Fast and Robust Estimation for Unit-Norm Constrained Linear Fitting Problems

Daiki Ikami Toshihiko Yamasaki Kiyoharu Aizawa
The University of Tokyo, Japan
{ikami, yamasaki, aizawa}@hal.t.u-tokyo.ac.jp

Abstract

M-estimator using iteratively reweighted least squares (IRLS) is one of the best-known methods for robust estimation. However, IRLS is ineffective for robust unit-norm constrained linear fitting (UCLF) problems, such as fundamental matrix estimation because of a poor initial solution. We overcome this problem by developing a novel objective function and its optimization, named iteratively reweighted eigenvalues minimization (IREM). IREM is guaranteed to decrease the objective function and achieves fast convergence and high robustness. In robust fundamental matrix estimation, IREM performs approximately 5-500 times faster than random sampling consensus (RANSAC) while preserving comparable or superior robustness.

1. Introduction

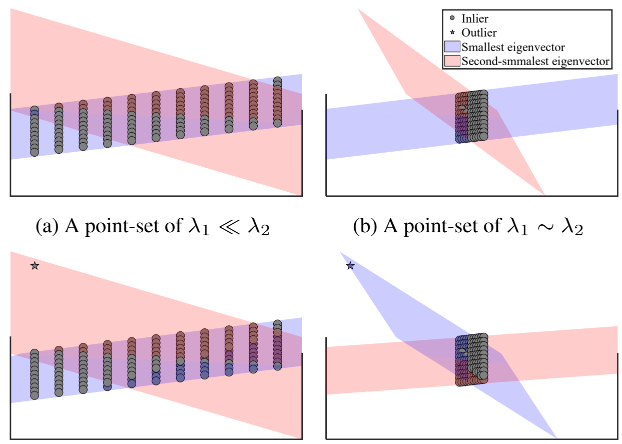
The robust estimation of unit-norm constrained linear fitting (UCLF) problem can be formulated as

$$\min_{\mathbf{x}} \sum_{i}^{n} p(r_i) \;\; s.t. \;\; r_i = \mathbf{a}_i^\top \mathbf{x}, \|\mathbf{x}\|_2^2 = 1, \tag{1}$$

where $p(r)$ is a robust loss function. This formulation is often used in cases such as conic estimation [24] and fundamental matrix estimation [11].

M-estimation using the iteratively reweighted least squares (IRLS) algorithm [12] is one of the most popular techniques in robust linear regression. Although we can use this technique for the robust UCLF problem, it is ineffective because of a poor least squares (LS) solution, which is used as the initial solution for IRLS.

We will demonstrate why the initial LS solution is such a poor solution to the UCLF problem. Consider the robust plane fitting problem, which can be formulated as Eq. (1)[1]. The plane fitting results by LS are shown in Fig. 1. The LS solution in Fig. 1(d) is too distant from the solution of



(a) A point-set of $\lambda_1 \ll \lambda_2$ (b) A point-set of $\lambda_1 \sim \lambda_2$

(c) Add a single outlier for Fig. 1(a). (d) Add a single outlier for Fig. 1(b).

Fig 1: LS plane fitting for different point-sets with different λ, where λ_j is the jth eigenvalue of the point-set matrix $\mathbf{A}$ in ascending order. Circles and stars denote inliers and outliers, respectively. The blue planes denote the smallest eigenvector of the matrix $\mathbf{A}$, which minimize the quadratic loss of the distances. The red planes are given by the second-smallest eigenvector, which minimize the quadratic loss under the orthogonality to the smallest eigenvector. The LS solution is good for an initial solution of IRLS in the case of Fig. 1(c); however, this does not apply to the case shown in Fig. 1(d). Instead, the red plane is an appropriate initial solution for robust estimation.

robust estimation. In addition, UCLF is a nonconvex optimization problem that has multiple local minima; therefore, existing iterative methods starting from a poor initial solution frequently become trapped in a poor local minimum.

In this study, we propose a novel approximate objective function of the M-estimator to overcome the poor initial solution problem by utilizing a *hidden* good initial solution, as in the red plane of Fig. 1(d). Our approximate objective function can be minimized by a convergence-guaranteed algorithm, named iteratively reweighted eigenvalues minimization (IREM). IREM is not guaranteed to find the global optimal solution because of the nonconvexity of the opti-

[1] Let $\mathbf{a}_i = [x_i, y_i, z_i, 1]^\top$ be the x-, y-, and z-coordinate values of data points and let $\mathbf{x} = [a, b, c, d]^\top$ be the plane parameters, with the data points on the plane satisfying $\mathbf{a}_i^\top \mathbf{x} = ax_i + by_i + cz_i + d = 0$. Therefore, robust plane fitting can be formulated as Eq. (1)

Table 1: Representative loss functions $p(r)$ and their corresponding functions $\psi(r)$ in Eq. (3) and $f(w)$ in Eq. (4).

	Least Absolute	Huber	Cauchy	Talwar
$p(r)$	$2\lvert r\rvert$	$r^2 \quad (\lvert r\rvert \le c)$ $2c\lvert r\rvert - c^2 \quad (\lvert r\rvert > c)$	$c^2 \log\left(1 + \left(\frac{r}{c}\right)^2\right)$	$r^2 \quad (r^2 \le c)$ $c \quad (r^2 > c)$
$\psi(r)$	$\frac{1}{r}$	$1 \quad (\lvert r\rvert \le c)$ $\frac{1}{r} \quad (\lvert r\rvert > c)$	$\frac{1}{1+(r/c)^2}$	$1 \quad (r^2 \le c)$ $0 \quad (r^2 > c)$
$f(w)$	$1/w$ $(0 \le w)$	$c^2(1/w - 1)$ $(0 \le w \le 1)$	$c^2(w - \log w - 1)$ $(0 \le w)$	$c(1 - w)$ $(w \in \{0, 1\})$

mization problem. However, the results of experimental evaluations demonstrated that IREM finds a good solution even for a high outlier rate of 40–70%. IREM requires only $O(n)$ computation during each iteration, where n is the total number of observation data points, and requires about 5–20 iterations to reach convergence.

Our contribution is as follows:

1. We present why IRLS does not work well in the UCLF problem. It is because of a poor initial solution from small eigenvalues of the observation matrix.
2. We propose a novel objective function and its optimization algorithm. Our algorithm requires $O(n)$ during each iteration and about 5-20 iterations for convergence even at high outlier rate.
3. In robust fundamental matrix estimation, our algorithm achieves 5–500 times faster estimation than traditional RANSAC with comparable robustness.

Notation: Bold uppercase letters ($\mathbf{X}$), bold lowercase letters ($\mathbf{x}$), and plain lowercase letters (x) denote matrices, column vectors, and scalars, respectively.

2. Related works

2.1. Robust Estimation

2.1.1 M-estimator

M-estimators use a robust loss function $p(r)$ to reduce the influence of large errors. The optimization problem is formulated as $\min_{\mathbf{x}} \sum_i p(r_i(\mathbf{x}))$, where $r_i(\mathbf{x})$ is ith residual and $\mathbf{x}$ is the model parameter vector. The popular robust loss functions are summarized in Table 1.

The robust optimization problem can be solved by the IRLS algorithm [12] which iterates the following two steps:

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \sum_i^n w_i^{(t)} \left(r_i(\mathbf{x})\right)^2, \tag{2}$$

$$w_i^{(t+1)} = \psi\left(r_i\left(\mathbf{x}^{(t+1)}\right)\right), \tag{3}$$

where $\psi(r)$ is the weight function derived from $p(r)$ as in Table 1. IRLS iterates Eq. (2) and Eq. (3) until convergence. Note that IRLS is a kind of majorize-minimization [13]. We can construct a majorizer of $p(r)$ by weighted least squares [2, 23, 27]:

$$\sum_i^n p(r_i) = \min_{\mathbf{w}} \sum_i^n \left(w_i r_i^2 + f(w_i)\right), \tag{4}$$

where w_i is the weight of the ith datum and $f(w)$ is a function derived from $p(r)$ as in Table 1. Note that IRLS is equivalent to an alternating optimization strategy for Eq. (4). For the Talwar loss example, the optimization problem based on Eq. (4) is

$$\min_{\mathbf{x},\mathbf{w}} \sum_i^n \left(w_i r_i(\mathbf{x})^2 + c(1 - w_i)\right) \text{ s.t. } w_i \in \{0, 1\}. \tag{5}$$

We can solve Eq. (5) for w_i as

$$w_i = \begin{cases} 1 & (r_i^2 \le c) \\ 0 & (r_i^2 > c) \end{cases} \tag{6}$$

From Eq. (6), we can find that Eq. (5) is equivalent to the Talwar loss minimization problem, and alternating minimization for Eq. (5) is equivalent to IRLS, as in Eq. (3).

2.1.2 RANSAC

RANSAC [10] is one of the most popular robust estimation methods based on a stochastic optimization. RANSAC has two steps: a hypothesis of the model parameter is constructed by a minimal sample subset, which is randomly selected from the observed dataset. Then the score of the hypothesis is computed from all observed dataset. RANSAC iterates this process until a model parameter with sufficient score is found.

The computational cost of RANSAC is increased by outlier rate and the parameter dimension. Let w be the proportion of inliers and d is the dimension of the parameter. We can select a sample set of all inliers with the probability of w^d. Therefore we can select an all-inliers set at least once in trials n with the probability $p = 1-(1-w^d)^n$ and obtain the following formula:

$$n = \frac{\log(1-p)}{\log(1-w^d)}. \tag{7}$$

Eq. (7) indicates that RANSAC requires about n iterations for correct estimation at a confidence p. Therefore RANSAC requires significant computation cost in high outlier rate and the large parameter dimensions.

There are various extensions to RANSAC, for example, various score function [17, 22] and making use of similarity scores [7]. Raguram *et al.* presented USAC [16], which is one of the sophisticated methods, such as the SPRT test [8] and local optimization [9].

2.2. Fundamental matrix estimation

Although our method can be applied to any robust UCLF problem, in this paper we especially focus on robust fundamental matrix estimation, which is an important task in computer vision applications, such as stereo problems and 3D reconstruction. Fundamental matrix estimation generally has three steps: (i) obtaining corresponding points (including outliers) by extracting and matching feature points; (ii) removing outliers by robust estimation techniques; (iii) estimating the fundamental matrix from correct corresponding points. Although the third step, such as imposing epipolar constraint and rank-2 constraint to the fundamental matrix estimation [20, 25, 4, 26], is an important research area, our research focuses on the second step: outlier rejection.

The fundamental matrix is estimated from corresponding points, which are extracted by local feature matching. Given correct corresponding points $\mathbf{p} = [p_x, p_y, 1]^\top$ and $\mathbf{p'} = [p'_x, p'_y, 1]^\top$ between two images, the fundamental matrix $\mathbf{F}$ satisfies the epipolar constraint:

$$\mathbf{p}^\top \mathbf{F} \mathbf{p'} = \mathbf{a}^\top \mathbf{x} = 0, \tag{8}$$

$$\mathbf{a} = [p_x p'_x, p_y p'_x, p'_x, p_x p'_y, p_y p'_y, p'_y, p_x, p_y, 1]^\top,$$

$$\mathbf{x} = [F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33}]^\top.$$

From epipolar constraint, we can estimate the fundamental matrix by solving the following LS problem:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|_2^2 \ \ s.t. \ \ \|\mathbf{x}\|_2^2 = 1. \tag{9}$$

where $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]^\top \in \mathbb{R}^{n \times 9}$ is a matrix of n correct correspondences. This is well known as the eight-point algorithm [14, 11]. Eq. (9) can be solved by the smallest eigenvector of $\mathbf{A}$, which is obtained by singular value decomposition (SVD). Note that Eq. (9) is a nonconvex optimization problem because the solution set is restricted to a sphere, which is a nonconvex set.

Robust fundamental matrix estimation is formulated as Eq. (1), which uses a robust loss function instead of the quadratic function in Eq. (9). In the IRLS algorithm for M-estimator, the LS solution is used as an initial guess. However, IRLS often fails to estimate the parameters accurately because of the poor initial solution and the nonconvexity of the optimization problem. It is reported that the IRLS from the initial LS solution converges to a poor local minimum for 10% to 15% of outliers [19, 21].

The poor initial solution problem can be avoided by probabilistic approaches such as RANSAC [10], which does not require an initial solution. These methods are practical in terms of their speed and robustness, but they are significantly problematic because of their hugely increased computational cost with a high frequency of outliers, as described in Section 2.1.2.

Recently, Cheng *et al.* [5] introduced a new robust optimization method that considers the rank-2 constraint. Although their method obtained robust estimates, it requires a considerable amount of processing time because of the requirement to solve semidefinite programming problems[2].

3. Preliminary: Robust UCLF Problem

Before describing our method, we introduce the smallest eigenvalue minimization problem, which is equivalent to the robust UCLF problem of Eq. (1). We then present the cause of the poor initial LS solution problem as in Fig. 1.

3.1. The smallest eigenvalue minimization problem

In this section, we show that the robust UCLF problem can be transformed into a smallest eigenvalue minimization problem. Using the residual $r_i = \mathbf{a}_i^\top \mathbf{x}$ and Eq. (4), the robust UCLF problem of Eq. (1) can be transformed into

$$\min_{\mathbf{x},\mathbf{w}} \sum_{i=1}^{n} \left(w_i (\mathbf{a}_i^\top \mathbf{x})^2 + f(w_i) \right) \quad \text{s.t.} \quad \|\mathbf{x}\|_2^2 = 1$$

$$= \min_{\mathbf{x},\mathbf{w}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{x} + \sum_{i=1}^{n} f(w_i) \quad \text{s.t.} \quad \|\mathbf{x}\|_2^2 = 1, \tag{10}$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{w})$ is a diagonal matrix.

Just like solving Eq. (9), the optimization problem Eq. (10) can be solved for $\mathbf{x}$ by $\mathbf{x} = \mathbf{u}_1$, where $\mathbf{u}_i$ is the ith eigenvector of $\mathbf{A}^\top \mathbf{W} \mathbf{A}$ sorted in ascending order. From $\mathbf{u}_1^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{u}_1 = \lambda_1$, where λ_i is the ith eigenvalue of the matrix $\mathbf{A}^\top \mathbf{W} \mathbf{A}$, Eq. (10) we can be transformed into the following equivalent optimization problem:

$$\min_{\mathbf{w}} \lambda_1(\mathbf{w}) + \sum_{i=1}^{n} f(w_i), \tag{11}$$

Therefore, a robust UCLF problem Eq. (10) is equivalent to a smallest eigenvalue minimization problem on $\mathbf{w}$.

3.2. Poor initial solution in robust UCLF

In this section, we explain why an LS solution tends to be a poor initial solution in robust UCLF problems, as shown in Fig 1(d). Let $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n]^\top$ have no outlier and inlier noise, *i.e.* $\lambda_1 = 0$, and λ_j and $\mathbf{u}_j$ be the jth eigenvalue and eigenvector of $\mathbf{A}^\top \mathbf{A}$, respectively. Note that $\mathbf{a}_i^\top \mathbf{u}_1 = 0$ for all i, and $\mathbf{u}_1$ is the LS solution of $\mathbf{A}$.

We consider an LS solution for the matrix $\hat{\mathbf{A}} = [\mathbf{A}^\top, \hat{\mathbf{a}}]^\top$, where $\hat{\mathbf{a}}$ is an outlier *i.e.* $\hat{\mathbf{a}}^\top \mathbf{u}_1 > 0$. A solution $\mathbf{x}$ can be written as a linear combination of $\mathbf{u}_j$: $\mathbf{x} = \sum_j c_j \mathbf{u}_j$, where $\|\mathbf{c}\|_2^2 = 1$. From $\lambda_j = \|\mathbf{A}\mathbf{u}_j\|_2^2$, we obtain

$$\|\hat{\mathbf{A}}\mathbf{x}\|_2^2 = \sum_j c_j^2 \lambda_j + \left(\sum_j c_j (\hat{\mathbf{a}}^\top \mathbf{u}_j) \right)^2. \tag{12}$$

We can minimize Eq. (12) on $\mathbf{c}$ with the constraint of $\|\mathbf{c}\|_2^2 = 1$:

$$c_j = \frac{r_j \sum_l r_l}{2(\lambda_j + z)}, \tag{13}$$

[2] They reported about 10 seconds for each iteration with 100 points

where $r_j = \hat{\mathbf{a}}^\top \mathbf{u}_j$ and z is the normalization term to satisfy $\|\mathbf{c}\|_2^2 = 1$. If $\lambda_j \gg \lambda_1$ for all $j = 2, 3, ..., d$, $c_1 \sim 1$ *i.e.* $\mathbf{u}_1$ is the near LS solution for the matrix $\hat{\mathbf{A}}$. However, if there exists j such that $r_j/(\lambda_j + z) > r_1/(\lambda_1 + z)$, $\mathbf{u}_j$ is the near LS solution for the matrix $\hat{\mathbf{A}}$, as shown in Fig 1(d).

4. Proposed method

In this section, we propose an eigenvalues minimization problem, which is an approximation of Eq. (11), and an algorithm to minimize our objective function.

4.1. Preliminary observation

As described in Section 3.2, an LS solution, the smallest eigenvector, may be not suitable for an initial solution in robust UCLF problems. We find that the other eivenvectors seem to be helpful for robust estimation in such cases. Figs. 2 (b)–(j) show the residuals computed by each eigenvector in fundamental matrix estimation. Although we cannot judge the second correspondence as an outlier from the smallest eigenvector, we can judge from the third smallest eigenvector. From this observation, we develop a method using multiple eigenvectors to avoid a poor initial solution.

4.2. Problem formulation

We propose an approximation of Eq. (11) as follows:

$$\min_{\mathbf{w}} \frac{1}{\sum_{j=1}^{k} \frac{1}{\lambda_j}} + \sum_{i=1}^{n} f(w_i). \tag{14}$$

For simplicity of notation, $\lambda_i(\mathbf{w})$ is written as λ_i.

We will explain the rationale of this approximation as follows. Assume that inliers contain no noise and use a hard threshold loss function (*e.g.* Talwar loss). If the ith observation is an inlier, $r_i = \mathbf{a}_i^\top \mathbf{x} = 0$. If the ith observation is an outlier, $w_i = 0$ when $\mathbf{w}$ is near the solution. Thus, $\sum_i w_i r_i^2 = \lambda_1 = 0$ holds. In addition, if Eq. (11) has a unique solution for $\mathbf{x}$, $\lambda_j \neq 0$ holds for $j \geq 2$. Therefore, $\lambda_1 \sim 0$ and $\lambda_1/\lambda_j \sim 0$ $(j \geq 2)$ are satisfied and the first term of Eq. (14) can be approximated as

$$\frac{1}{\sum_{j=1}^{k} \frac{1}{\lambda_j}} = \lambda_1 - \frac{\lambda_1^2 \sum_{j=2}^{k} \frac{1}{\lambda_j}}{1 + \lambda_1 \sum_{j=2}^{k} \frac{1}{\lambda_j}} \sim \lambda_1. \tag{15}$$

Therefore, our approximation Eq. (14) is close to the original objective function Eq. (11) when $\mathbf{w}$ is near the optimal solution.

This approximation has good properties: (i) Eq. (14) is a k eigenvalues minimization problem and it is expected to exploit not only the smallest eigenvector but also other eigenvectors, such as the second smallest eigenvector in Fig. 1(d); and (ii) it has a fast convergence algorithm.

4.3. Optimization

We describe our optimization method for Eq. (14). Our optimization is based on majorize-minimization, which is guaranteed to decrease (or unchange) the original objective function. First, use Jensen's inequality to construct the majorizer function as

$$\frac{1}{\sum_{j=1}^{k} \frac{1}{\lambda_j}} = \frac{1}{\sum_{j=1}^{k} z_j \frac{1}{z_j \lambda_j}} \leq \sum_{j=1}^{k} z_j \frac{1}{\frac{1}{z_j \lambda_i}} = \sum_{j=1}^{k} z_j^2 \lambda_j, \tag{16}$$

where $z_1, \ldots, z_k$ are positive numbers that sum to 1. The equality holds if and only if

$$z_j = \frac{1}{\lambda_j \left(\sum_{j=1}^{k} \frac{1}{\lambda_j} \right)}. \tag{17}$$

The right-hand side of Eq. (16) is the majorizer function of the first term of Eq. (14). Therefore, we can use majorize-minimization [13] and obtain the update rules of $\mathbf{w}$ and $\boldsymbol{\alpha}$:

$$\mathbf{w}^{(t)} = \min_{\mathbf{w}} \sum_{j=1}^{k} \alpha_j^{(t)} \lambda_j + \sum_{i=1}^{n} f(w_i) \tag{18}$$

$$\alpha_j^{(t)} = \frac{1}{\left(\lambda_j^{(t+1)}\right)^2 \left(\sum_{j=1}^{k} \frac{1}{\lambda_j^{(t+1)}} \right)^2}. \tag{19}$$

Eq. (18) cannot be minimized directly; thus, we use majorize-minimization again. The Courant-Fisher min-max theorem provides majorizer functions of eigenvalues as

$$\lambda_i = \min_{S} \max_{x \in S} \mathbf{x}^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{x} \;\; s.t. \;\; \|\mathbf{x}\|_2^2 = 1, \dim(S) = i, \tag{20}$$

where S_j is a linear subspace of $\mathbf{x}$. From Eq. (20), Eq. (18) can be rewritten into the following optimization problem:

$$\min_{\substack{\mathbf{w}, S_j \\ \forall j=1,\ldots,k}} \max_{\substack{\mathbf{x}_j \in S_j \\ \forall j=1,\ldots,k}} \sum_{j=1}^{k} \alpha_j^{(t)} \left(\mathbf{x}_j^\top \mathbf{A}^\top \mathbf{W} \mathbf{A} \mathbf{x}_j \right) + \sum_{i=1}^{n} f(w_i)$$
$$\text{s.t. } \dim(S_j) = j, \|\mathbf{x}_j\|_2^2 = 1 \;\; \forall j = 1, \ldots, k. \tag{21}$$

Then, we employ an alternating optimization approach to solve Eq. (21).

Update S_j and $\mathbf{x}_j$ for all $j = 1, \ldots, k$: Our optimization problem is separable for each $\{S_j, \mathbf{x}_j\}$. Therefore, we can minimize these variables independently. The update rule of $\{S_j, \mathbf{x}_j\}$ can be obtained by the Courant–Fisher min–max theorem:

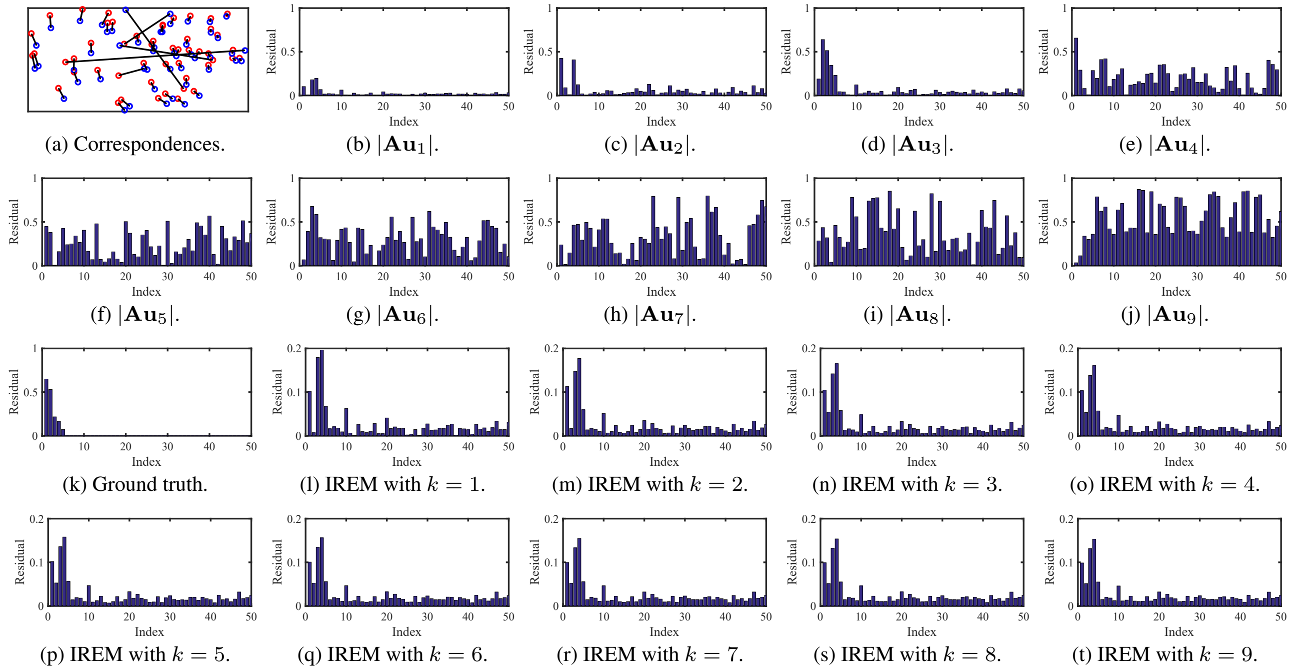
$$S_j^{(t+1)} = \text{span}(\mathbf{u}_1, \ldots, \mathbf{u}_j), \tag{22}$$

$$\mathbf{x}_j^{(t+1)} = \mathbf{u}_j, \tag{23}$$

where $\mathbf{u}_j$ is jth eigenvector of the matrix $\mathbf{A}^\top \mathbf{W}^{(t)} \mathbf{A}$.

Update $\mathbf{w}$: Given S_j and $\mathbf{x}_j$ for all $j = 1, \ldots, k$, our optimization problem is simplified as:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} \left(w_i \left(r_i^{(t+1)} \right)^2 + f(w_i) \right), \tag{24}$$



(a) Correspondences. (b) $|\mathbf{Au}_1|$. (c) $|\mathbf{Au}_2|$. (d) $|\mathbf{Au}_3|$. (e) $|\mathbf{Au}_4|$.

(f) $|\mathbf{Au}_5|$. (g) $|\mathbf{Au}_6|$. (h) $|\mathbf{Au}_7|$. (i) $|\mathbf{Au}_8|$. (j) $|\mathbf{Au}_9|$.

(k) Ground truth. (l) IREM with $k = 1$. (m) IREM with $k = 2$. (n) IREM with $k = 3$. (o) IREM with $k = 4$.

(p) IREM with $k = 5$. (q) IREM with $k = 6$. (r) IREM with $k = 7$. (s) IREM with $k = 8$. (t) IREM with $k = 9$.

Fig 2: An experiment of robust fundamental matrix estimation. (a) Corresponding points (45 inliers and 5 outliers). (b)–(j) The value of $|\mathbf{Au}_j|$ for $j = 1, \dots, 9$. (k) Residuals obtained by the correct parameter. (l)–(t) The first-iteration residuals of IREM calculated by Eq. (25), which is the weighted sum of $|\mathbf{Au}_j|$ for $j = 1, \dots, k$. IREM with $k = 1$ (equivalent to IRLS) regards the second correspondence as an inlier; however, IREM with $k \geq 3$ correctly regards it as an outlier.

Alg. 1 Iteratively reweighted eigenvalues minimization

Input: Observation matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$
Initialize: All 1 for $\mathbf{w}^{(0)}$, $t = 0$
while not converged **do**
  $[\mathbf{U}, \mathbf{L}, \mathbf{V}] = \mathrm{svd}(\mathbf{A}^\top \mathbf{W}^{(t)} \mathbf{A})$ and update $\mathbf{x}_j^{(t+1)} = \mathbf{u}_j$, $\lambda_j^{(t+1)} = L_{jj}\ \forall j = 1, \dots, k$
  Update $\alpha_j^{(t+1)}$ by Eq. (19).
  Update $\mathbf{w}^{(t+1)}$ by $w_i^{(t+1)} = \psi(r_i^{(t+1)})$.
  $t = t + 1$.
end while
Output: $\mathbf{x}_1$

where $r_i^{(t)}$ is given by

$$r_i^{(t)} = \sqrt{\sum_{j=1}^{k} \alpha_j \left(r_{ij}^{(t)}\right)^2}, \tag{25}$$

$$r_{ij}^{(t)} = \mathbf{a}_i^\top \mathbf{x}_j^{(t)}. \tag{26}$$

Eq. (24) is the same formulation as Eq. (4); therefore, we can update $\mathbf{w}$ by the weight function $\psi(r)$:

$$w_i^{(t+1)} = \psi\left(r_i^{(t+1)}\right). \tag{27}$$

The proposed algorithm, named iteratively reweighted eigenvalues minimization (IREM), is summarized in Algorithm 1. IREM has a complexity of $O(m^3)$ time for SVD, $O(m^2 n)$ for the computation of $\mathbf{A}^\top \mathbf{W} \mathbf{A}$, and $O(kmn)$ for the computation of $\mathbf{r}^{(t+1)}$ during each iteration.

In Fig. 2(l)–2(t) we show the residual $r_i^{(1)}$ obtained during the first iterative cycle from IREM for each k. Based on Eq. (25) and Eq. (26), the IREM residual $r_i^{(1)}$ is calculated by the weighted sum of the $\mathbf{Au}_j$ for $j = 1, \dots, k$ which are shown in Fig. 2(b)–2(j). IREM provides more optimal residuals when $k \geq 3$ because of the third eigenvalue residual, which indicates that the residual is large for the second observations, as in Fig. 2(d). The results of IREM with $k \geq 3$ are almost identical. This is because the weight of $\mathbf{Au}_j$ is smaller for a larger eigenvalue, as in Eq. (25) and Eq. (19).

4.4. Loss function and other details

Although any robust loss function can be employed for our method, the Talwar loss function is selected because of its robustness. We explain some techniques for optimizing the Talwar loss function.

4.4.1 Avoiding poor local minima from the Talwar loss

IREM handles the local minima problem caused by UCLF problem, however, the Talwar loss function also provides the local minima problem owing to its high nonconvexity. To address this problem, we use the graduated nonconvexity [3] approach, which first solves the convex optimization and then gradually transforms the convex problem into the original nonconvex problem. In this study, we initially use a large-cost parameter $c^{(1)}$ to find a stable solution, before using smaller values for $c^{(t)}$ in each iteration.

4.4.2 Fast computation of $\mathbf{A}^\top \mathbf{W}\mathbf{A}$

Given $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]^\top \in \mathbb{R}^{n\times m}$, the computation of $\mathbf{B}^{(t)} = \mathbf{A}^\top \mathbf{W}^{(t)}\mathbf{A}$ generally has a complexity of $O(m^2 n)$. This computational complexity can be reduced by the binary nature of $\mathbf{W}$ as

$$\mathbf{B}^{(t+1)} = \mathbf{B}^{(t)} + \sum_{i=1}^{n} \left(w_i^{(t+1)} - w_i^{(t)}\right) \mathbf{a}_i \mathbf{a}_i^\top. \quad (28)$$

The summation of $\mathbf{a}_i \mathbf{a}_i^\top$ need not be calculated if $w_i^{(t+1)} = w_i^{(t)}$. Using this technique, the computational complexity of IREM of $O(m^2 n + m^3)$ is reduced to $O(lm^2 + kmn + m^3)$ during each iteration, where l is the number of changed elements between $\mathbf{w}^{(t+1)}$ and $\mathbf{w}^{(t)}$.

4.4.3 Parameter c

The Talwar loss function has a parameter c which determines the boundary between inliers and outliers. In this paper we use c as a pre-defined parameter. For automatic selection, we can use existing methods such as median absolute deviation (MAD) [18].

5. Experiments

We evaluated the performance of our approach in several robust UCLF problems. All experiments were run on an Intel Core i7-6500U CPU (2.50 GHz) with 8 GB RAM, and were implemented in MATLAB except for USAC [16], for which the publicly available code implemented in C++ was used. For both the IRLS and IREM algorithms, we utilized graduated nonconvexity by $c^{(t+1)} = \max(\min(0.5c^{(t)}, \mu^{(t)}), c_{min})$, where $\mu^{(t)}$ is the mean-squared error of inliers at the tth iteration.

5.1. Robust conic fitting

In this section we consider robust conic fitting problem. A conic satisfies the following equation:

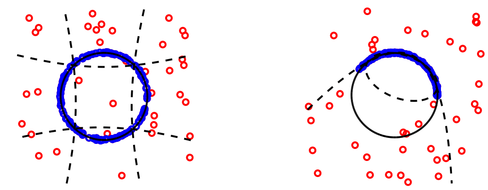
$$ax^2 + bxy + cy^2 + dx + ey + f = 0. \quad (29)$$

Given n two-dimensional points $\{x_i, y_i\}$ for $i = 1, \ldots, n$, a robust conic fitting problem can be formulated as Eq. (1) by using

$$\mathbf{a} = [x_i^2, x_i y_i, y_i^2, x_i, y_i, 1]^\top, \quad \mathbf{x} = [a, b, c, d, e, f]^\top. \quad (30)$$

Therefore we can use IREM to solve robust conic fitting.

We generated inlier points $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_n]^\top \in \mathbb{R}^{(n\times 2)}$, where $\mathbf{b}_i = [\cos(r), \sin(r)]^\top + [n_x, n_y]^\top$, r is a random value from $[0, 2\pi r_{max}]$ and n_x, n_y are Gaussian



(a) $r_{max} = 1.0$. (b) $r_{max} = 0.4$

Fig 3: Examples of conic. The solid line represents the LS solution on $\mathbf{B}$. Blue and red points are inliers and outliers, respectively, The broken lines show the parameter of $\mathbf{u}_1 + \mathbf{u}_2$ and $\mathbf{u}_1 - \mathbf{u}_2$, where $\mathbf{u}_i$ is the ith eigenvector of $\mathbf{B}$.

Table 2: Robust conic fitting with different r_{max}. Averaged objective values are reported.

r_{max}	λ_2/λ_1	IREM	IRLS
1.0	43.6	**0.0152**	**0.0152**
0.9	41.0	**0.0151**	0.0155
0.8	38.1	**0.0150**	0.0164
0.7	26.8	**0.0151**	0.0174
0.6	13.8	**0.0151**	0.0183
0.5	6.24	**0.0149**	0.0179
0.4	2.62	**0.0149**	0.0169
0.3	1.26	**0.0093**	0.0132

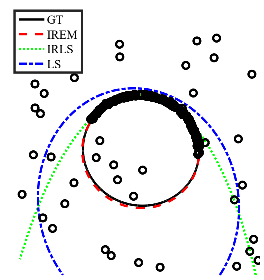


Fig 4: Robust conic fitting on $r_{max} = 0.4$.

noise from $\mathcal{N}(0, 0.01^2)$. We also generated outlier points $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_m]^\top \in \mathbb{R}^{(m\times 2)}$, where $\mathbf{c}_i$ was sampled from uniform distribution in $[-2, 2]^2$. Fig. 3 show the examples of conic with different r_{max}.

First, we consider the influence of eigenvalues of inlier points matrix $\mathbf{B}$. Let λ_j be the eigenvalues of $\mathbf{B}$. Table 2 shows the ratio of λ_2/λ_1 for different r_{max}. λ_2 is smaller when r_{max} is smaller. Small λ_2 causes near-nonuniqueness of LS solution, as described in Section 3.2 and shown in Fig. 3(b), and therefore LS solution is easily influenced by outliers, as shown in Fig. 3.

Table 2 shows the results of robust conic fitting on $\mathbf{A} = [\mathbf{B}^\top, \mathbf{C}^\top]^\top$. We report averaged objective values of Eq. (1) over 100 trials. IRLS achieves comparable performance in $r_{max} = 1.0$. However, IRLS degrades the performance in small r_{max} because of a bad initial solution, as shown in Fig. 4. IREM outperforms IRLS, especially in small r_{max}.

5.2. Robust fundamental matrix estimation

We compared our proposed method with IRLS, RANSAC [10], and USAC [16]. USAC is a combination of many techniques such as PROSAC [7], SPRT test [8], and local optimization (LO). We ensured a fair comparison by not using the similarity scores of correspondences in USAC.

In our method, we used normalized data obtained by Hartley's normalization [11] and the parameter $k = 9$ and $c_{min} = 5 \times 10^{-5}$ for the experiments with synthetic and

Table 3: Results of comparison for different outlier rates (1,000 correspondences, $t_s = 1$). The average mean Sampson errors and recovery rates were reported. The best and second best results are highlighted in bold and italics, respectively.

Outlier Rate	IREM (Ours)	IRLS	RANSAC 100	RANSAC 1,000	RANSAC 10,000	USAC w/ LO	USAC w/o LO
0.1	**0.688 / 98.3**	4.06 / 84.4	1.08 / 93.6	0.829 / 96.8	*0.724 / 97.8*	1.31 / 94.4	1.20 / 94.1
0.2	**0.693 / 98.2**	11.2 / 73.7	2.30 / 87.8	1.02 / 95.2	*0.853 / 97.2*	1.79 / 93.6	1.97 / 92.7
0.3	**0.753 / 97.9**	25.2 / 66.2	5.22 / 82.8	1.71 / 92.7	*1.13 / 96.0*	2.10 / 94.5	3.29 / 92.0
0.4	**0.848 / 97.5**	62.8 / 61.5	20.5 / 71.1	4.37 / 88.2	*1.74 / 94.9*	3.00 / 94.4	4.91 / 91.0
0.5	**0.909 / 97.1**	80.3 / 57.9	34.0 / 60.0	11.2 / 80.1	3.63 / 91.8	*3.34 / 93.8*	5.31 / 91.9
0.6	**1.06 / 96.6**	694 / 36.3	148 / 40.4	37.6 / 68.6	9.87 / 86.0	*5.30 / 93.0*	6.42 / 91.9
0.7	**1.80 / 95.0**	2357 / 12.8	583 / 22.6	105 / 52.3	40.2 / 71.9	*5.86 / 92.8*	7.76 / 91.4

Table 4: Results of robust fundamental matrix estimation under different t_s (1,000 correspondences, 0.1 outlier rate). Averaged objective values and recovery rates are reported.

t_s	λ_2/λ_1	IREM	IRLS
1	19.6	**0.0271 / 98.20**	0.0302 / 84.82
2	66.5	**0.0274 / 98.24**	0.0293 / 90.26
3	140	**0.0280 / 98.37**	0.0294 / 92.26
4	234	**0.0286 / 98.28**	0.0290 / 96.26
5	339	**0.0293 / 98.18**	0.0294 / 97.80

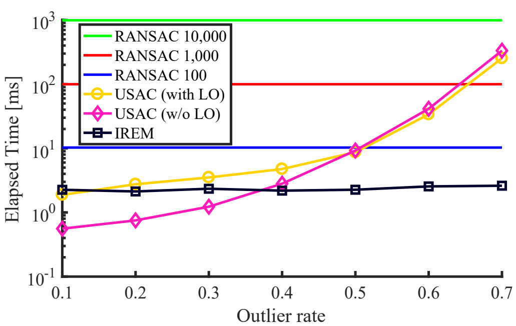


Fig 5: Elapsed time in milliseconds (1000 correspondences). Note that USAC was implemented in C++; others were implemented in MATLAB.

real data. For RANSAC and USAC, we used raw data and the Sampson error measure:

$$d_s(\mathbf{x},\mathbf{x}',\mathbf{F})=\frac{\left(\mathbf{x}'^{\top}\hat{\mathbf{F}}\mathbf{x}\right)^2}{\left(\hat{\mathbf{F}}\mathbf{x}\right)_1^2+\left(\hat{\mathbf{F}}\mathbf{x}\right)_2^2+\left(\hat{\mathbf{F}}\mathbf{x}'\right)_1^2+\left(\hat{\mathbf{F}}\mathbf{x}'\right)_2^2}, \quad (31)$$

where $(\hat{\mathbf{F}}\mathbf{x})_j$ corresponds to the jth entry of $\hat{\mathbf{F}}\mathbf{x}$. For RANSAC and USAC, the normalized eight-point algorithm was applied to refine the final results after removing the outliers. For USAC, we used the SPRT test and LO (optional), and set the confidence parameter $p = 0.9$.

5.2.1 Experiments using synthetic data

We randomly generated 3D points within a box $[-2, 2]$, $[-2, 2]$, $[1, 2]$ and obtained the corresponding inlier points by projecting the 3D points onto a 640×480 2D image

Table 5: Results of comparison for different total numbers of correspondences ($t_s = 1$, outlier rate = 0.5). The average mean Sampson error was reported.

# of Points	IREM	RANSAC 100	RANSAC 1,000	RANSAC 10,000	USAC w/ LO
100	8.78	38.5	13.0	**4.27**	*6.91*
300	**1.50**	48.2	11.8	3.69	*3.53*
1000	**0.909**	34.0	11.2	3.63	*3.34*

plane. In each experiment, we used a fixed-camera projection matrix defined by $\mathbf{P}_1 = \mathbf{K}[\mathbf{I}_{3\times3}, [0,0,0]^\top]$ and $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}, \mathbf{t}]$, where $\mathbf{t} = t_s[-3,-2,1]^\top$ and $\mathbf{R}$ were determined from the rotation angle $\pi/36$ and rotation axis $[1, 2, 3]^\top$, and the camera intrinsic matrix was $\mathbf{K}$ such that $k_{11} = k_{22} = 700$, $k_{13} = 320$, $k_{23} = 240$, and $k_{33} = 1$. We added Gaussian noise from $\mathcal{N}(0, 1^2)$ to each of the 2D correspondences and generated the outlier correspondences by replacing points randomly in the range of 640×480. For each condition, we performed the test 100 times and reported the average value for each method.

The mean Sampson error of the inlier correspondences was used as the evaluation criterion: $\frac{1}{n}\sum_{i=1}^{n} d_s(\mathbf{x}_i, \mathbf{x}'_i, \hat{\mathbf{F}})$, where n is the total number of inliers. Inliers were defined as points that satisfied $d_s(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{F}_{gt}) < \tau$ (using $\tau = 3$), where $\mathbf{F}_{gt}$ was computed by $\mathbf{P}_1$ and $\mathbf{P}_2$. We also used the recovery rate: the percentage of the correctly identified inliers.

We first compared IREM with IRLS under different translation vector scales t_s. Table 4 shows the values of λ_2/λ_1, where λ_i is the ith eigenvalue of the observation matrix from only inlier correspondences as with Section 5.1, the average objective values of Eq. (1), and recovery rates. We used 1,000 correspondences with 10% of outliers. As in the case with robust conic fitting, IRLS does not perform well in the cases of small λ_2 despite 10% of outliers, while IREM remains high performance.

We also compared IREM with RANSAC and USAC. The error and computational time are shown in Table 3 and Fig. 5, respectively. IRLS does not work well in 10% of outliers; however, IREM achieved significant robustness. IREM requires on average 14.6 iterations for convergence,

Table 6: Experimental results for the Oxford multiview dataset. For RANSAC and USAC, 10 trials were performed, and the average and the best results were reported. The results achieving above 99% and 95% recovery of inliers are highlighted in bold and italics, respectively. Note that USAC was implemented in C++; others were implemented in MATLAB.

Pairs	# of Points	# of Inliers	mean Sampson error / recovery rate						Elapsed Time [ms]			
			IREM (Ours)	IRLS	RANSAC average	RANSAC best	USAC average	USAC best	IREM (Ours)	IRLS	RANSAC	USAC
#1	1313	1040	**0.157 / 100**	**0.162 / 100**	**0.280 / 99.2**	**0.158 / 100**	*0.649 / 95.6*	**0.178 / 99.9**	3.2	2.0	127	4.3
#2	727	525	**0.163 / 100**	**0.164 / 100**	*0.654 / 95.8*	**0.170 / 100**	**0.174 / 100**	**0.171 / 100**	1.6	1.6	100	3.8
#3	635	589	**0.152 / 99.8**	8.93 / 68.6	**0.153 / 100**	**0.152 / 100**	**0.154 / 100**	**0.154 / 100**	2.3	3.0	84.1	1.9
#4	233	198	**0.156 / 100**	16.9 / 71.7	**0.159 / 99.7**	**0.133 / 100**	**0.200 / 99.1**	**0.135 / 99.5**	1.4	1.2	65.0	2.5
#5	343	169	*0.322 / 97.6*	81.2 / 45.0	7.58 / 75.4	0.715 / 94.1	1.32 / 88.8	1.19 / 88.8	2.6	1.0	78.4	6.4
#6	136	79	1.23 / 92.4	31.3 / 50.6	*0.579 / 97.9*	*0.386 / 98.7*	*0.500 / 98.7*	*0.424 / 98.7*	1.2	0.9	60.0	1.8

and the computation times for one iteration of IREM and RANSAC is almost the same. Compared with RANSAC, IREM was about 500 times faster with smaller error than the 10,000-iteration RANSAC method. Compared with USAC, IREM is faster for a high outlier rate even though our method was implemented in MATLAB, which is generally slower than C++. USAC generated 60.6 hypotheses for the case of 30% outliers and 3.66×10^4 hypotheses for the case of 70% outliers.

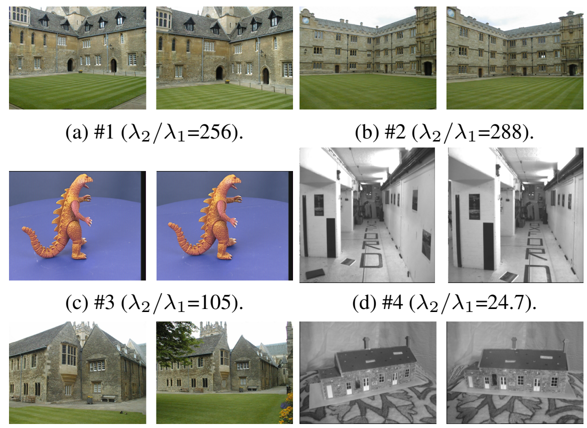
We also compared under different total numbers of corresponding points, as in Table 5. The performance of IREM is worse when using less total number of correspondences, in contrast to RANSAC. An estimator generally performs better on larger amount of data, and IREM works well on a large number of points because it uses all data to estimate a parameter. However, RANSAC does not benefit from large data because it uses only a minimal subset to generate a model parameter hypothesis.

5.2.2 Experiments using real data

In the experiments using real data, six image pairs were selected from the Oxford multiview datasets[3], which contain ground-truth camera projection matrices, as shown in Fig. 6.

We extracted FAST corners and then matched them using SURF features [1] around the corners. The inliers were obtained from the corresponding points that satisfied $d_s(\mathbf{x}_i, \mathbf{x}'_i, \hat{\mathbf{F}}) < \tau$ (using $\tau = 3$). We then estimated the fundamental matrix and evaluated the performance of each method in the same manner as in Section 5.2.1.

We compared our method to IRLS, RANSAC, and USAC. For RANSAC, the number of iterations was set to 1,000. For USAC, we used the SPRT test, LO and set the confidence parameter $p = 0.9$. The comparison results are shown in Table 6. Except for pairs #1 and #2, IRLS failed to estimate the correct parameter because of small λ_2; especially, pair #3 contains only less than 10% of outliers. RANSAC and USAC either failed to find the correct parameter or required intensive computation for a high outlier rate, such as for pair #5. IREM provided a good estimation of the real dataset, as well as running much faster. For pair



(a) #1 (λ_2/λ_1=256). (b) #2 (λ_2/λ_1=288).

(c) #3 (λ_2/λ_1=105). (d) #4 (λ_2/λ_1=24.7).

(e) #5 (λ_2/λ_1=105). (f) #6 (λ_2/λ_1=92.5).

Fig 6: Image pairs from the Oxford dataset. We show values λ_2/λ_1, where λ_i is the ith eigenvalue of the observation matrix from inlier correspondences.

#6, IREM failed to estimate the most correct parameter because of small amount of points. Overall, IREM works well on real data with high robustness and fast computation.

6. Conclusions

In this study, we revealed why IRLS does not work well in robust UCLF problem and proposed a novel algorithm, which we named IREM. We demonstrated the very fast convergence and high robustness of IREM based on experimental results. In terms of robust fundamental matrix estimation, IREM operated about 5–500 times faster than RANSAC while preserving robustness and achieved comparable or superior performance to the state-of-the-art extension of RANSAC.

In our future research, we plan to extend the IREM algorithm to other robust subspace estimation problem, such as multiple structures [6, 15]. In robust fundamental matrix estimation, it would be challenging to take into account the rank-2 constraint for our method.

Acknowledgements. This research is partially supported by CREST (JPMJCR1686).

[3] Available at http://www.robots.ox.ac.uk/~vgg/data/data-mview.html

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 8

[2] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996. 2

[3] A. Blake and A. Zisserman. *Visual reconstruction*. MIT Press, 1987. 6

[4] F. Bugarin, A. Bartoli, D. Henrion, J.-B. Lasserre, J.-J. Orteu, and T. Sentenac. Rank-constrained fundamental matrix estimation by polynomial global optimization versus the eight-point algorithm. *JMIV*, 53(1):42–60, 2015. 3

[5] Y. Cheng, J. A. Lopez, O. Camps, and M. Sznaier. A convex optimization approach to robust fundamental matrix estimation. In *CVPR*, pages 2170–2178, 2015. 3

[6] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *ICCV*, pages 413–420. IEEE, 2009. 8

[7] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *CVPR*, pages 220–226, 2005. 2, 6

[8] O. Chum and J. Matas. Optimal randomized RANSAC. *TPAMI*, 30(8):1472–1482, 2008. 2, 6

[9] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, pages 236–243, 2003. 2

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 3, 6

[11] R. I. Hartley. In defense of the eight-point algorithm. *IEEE TPAMI*, 1997. 1, 3, 6

[12] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. 1, 2

[13] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004. 2, 4

[14] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. 3

[15] S. Mittal, S. Anand, and P. Meer. Generalized projection-based m-estimator. *TPAMI*, 34(12):2351–2364, 2012. 8

[16] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm. USAC: a universal framework for random sample consensus. *IEEE TPAMI*, 35(8):2022–2038, 2013. 2, 6

[17] P. J. Rousseeuw. Least median of squares regression. *JASA*, 79(388):871–880, 1984. 2

[18] P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283, 1993. 6

[19] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM review*, 41(3):513–537, 1999. 3

[20] Y. Sugaya and K. Kanatani. High accuracy computation of rank-constrained fundamental matrix. In *BMVC*, pages 1–10, 2007. 3

[21] P. H. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3):271–300, 1997. 3

[22] P. H. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 78(1):138–156, 2000. 2

[23] C. Zach. Robust bundle adjustment revisited. In *ECCV*, pages 772–787, 2014. 2

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

[25] Y. Zheng, S. Sugimoto, and M. Okutomi. A branch and contract algorithm for globally optimal fundamental matrix estimation. In *CVPR*, pages 2953–2960, 2011. 3

[26] Y. Zheng, S. Sugimoto, and M. Okutomi. A practical rank-constrained eight-point algorithm for fundamental matrix estimation. In *CVPR*, pages 1546–1553, 2013. 3

[27] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *ECCV*, pages 766–782, 2016. 2