geometrically or statistically meaningful, the algebraic reconstruction could be unreliable (see [5], [12]), especially for reconstruction problems when the camera is moving mostly forward or when there are large noises or outliers.

2.1.2 Distance in 3D Space

In the case of calibrated cameras, $f_i(\mathbf{Z})$ can be defined as the *euclidean* distance from the 3D point $\mathbf{Z}$ to the ray back projected from $\mathbf{x}_i$ (the ray connecting the camera optical center and $\mathbf{x}_i$). In the case of two views, this distance function leads to 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 $\hat{\mathbf{x}}_i = \pi_i(\tilde{\mathbf{Z}})$

$$f_i(\tilde{\mathbf{Z}}) = \| \mathbf{x}_i - \hat{\mathbf{x}}_i \|_l = \| \mathbf{x}_i - \pi_i(\tilde{\mathbf{Z}}) \|_l, \tag{7}$$

where $\hat{\mathbf{x}}_i = \pi_i(\tilde{\mathbf{Z}})$ is the reprojection of $\tilde{\mathbf{Z}}$ 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.
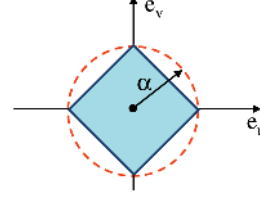

Fig. 1. Distance between $\mathbf{x} = (u, v)$ and $\hat{\mathbf{x}} = (\hat{u}, \hat{v})$. The solid square shows the contour in which the L_1 norm error $e_1 = |e_u| + |e_v| = \alpha$, while the dash line shows the contour in which the L_2 norm error $e_2 = \sqrt{e_u^2 + e_v^2} = \alpha$. Here, $e_u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

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 (7), 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

For a pin-hole camera, the reprojection error function defined in the image domain shares a common formulation in many geometric reconstruction problems. Identifying such common formulation enables us to apply our algorithm to different problems.

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

$$f(\mathbf{X}) = \frac{p(\mathbf{X})}{q(\mathbf{X})}, \tag{8}$$

where

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

Note that, for our applications, $q(\mathbf{X})$ is usually a linear function that constrains the admissible $\mathbf{X}$s. A linear function is also a concave function.

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 (8), 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

Theorem 1. *In affine or euclidean reconstruction,*[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 $\mathbf{x} = (u, v)^\top$ in the image, as shown in (1), (3), and (4), can be rearranged into the following general formulation:

$$\hat{\mathbf{x}} = \left(\frac{\mathbf{a}^\top \mathbf{X}}{\mathbf{c}^\top \mathbf{X}}, \; \frac{\mathbf{b}^\top \mathbf{X}}{\mathbf{c}^\top \mathbf{X}} \right)^\top. \tag{9}$$

---

1. In a way similar to the method in [5], our algorithm in this paper can be extended to projective reconstruction.

Here, $\mathbf{X}$ is the vector to be estimated. $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are known vectors.

The reprojection error function is

$$f(\mathbf{X}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_l = \|\frac{1}{q(\mathbf{X})}(p_u(\mathbf{X}), p_v(\mathbf{X}))^\top\|_l, \qquad (10)$$

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

$$\begin{aligned} p_u(\mathbf{X}) &= (u\mathbf{c}^\top - \mathbf{a}^\top)\mathbf{X}, \\ p_v(\mathbf{X}) &= (v\mathbf{c}^\top - \mathbf{b}^\top)\mathbf{X}, \\ q(\mathbf{X}) &= \mathbf{c}^\top \mathbf{X}. \end{aligned} \qquad (11)$$

It is obvious that $q(\mathbf{X})$ is a linear function of $\mathbf{X}$ and, therefore, it is a concave function.

In affine or euclidean reconstruction, the cheirality constraint [4], [19], which states that the 3D points visible in the image must be in front of the camera, can be expressed as $\mathbf{c}^\top \mathbf{X} > 0$. Therefore, we have $q(\mathbf{X}) > 0$. The reprojection error function in (10) can then be rewritten as

$$f(\mathbf{X}) = \frac{1}{q(\mathbf{X})}\|(p_u(\mathbf{X}), p_v(\mathbf{X}))\|_l = \frac{p(\mathbf{X})}{q(\mathbf{X})}. \qquad (12)$$

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 $p(\mathbf{X}) \geq 0$. $\qquad\square$

2.2.2 Uncertainty-Weighted Reprojection Error Function

When the measurement uncertainty on the location of each 2D feature point is available, the reprojection error at each feature should be weighted by its corresponding uncertainty. In this section, we show that such uncertainty-weighted reprojection error function still conforms to the general formulation in Definition 1. As a result, our algorithm in this paper can handle (directional) uncertainties.

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. [20], [21]):

$$\mathbf{Q}^{-1} = \frac{1}{s} \sum_{(u,v) \in w} \begin{pmatrix} I_u I_u, & I_u I_v \\ I_u I_v, & I_v I_v \end{pmatrix}, \qquad (13)$$

where w is a small window centered around the feature point in the image I of the ith 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 [21], which takes into account not only the image pattern but also the image pixel noises.

Fig. 2 shows the three different types feature uncertainty:

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

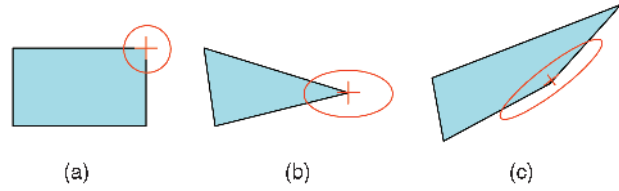
Fig. 2. Uncertainty in feature point locations. (a) Scalar uncertainty with covariance matrix $\mathbf{Q} = diag(\sigma, \sigma)$. (b) Directional but uncorrelated noises with covariance matrix $\mathbf{Q} = diag(\sigma_1, \sigma_2)$. (c) Directional and correlated noises, with covariance matrix $\mathbf{Q}$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

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

$$\mathbf{Q}^{-1} = \mathbf{U}\Sigma^{-1}\mathbf{U}^\top. \qquad (14)$$

Denote $\mathbf{B} = \Sigma^{-1/2}\mathbf{U}^\top$, which is a linear 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

$$\mathbf{x}' = (u', v')^\top = \mathbf{B}(u, v)^\top, \qquad (15)$$

$$\hat{\mathbf{x}}' = \mathbf{B}\hat{\mathbf{x}} = \frac{1}{\mathbf{c}^\top \mathbf{X}}\mathbf{B}\begin{pmatrix}\mathbf{a}^\top \\ \mathbf{b}^\top\end{pmatrix}\mathbf{X}, \qquad (16)$$

where $\mathbf{a}^\top$ and $\mathbf{b}^\top$ follow the notation in (9). The covariance matrix of the noise in the covariance-weighted data space now becomes isotropic and takes the form of $diag(1, 1)$. Denote

$$\mathbf{A} = \left[\mathbf{B}\begin{pmatrix}u \\ v\end{pmatrix}\mathbf{c}^\top - \mathbf{B}\begin{pmatrix}\mathbf{a}^\top \\ \mathbf{b}^\top\end{pmatrix}\right]. \qquad (17)$$

The covariance-weighted reprojection error function is

$$f^w(\mathbf{X}) = \|\mathbf{x}' - \hat{\mathbf{x}}'\| = \left\|\frac{\mathbf{A}\mathbf{X}}{\mathbf{c}^\top \mathbf{X}}\right\|. \qquad (18)$$

Again, the cheirality constraint [4], [19], 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, (18) can then be written as

$$f^w(\mathbf{X}) = \frac{1}{\mathbf{c}^\top \mathbf{X}}\|\mathbf{A}\mathbf{X}\|. \qquad (19)$$

The norm function $p(\mathbf{X}) = \|\mathbf{A}\mathbf{X}\|$ is convex and the function in (19) conforms to the general formulation in Definition 1. When $\|\cdot\|$ in (19) 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 [5], [16]

$$f(\mathbf{X}) = |\tan(\theta)| = \left| \frac{\mathbf{x} \times \mathbf{r}}{\mathbf{x}^\top \mathbf{r}} \right|, \qquad (20)$$

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

$$F_s = \frac{1}{M} \sum_i f_i^2(\mathbf{X}), \qquad (21)$$

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 [5].

Hartley and Schaffalitzky [5] proposed using the pointwise maximum of the reprojection errors as the cost function

$$F_\infty(\mathbf{X}) = \max_i f_i(\mathbf{X}). \qquad (22)$$

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

$$F_\infty(\mathbf{X}) = \max_i f_i^w(\mathbf{X}). \qquad (23)$$

It was shown in [5] 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 [5], $F_\infty(\mathbf{X})$ is sensitive to outliers.

To deal with the outliers, we propose using the *pointwise mth* smallest reprojection errors as the cost function

$$F_m(\mathbf{X}) = \underset{i}{m\text{th}} f_i(\mathbf{X}). \qquad (24)$$

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 [12], which can handle noisy measurements with up to 50 percent of outliers.

3 MINIMIZING THE COST FUNCTION

Both F_∞ and F_m are constructed from point-wise operations on a family of functions. It is not clear how classical gradient-based approaches can be applied to minimizing them. Random line search in the parameter space was proposed in [5] to minimize F_∞, and random sampling [12], [22] is often used to detect outliers and to minimize F_m. These randomized approaches converge only in a probabilistic manner. Moreover, random line search is not scalable when the unknowns are high-dimensional.

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 $F(\mathbf{X})$

$$S_\alpha = \{ \mathbf{X} \mid F(\mathbf{X}) \leq \alpha \}. \qquad (25)$$

If S_α is nonempty, 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:

$$\begin{aligned} &\text{find } \mathbf{X}, \\ &s.t. \ \mathbf{X} \in S_\alpha. \end{aligned} \qquad (26)$$

Based on the above observation, we can use the bisection algorithm (see [8]) 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 reprojection 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.

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 (26), 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 (26) becomes a convex feasibility problem, as a result the function $F(\mathbf{X})$ can be minimized efficiently (see [8], Algorithm 4.1, p. 145). A function with such convex α-sublevel set is called a *quasiconvex* function.

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 (26).

5:     **if** (26) is feasible, **then** $h = \alpha$;

6:     **else** $l = \alpha$.

7: **end while**

Fig. 3. Classical bisection algorithm (see [8]) to pin down the optimal value by searching in the one-dimensional range domain.

Definition 2 (see [8]). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is **quasiconvex** if its domain $dom(f)$ and all its sublevel sets,*

$$S_\alpha = \{x \in dom(f) \,|\, f(x) \leq \alpha\} \tag{27}$$

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 dashline segment $\overline{AB}$ that lies below the function indicates the nonconvexity of the function.

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

Theorem 2. *A reprojection error function that conforms to the general form defined in (8) is a quasiconvex function.*

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

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

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 $dom(f) = \{\mathbf{X} \,|\, q(\mathbf{X}) > 0\}$ and S_α are all convex, we conclude that $f(\mathbf{X})$ is quasiconvex. $\square$

3.3 Minimizing Cost Function f_∞

We have shown that the reprojection error function is quasiconvex. The pointwise maximum of quasiconvex functions is also quasiconvex.

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

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

$$\begin{aligned} S_\alpha = \{\mathbf{X} \,|\, & q_i(\mathbf{X}) > 0; \; p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq 0; \\ & i = 1, \cdots, N\}. \end{aligned} \tag{28}$$
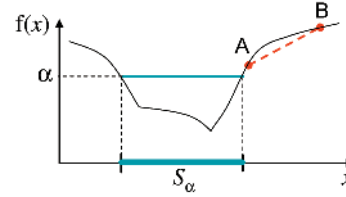

Fig. 4. A quasiconvex function. All of its α-sublevel sets $\{S_\alpha\}$ are convex. But this quasiconvex function is not convex, as can be seen from the line segment $\overline{AB}$ that lies below the function.

The feasibility problem of the bisection algorithm in (26) can now be solved by the following *convex* program:

$$\begin{aligned} \min_{\mathbf{X}, \gamma} \; & \gamma \\ s.t. \; & -q_i(\mathbf{X}) + \epsilon \leq \gamma, \\ & p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma, \\ & i = 1, \dots, N. \end{aligned} \tag{29}$$

Here, ϵ is a small positive number.[2]

Equation (29) first relaxes the inequality constraints (in (28)) by γ, then minimizes the relaxation. Denote γ^* the optimal value of (29). If $\gamma^* \leq 0$, then S_α of $F_\infty(\mathbf{X})$ is nonempty and the problem in (26) is feasible; otherwise, (26) is infeasible. Note that we do not need to solve (29) 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 [5]. To deal with outliers, we use the robust cost function $F_m(\mathbf{X})$, which is defined as the mth smallest reprojection error (see (24)). $F_m(\mathbf{X})$ 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 $\mathbf{X}_0$.

Theorem 4. 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

$$f_1(\mathbf{X}_0), f_2(\mathbf{X}_0), \cdots, f_N(\mathbf{X}_0) \tag{30}$$

into the nondecreasing order

$$f_{(1)}(\mathbf{X}_0) \leq \cdots \leq f_{(m)}(\mathbf{X}_0) \leq \cdots \leq f_{(N)}(\mathbf{X}_0). \tag{31}$$

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 (31) must therefore satisfy

2. We set $\epsilon = 1$ in practice.

$f_{(i)}(\mathbf{X}_0) \leq \alpha$. As a result, $\mathbf{X}_0$ belongs to the intersection of the m α-sublevel sets of the first m functions in (31).

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

$$f^{(i)}(\mathbf{X}_0) \leq \alpha, \quad i = 1, \cdots, m. \qquad (32)$$

Now, if $F_m(\mathbf{X}_0) = f_{(m)}(\mathbf{X}_0) > \alpha$, then from the sorted sequence in (31) we know that the number of less-than-α reprojection errors is less than m. This contradicts (32) 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

The significance of Theorem 4 is that the feasibility problem (in the bisection algorithm to minimizing F_m) can now be rewritten as

$$\begin{aligned} &\text{find } \mathbf{X} \\ &s.t. \ \mathbf{X} \in_m \{S_\alpha^1, S_\alpha^2, \cdots, S_\alpha^N\}. \end{aligned} \qquad (33)$$

In other words, we need to determine if there exist m α-sublevel sets whose common intersection is nonempty.

Exact global minimization. The formulation of (33) allows us to locate the *global* minimum of F_m, by checking the feasibility of every possible group of m sublevel sets in each iteration of the bisection algorithm, where for each group its feasibility can be exactly determined by the convex program of (29). In the worst case, this requires $\binom{N}{m}$ convex programs to solve (33) in order to locate the global minimum of F_m, which is good for small N, but could be computational too expensive for large N. When N is large, we can use either integer programming or its convex approximation.

Minimizing number of infeasibility.

Theorem 5. *The feasibility problem in (33) can be formulated exactly by integer programming:*

$$\begin{aligned} &\min_{\mathbf{X}, \gamma} \gamma_1 + \gamma_2 + \cdots + \gamma_N \\ &s.t. \ - q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ &\qquad p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ &\qquad \gamma_i = \{0, v\}, \\ &\qquad i = 1, \ldots, N. \end{aligned} \qquad (34)$$

Here, $v > 0$ is an arbitrary large positive integer.

Given optimal values $(\mathbf{X}^*, \gamma^*)$ of the above minimization problem, if $\gamma_i = 0$, then the ith reprojection error $f_i(\mathbf{X}) \leq \alpha$, i.e., $\mathbf{X}$ is inside the ith α-sublevel set. On the other hand, if $\gamma_i = v$, then $\mathbf{X}$ is outside the ith α-sublevel set. When $(\gamma_1 + \gamma_2 + \cdots + \gamma_N)$ is minimized, we obtain a solution $\mathbf{X}^*$, where the number of infeasible constraints is minimized. In other words, we seek a solution which maximizes the number of α-sublevel sets who have nonempty common intersection. If this number is larger than m, then (33) is feasible; otherwise, it is infeasible.

Minimizing sum of infeasibility. For large-scale problems, where integer programming may become computational expensive or prohibited, we can use LP or SOCP as an approximation.

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

$$\begin{aligned} &\min_{\mathbf{X}, \gamma} \gamma_1 + \gamma_2 + \cdots + \gamma_N \\ &s.t. \ - q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ &\qquad p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ &\qquad \gamma_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \qquad (35)$$

Denote $\gamma^* = (\gamma_1^*, \gamma_2^*, \cdots, \gamma_N^*)$ the optimal value of the above convex program achieving at $\mathbf{X}^*$. Denote g the number of zero elements in γ^*. If $g \geq m$, then the problem defined by (33) must be feasible.

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 (33) 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 (33) to be feasible. The exact conclusion about the infeasibility of (33) 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 γ^*). Theorem 6 finds the minimum sum-of-infeasibilities $\sum_i \gamma_i^*$ and uses it to approximate the minimum number of infeasibilities. Though the global minimum of F_m is not guaranteed with such approximation, the bisection algorithm gives an upper bound on the true minimum value of F_m. The reason is that the sum of infeasibility $\|\gamma\|_1 = \sum_i \gamma_i$ 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 bounded by the image size. As a result, the bisection algorithm using Theorem 6 can usually achieve a tight upper bound on the true minimum value of F_m.

Minimizing sum of weighted infeasibility. We can further improve the result by using weighted sum of infeasibility $\mathbf{w}^\top \gamma$ in (35), 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, as shown in the following convex program:

$$\begin{aligned} &\min_{\mathbf{X}, \gamma} w_1 \gamma_1 + w_2 \gamma_2 + \cdots + w_N \gamma_N \\ &s.t. \ - q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ &\qquad p_i(\mathbf{X}) - \alpha q_i(\mathbf{X}) \leq \gamma_i, \\ &\qquad \gamma_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \qquad (36)$$

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 $r_i = |\mathbf{X}_i - \frac{p_i}{q_i}|$. Note that (35) is a special case of (36). In our experiments presented in this paper, we use an iterative version of the first scheme.

3.4.3 Discussion on Minimizing F_m

F_m is a robust error function that is no longer quasiconvex and is subject to local minima. If we can determine the feasibility of its α-sublevel set (i.e., solve (26)), then we can still locate its global minima. However, solving (26) is a hard problem due to its nonconvexity. In Theorem 4, we show that determining such feasibility is equivalent to finding m (out of N) convex

cones with common intersection, as shown in (33). As a result, we can locate the global minimum of F_m for small-scale problems by checking the intersection of every m convex cones (each intersection checking is a convex programming). For larger scale problems, we have presented an approximation method (and its iterative version) to solve (33). Theorem 4 also enables us to design RANSAC-like methods to determine, in a probabilistic way, the feasibility of the α-sublevel set of F_m. For example, we can randomly choose k (a small number) cones, solving for the intersection of these k cones (which is a SOCP with k cones) and then check if such intersection is inside at least m cones (which is straightforward). The probability of obtaining the global minimum is then determined by the number of random sample sets and the fraction of outliers. We leave this as part of our future work in further investigating the use of L_∞-norm given noisy data with outliers.

3.5 Feasibility by LP or SOCP

When L_1 or L_2-norm error metric is used in defining the reprojection error function in the image domain, the convex program for feasibility becomes small-scale linear programs (LP) or second-order convex programs (SOCP), respectively. Both LP and SOCP are well-studied and existing efficient algorithms and implementations are ready to use (e.g., the Matlab built-in function *linprog* or SeDuMi [23]). In the following, we present the detailed LP and SOCP for the convex program in (35), which also applies to (29) as it is a special case of (35).

3.5.1 L_1-Norm Error Metric Leads to LP

When L_1-norm is used to compute the reprojection error in image domain, the convex program in (35) becomes the following linear program:

$$\min_{\mathbf{X},\gamma} \gamma_1 + \gamma_1 + \cdots + \gamma_N$$
$$\begin{aligned} s.t. \ & - q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ & - \alpha q_i(\mathbf{X}) + p_{ui}(\mathbf{X}) - p_{vi}(\mathbf{X}) \leq \gamma_i, \\ & - \alpha q_i(\mathbf{X}) + p_{ui}(\mathbf{X}) + p_{vi}(\mathbf{X}) \leq \gamma_i, \\ & - \alpha q_i(\mathbf{X}) - p_{ui}(\mathbf{X}) - p_{vi}(\mathbf{X}) \leq \gamma_i, \\ & - \alpha q_i(\mathbf{X}) - p_{ui}(\mathbf{X}) + p_{vi}(\mathbf{X}) \leq \gamma_i, \\ & \gamma_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{37}$$

Here, p_{ui}, p_{vi}, and q_i are all linear functions of $\mathbf{X}$ (see (11) for the definition).

3.5.2 L_2-Norm Error Metric Leads to SOCP

When L_2-norm is used to compute the reprojection error in image domain, (35) becomes

$$\min_{\mathbf{X},\gamma} \gamma_1 + \gamma_1 + \cdots + \gamma_N$$
$$\begin{aligned} s.t. \ & - q_i(\mathbf{X}) + \epsilon \leq \gamma_i, \\ & \|\mathbf{A}_i\mathbf{X}\|_2 \leq \alpha q_i(\mathbf{X}) + \gamma_i, \\ & \gamma_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{38}$$

Here, $\mathbf{A}_i$ is a 2×3 matrix defined in (17). $\alpha q_i(\mathbf{X}) + \gamma_i$ is a linear function of $\mathbf{X}$. Therefore, the inequality

$$\|\mathbf{A}_i\mathbf{X}\|_2 \leq \alpha q_i(\mathbf{X}) + \gamma_i \tag{39}$$
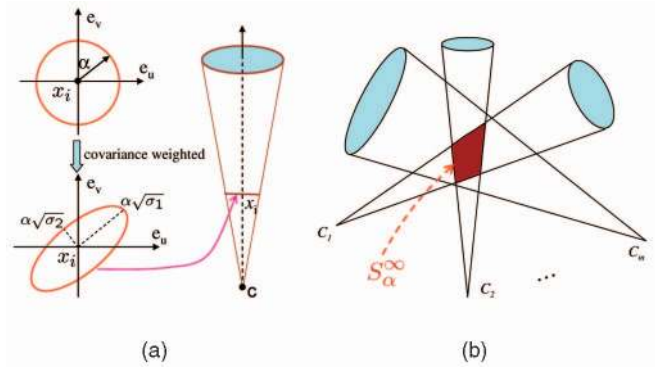

Fig. 5. Geometric illustration of 3D reconstruction using quasiconvex optimization. Here, L_2 norm is used in the reprojection error function. (a) The shape and size of each cone are determined by α and covariance matrix $\mathbf{Q}_i$. (b) The algorithm seeks minimum α such that the cones have nonempty **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.

defines a second-order convex cone [8]. As a result, (38) is a second-order cone programming (SOCP).

The minimization algorithm we presented in this section has intuitive geometric interpretation. We use uncertainty-weighted multiview triangulation as an example to illustrate. For each 2D 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 $\mathbf{Q}_i = \mathbf{U} \operatorname{diag}(\sigma_1, \sigma_2)\mathbf{U}^\top$, as shown in Fig. 5a. 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 $\mathbf{U}$. 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 $F_\infty(\mathbf{X})$ is the intersection of all α-sublevel sets $\{S_i^w\}$. Minimizing the covariance-weighted cost $F_\infty(\mathbf{X})$ is therefore equivalent to determining if there exists common intersection of the convex elliptical cones $\{S_i^w\}$, as shown in Fig. 5b. The bisection algorithm then seeks a minimum value α such that the common intersection of the convex elliptical cones $\{S_i^w\}$ is nonempty. To minimize the robust function $F_m(\mathbf{X})$, the algorithm determines whether or not there exist m-out-of-N convex cones that have nonempty common intersection.

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.
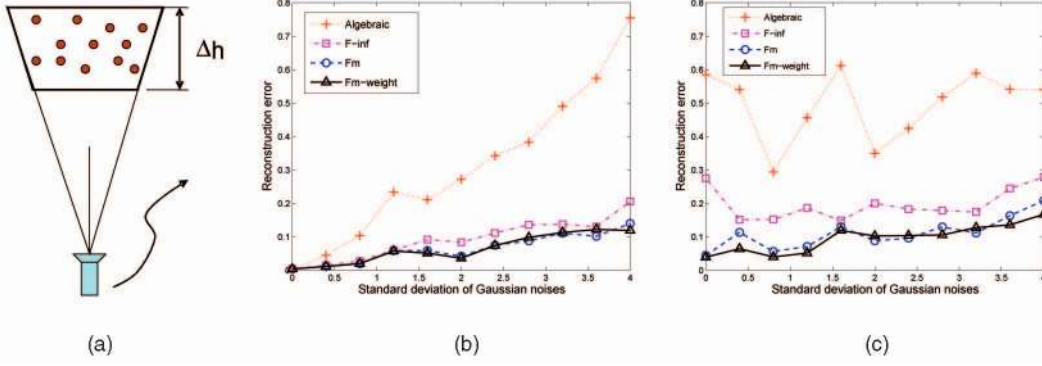
Fig. 6. Multiview triangulation: synthetic data (image size: 200×200). (a) The camera is rotating and translating, imaging a 3D scene consists of 40 points. (b) Reconstruction errors (normalized by (40)) with zero-mean Gaussian noises added to 2D point coordinates. (c) Reconstruction errors under both Gaussian noises and outliers. The unit is pixel in the standard deviation of Gaussian noises.

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

4 EXPERIMENTS

We apply our quasiconvex optimization algorithm to multiview triangulation and sequential structure from motion (SFM) (see [24]), and evaluate the performance using both synthetic and real data. Using multiview triangulation and planar homography estimation as two examples, we also show that, when (directional) uncertainties are available, the use of covariance-weighted error functions can greatly improve the reconstruction results.

4.1 Multiview Triangulation: Synthetic Data

The synthetic scene contains 40 3D points, distributed at different depth, that are imaged by a moving synthetic camera, as shown in Fig. 6a. 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 (36). The reconstruction results from the algebraic approach (see Section 2.1.1) are included for comparison purpose.

Fig. 6 shows the average reconstruction errors, where Fig. 6b shows results when Gaussian noises are added to the 2D positions at increasing variances, and Fig. 6c shows the results with both Gaussian noises and 50 percent of outliers. The reconstruction error is normalized by

$$err = \frac{\|\mathbf{Z} - \mathbf{Z}_T\|_2}{\|\mathbf{Z}_T\|_2}, \qquad (40)$$

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 appears to successfully minimize 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.

4.2 Multiview Triangulation: Real Data with "Ground Truth"

We use the *corridor* sequence[3] in which the camera is moving forward along the corridor. Fig. 7a 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. Figs. 7b and 7c 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 the value of F_∞ is determined by outliers. Its performance depends on the "strength" of the outliers. Fig 7d 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 outlier strength (in 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.

4.3 Application: Sequential Structure from Motion

Our target application is vision-aided small and micro aerial vehicle (SMAV) navigation, in which sequential SFM [24] is applied to estimate both the camera motions and the 3D. We apply our F_m minimization method to the sequential SFM.

There are two steps in our sequential reconstruction. In the first step, we estimate the essential matrix (the camera is intrinsically calibrated) from feature correspondences. The camera rotation and translation are then recovered from the essential matrix. We observe that the estimation of camera rotation is much better than the estimation of the camera translation, especially in a typical SMAV motion which has a major forward-motion component. Therefore, in a second step, the camera translations and the 3D of the feature points are estimated from multiple frames using F_m minimization, i.e., reconstruction with known rotations (see Section 1.1.3). In the near future, we will use IMU sensors in the SMAV navigation and control. IMU sensors typically provide good
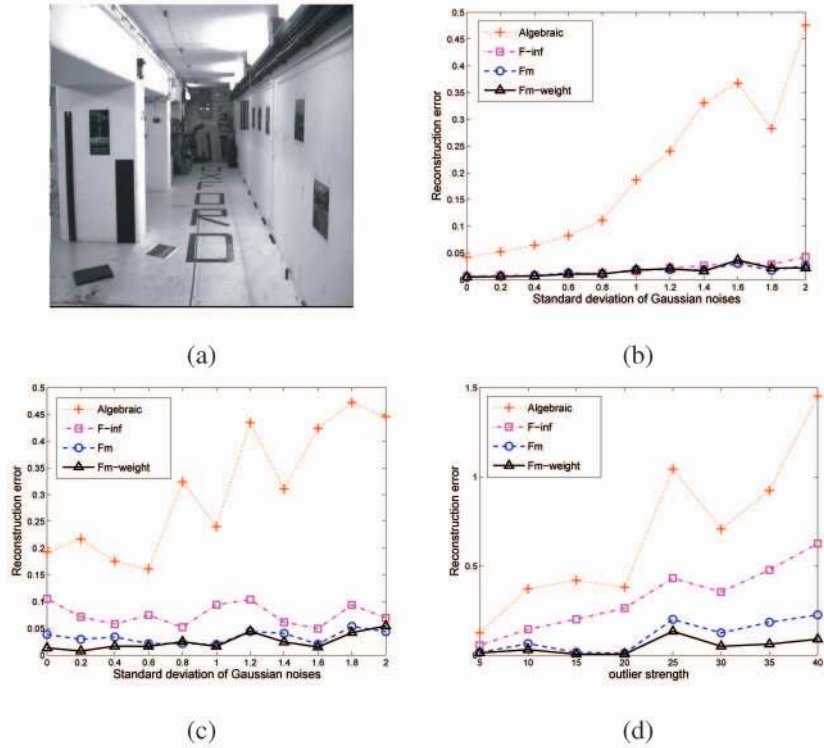
3. http://www.robots.ox.ac.uk/~vgg/data1.html.

Fig. 7. Multiview triangulation: corridor sequence (image size: 512×512). (a) The first image of this 11-frame sequence. (b) Reconstruction errors (normalized by (40)) 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). The unit is pixel in the standard deviation of Gaussian noises.

rotational estimations but very noisy positional estimations. The two-step approach is suitable for fusing such IMU sensor measurements with image measurements.

A 450-frame image sequence is taken by a intrinsically calibrated mini camera that was moved around by hand in an office. Fig. 8 (first column) shows the first, middle, and last frames in this sequence. Notice that the images captured by the mini camera have low quality, which results in noisy 2D feature tracking.

Fig. 8. Multiview triangulation in sequential SFM. The camera is moved (largely forward motion) around inside the office. The first column shows the first, middle, and last frame of the 450-frame sequence (image size 360×240), with tracked points superimposed. The image on the right is the 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.

Fig. 8 (second column) 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 Directional Uncertainty

We use planar homography estimation and multiview triangulation as two example applications to evaluate our algorithm for handling *directional* uncertainties. Using synthetic and real data, we compare the performances of quasiconvex minimization of F_∞ with and without uncertainty model. The uncertainty-weighted version of F_∞ is denoted by F_∞^w. The results from normalized linear algorithm are also included for comparison purpose.

4.4.1 Homography Estimation with Directional Uncertainty: Synthetic Data

We [6] and Kahl [7] have shown that planar homography can be estimated by SOCP minimization. Kahl [7] presented experimental results. In this section, we extend the works by taking uncertainty into account.

Fig. 9 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 H_T (the
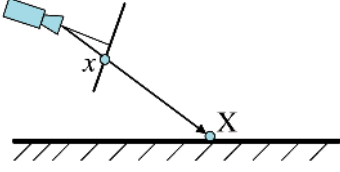
Fig. 9. The set up for synthetic data generation for homography estimation. The camera is looking at points on the ground plane. The image coordinates are normalized such that the camera focal length $f = 1$.

ground truth homography between the ground plane and the image plane) may become very small. In all the algorithms being compared, we do not assume $h_{33} = 1$. Furthermore, in the following experiments, we use two images and assume the noise is only in the second image.

We randomly generate 20 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

$$F_\infty(\mathbf{X}) = \max_i d(\tilde{\mathbf{x}}_i, \mathbf{H}\mathbf{x}_i), \qquad (41)$$

  where $d(\cdot, \cdot)$ denotes the euclidean distance.
- Root of Mean Squares (RMS) of reprojection errors.
- Maximum covariance-weighted reprojection error, defined in (23) and denoted by $F_\infty^w(\mathbf{x})$.

- Error in H defined as

$$e_H = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2(\mathbf{x}_i, \mathbf{H}\mathbf{x}_i)}, \qquad (42)$$

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

We apply our algorithm to estimate H from this synthetic data. We repeat the experiments for 20 times, and report the average error. Fig. 10 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 Fig. 10a, both F_∞ and F_∞^w have similar RMS error. Normalized linear algorithm has similar RMS error when r is small, but becomes unreliable when $r \geq 10$.

From Fig. 10d, which compares the estimated H against ground truth $\mathbf{H}_T$ 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, the performance of normalized linear algorithm becomes unreliable.

4.4.2 Multiview Triangulation with Directional Uncertainty: Synthetic Data

We use the set up in Fig. 6a to generate the synthetic data for multiview triangulation with uncertainty. The camera is rotating and translating and takes 10 consecutive views of 20 3D points located at different depth. Directional Gaussian noises are added to the locations of image points.
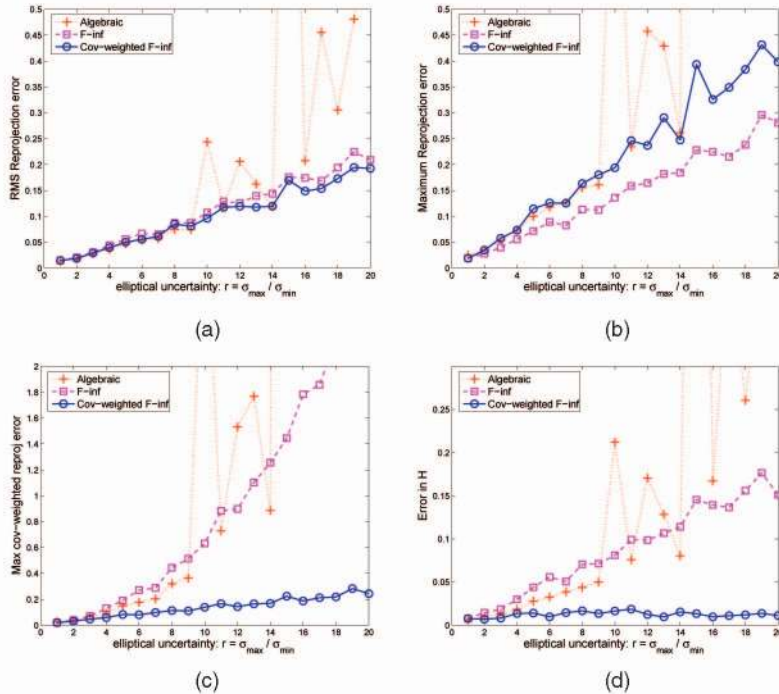

Fig. 10. Results from planar homography estimation. (a) RMS error. (b) Maximum of reprojection error. (c) Maximum of covariance-weighted reprojection error. (d) Error in H.
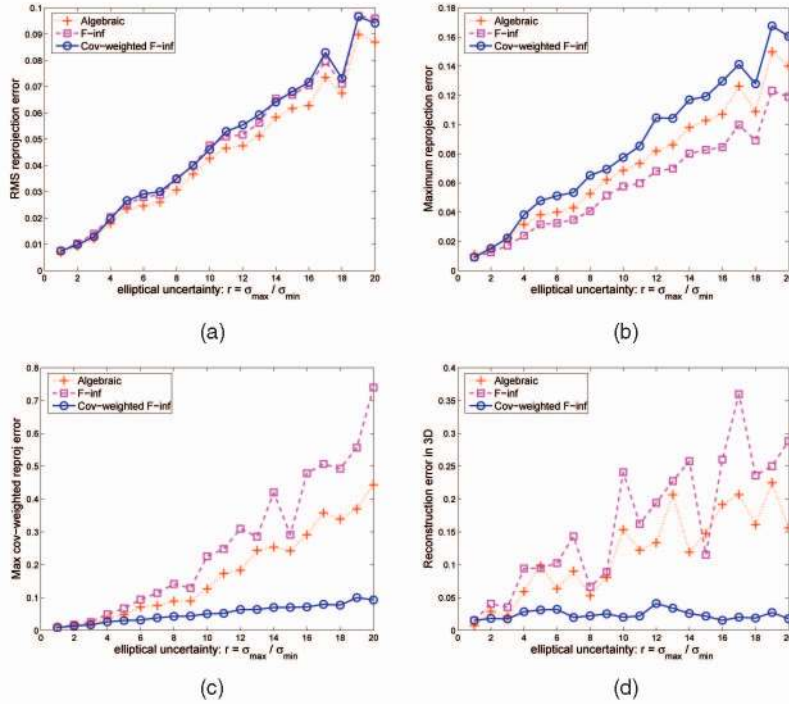
Fig. 11. Results from multiview triangulation. (a) RMS error. (b) Maximum of reprojection error. (c) Maximum of covariance-weighted reprojection error. (d) Reconstruction error compared against ground truth 3D.

Fig. 11 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 error metric of (40).

4.4.3 Infinite Directional Uncertainty

The performance of F_∞^w does not degrade even when r, the ellipticity of the noises, goes to essentially infinity, as can be seen from Table 1. This fact indicates that the normal optical flow can be modeled 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.

4.4.4 Homography Estimation with Directional Uncertainty: Real Data

We apply our algorithm to estimate the inter-image homography using real image data. Fig. 12 shows two input images. Here, Fig. 12a shows the first image. The other images are obtained by applying a known and gradually-changed

planar homography to Fig. 12a. 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 matrix defined in (13). The elliptical uncertainties for some feature points are shown in Fig. 12a.

Fig. 13 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 weights 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.
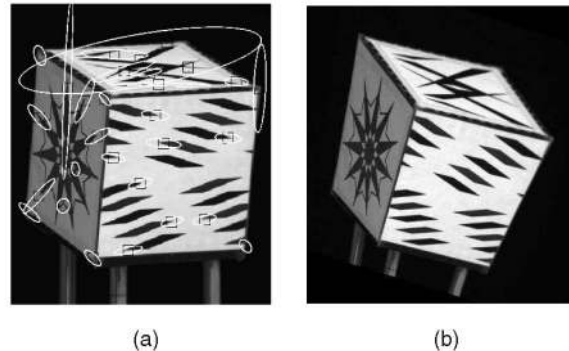
TABLE 1
Results under Infinite Elliptical Uncertainty $r = 10^5$

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

The table shows e_H for homography estimation and e_{3D} for triangulation.

Fig. 12. Two input images. (a) First image, where elliptical uncertainties for some feature points are shown. (b) Last image.
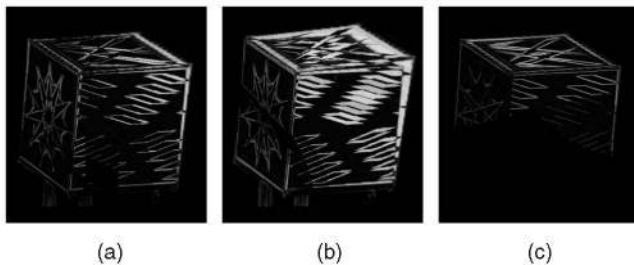
Fig. 13. Residual image by applying the estimated homography. Note that residuals are scaled up for visibility. The average pixel residual is shown as e for each case too. (a) Normalized linear algorithm, $e = 15.0798$. (b) Minimizing F_∞, $e = 28.2082$. (c) Minimizing F_∞^w, $e = 5.8896$.

5   CONCLUSION

We have presented a novel quasiconvex optimization framework to geometric reconstruction problems, which are formulated as a small number of small-scale convex programs, more specifically, Second-Order Cone Program (SOCP) when the reprojection error in each image is measured by L_2-norm (euclidean distance), and Linear Program (LP) when the reprojection error is measured by L_1-norm. We derived the algorithm based on sound mathematical grounds, and the algorithm requires very little of parameter tuning. The final algorithm is simple, robust, and has very intuitive geometric interpretation. In contrast to previous methods, our algorithm is deterministic and guarantees a predefined accuracy of the global minimization result.

The robust error function F_m is used to handle outliers. In general, F_m is not a quasiconvex function. To obtain its global minimum using the bisection algorithm, one needs to solve the feasibility problem in (26), which is a hard problem due to its nonconvexity. We show in Theorem 4 that the feasibility problem of (26) is equivalent to finding m (out of N) convex cones with common intersection (see (33)). For small-scale problems (e.g., 5-10 view 3D triangulation), we can afford to check the intersection of all possible m cones, thus achieve the global minimum. For larger scale problems, we need to resort to approximation methods, in such cases the global minimum is no longer guaranteed. We have presented one approximation solution (and its iterative version), which is a convex approximation of integer programming. For the scale of the problems we presented in this paper, the iterative approximation solution works well. In the experiment with synthetic data, the iterative version is able to find all the outliers, while the noniterative version fails to identify some outliers at about 5 percent to 10 percent of the time (it depends on the percentage of the outliers, as well as the strength of the outliers and we have reported the average reconstruction error). As part of our future work, we are also currently investigating a RANSAC-like method to solve the feasibility problem of (33) in a probabilistic way (see Section 3.4.3).

Our quasiconvex optimization method can take into account (directional) uncertainties in a straightforward and seamless way. Since line features can be represented as a point feature with infinite uncertainty in the line direction, our method allows points and lines to be simultaneously used in reconstruction.

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 [25], multibaseline stereo reconstruction, and efficient bundle adjustment [26] in structure from motinon. We are also extending our approach to multiview triangulation without feature correspondences.

REFERENCES

[1]   R. Hartley and P. Sturn, "Triangulation," *Computer Vision and Image Understanding,* vol. 68, no. 2, pp. 146-157, 1997.
[2]   R. Haralick, C. Lee, K. Ottenberg, and M. Nolle, "Review and Analysis of Solutions of the 3-Point Perspective Pose Estimation Problem," *Int'l J. Computer Vision,* vol. 13, no. 3, pp. 331-356, 1994.
[3]   L. Quan and Z. Lan, "Linear n $\geq$ 4-Point Pose Determination," *Proc. Int'l Conf. Computer Vision,* pp. 778-783, 1998.
[4]   R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge Univ. Press, 2000.
[5]   R.I. Hartley and F. Schaffalitzky, "L_∞ Minimization in Geometric Reconstruction Problems," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. (I)504-509, 2004.
[6]   Q. Ke and T. Kanade, "Quasiconvex Optimization for Robust Geometric Reconstruction," *Proc. Int'l Conf. Computer Vision,* pp. II:986-993, Oct. 2005.
[7]   F. Kahl, "Multiple View Geometry and the l_∞-Norm," *Proc. Int'l Conf. Computer Vision,* pp. II:1002-1009, Oct. 2005.
[8]   S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.
[9]   D. Nistér, "Automatic Dense Reconstruction from Uncalibrated Video Sequences," PhD Thesis, Royal Inst. of Technology KTH, Stockholm, Sweden, 2001.
[10]   H. Stewénius, F. Schaffalitzky, and D. Nistér, "How Hard Is 3-View Triangulation Really?" *Proc. Int'l Conf. Computer Vision,* pp. I:686-693, 2005.
[11]   C. Tomasi and T. Kanade, "Shape and Motion from Image Streams Under Orthography: A Factorization Method," *Int'l J. Computer Vision,* vol. 9, no. 2, pp. 137-154, 1992.
[12]   Z. Zhang, "Determining the Epipolar Geometry and Its Uncertainty: A Review," *Int'l J. Computer Vision,* vol. 27, no. 2, pp. 161-195, 1998.
[13]   D.D. Morris and T. Kanade, "A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models," *Proc. Int'l Conf. Computer Vision,* pp. 696-702, 1998.
[14]   M. Irani and P. Anandan, "Factorization with Uncertainty," *Proc. European Conf. Computer Vision,* pp. 539-553, 2000.
[15]   Q. Ke and T. Kanade, "Uncertainty Models in Quasiconvex Optimization for Geometric Reconstruction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1199-1205, 2006.
[16]   J. Oliensis, "Exact Two-Image Structure from Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1618-1633, Dec. 2002.
[17]   D. Diel, P. DeBitetto, and S. Teller, "Epipolar Constraints for Vision-Aided Inertial Navigation," *Proc. IEEE Workshop Motion and Video Computing,* pp. 221-228, 2005.
[18]   M. Uyttendaele, A. Criminisi, S. Kang, S. Winder, R. Hartley, and R. Szeliski, "High-Quality Image-Based Interactive Exploration of Real-World Environments," *IEEE Computer Graphics and Applications,* pp. 52-63, 2004.
[19]   R.I. Hartley, "Chirality," *Int'l J. Computer Vision,* vol. 26, no. 1, pp. 41-61, 1998.
[20]   J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 593-600, 1994.
[21]   R.M. Steele and C. Jaynes, "Feature Uncertainty Arising from Covariant Image Noise," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1063-1070, 2005.

[22] P. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, no. 1, pp. 138-156, 2000.
[23] J. Sturm, "Using SeDuMi 1.02, A MATLAB Toolbox for Optimization Over Symmetric Cones," *Optimization Methods and Software,* vol. 11-12, pp. 625-653, 1999.
[24] P.A. Beardsley, A. Zisserman, and D.W. Murray, "Sequential Updating of Projective and Affine Structure from Motion," *Int'l J. Computer Vision,* vol. 23, pp. 235-259, June 1997.
[25] K.N. Kutulakos and S.M. Seitz, "A Theory of Shape by Space Carving," *Int'l J. Computer Vision,* vol. 38, no. 3, pp. 199-218, 2000.
[26] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment—A Modern Synthesis," *Springer Lecture Notes in Computer Science,* vol. 1883, pp. 298-372, 2000.

Qifa Ke received the BS degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 1994, the MS degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1997, and the PhD degree in computer science from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2003. From 2003 to 2006, he was a systems scientist in the Computer Science Department of Carnegie Mellon University. He is now a research scientist at Ricoh Innovations Inc., California Research Center. Dr. Ke's research interests include document analysis, computer vision, pattern recognition, and machine learning. He is a member of IEEE.

Takeo Kanade received the PhD degree in electrical engineering from Kyoto University, Japan, in 1974. He is a UA Helen Whitaker University Professor of Computer Science and Robotics at Carnegie Mellon University. After holding a junior faculty position in the Department of Information Science, Kyoto University, he joined Carnegie Mellon University in 1980, where he was the director of the Robotics Institute from 1992 to 2001. Dr. Kanade has worked in multiple areas of robotics: computer vision, multimedia, manipulators, autonomous mobile robots, and sensors. He has written more than 250 technical papers and reports in these areas and has more than 15 patents. He has been the principal investigator of more than a dozen major vision and robotics projects at Carnegie Mellon. He has been elected to the National Academy of Engineering and to American Academy of Arts and Sciences. He is a fellow of the IEEE, a fellow of the ACM, a founding fellow of the American Association of Artificial Intelligence (AAAI), and the former and founding editor of the *International Journal of Computer Vision*. He has received several awards, including the C & C Award, Joseph Engelberger Award, Allen Newell Research Excellence Award, JARA Award, Marr Prize Award, and FIT Funai Accomplishment Award. Dr. Kanade has served on government, industry, and university advisory or consultant committees, including Aeronautics and Space Engineering Board (ASEB) of National Research Council, NASA's Advanced Technology Advisory Committee, PITAC Panel for Transforming Healthcare Panel, the Advisory Board of Canadian Institute for Advanced Research.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.