Fast Multidimensional Ellipsoid-Specific Fitting by Alternating Direction Method of Multipliers

Zhouchen Lin, *Senior Member, IEEE*, and Yameng Huang

Abstract—Many problems in computer vision can be formulated as multidimensional ellipsoid-specific fitting, which is to minimize the residual error such that the underlying quadratic surface is a multidimensional ellipsoid. In this paper, we present a fast and robust algorithm for solving ellipsoid-specific fitting directly. Our method is based on the alternating direction method of multipliers, which does not introduce extra positive semi-definiteness constraints. The computation complexity is thus significantly lower than those of semi-definite programming (SDP) based methods. More specifically, to fit n data points into a p dimensional ellipsoid, our complexity is $O(p^6 + np^4) + O(p^3)$, where the former O results from preprocessing data **once**, while that of the state-of-the-art SDP method is $O(p^6 + np^4 + n^3p^2)$ *for each iteration*. The storage complexity of our algorithm is about $\frac{1}{2}np^2$, which is at most $1/4$ of those of SDP methods. Extensive experiments testify to the great speed and accuracy advantages of our method over the state-of-the-art approaches. The implementation of our method is also much simpler than SDP based methods.

Index Terms—Multidimensional ellipsoid, ellipsoid-specific fitting, alternating direction method of multipliers

1 Introduction

Multidimensional ellipsoid-specific fitting problems, i.e., fitting an ellipsoid surface to a given set of data points, is widely used in computer vision. For example, in [4], [21] the ellipsoid primitive is used to fit the limbs and legs of the pedestrian for gait recognition. Grammalidis and Strintzis [10] used a similar technique to segment the human head. Post et al. [18] suggested making use of ellipsoids for feature visualization. Mahdavi and Salcudean [16] and Ge et al. [9] showed that ellipsoid fitting could facilitate prostate and lung detection in medical images. Rimon and Boyd [19] and Ju et al. [11] made use of the ellipsoid primitive to reduce the computation cost of obstacle collision detection in robotic tasks. Furthermore, in high dimensional spaces, the ellipsoid fitting problem could facilitate the task of pattern classification [13], where a finite number of the ellipsoidal regions are fit to data for multidimensional pattern classification problems. There are further applications in metrology [7] as well. Ellipsoid fitting can also be understood abstractly, i.e., finding a positive semi-definite matrix such that the residual error is minimized. So the camera self-calibration also involves an ellipsoid fitting problem [26] (see Section 3.2).

There have been many approaches for solving the ellipsoid-specific fitting problem. Previous work can be classified into two categories by minimizing either the geometric distance [1], [2], [8], [12], namely the shortest distance from the given point to the ellipsoid surface, or the algebraic distance [3], [5], [6], [20], [24], i.e., the deviations of the implicit equation, which describes the geometric feature of ellipsoid, from the expected value (i.e., zero) at each given point. The pros and cons of both approaches have been well studied in the literature [1], [2], [12], [17]. While geometric fitting has a clearer physical interpretation, the corresponding optimization problem is highly non-convex. Bad initialization, non-uniform point sampling, and heavy noise may lead to a bad local minimum solution (see Fig. 1). Moreover, it is difficult to deduce the geometric error in high-dimensional spaces. In contrast, algebraic fitting is much simpler due to its convexity. So it is more preferred in the literature, thanks to its easier numerical treatment and better statistical interpretation [17]. In this paper, we focus on the minimization of the algebraic distance.

In the algebraic fitting, when there is no noise, the ellipsoid fitting problem can be formulated as[1]:

$$\text{Find} \quad \mathbf{Q} \in \mathbb{R}^{(p+1)\times(p+1)},$$

$$s.t. \quad \mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i = 0, i = 1, \ldots, n, \text{and } \mathcal{S}_p(\mathbf{Q}) \succeq \mathbf{0}, \tag{1}$$

where $\mathbf{Q}$ is the parameterizations of a p dimensional ellipsoid surface, $\mathcal{S}_p(\cdot)$ is the linear operator that extracts the leading $p \times p$ submatrix, $\mathbf{M} \succeq \mathbf{0}$ means that $\mathbf{M}$ is a positive semi-definite matrix, $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,p}, 1]^T \in \mathbb{R}^{p+1}$ is the homogenous coordinate of a p dimensional point, and n is the number of data points. Since $\mathbf{Q}$ is a symmetric matrix, it can be parameterized by $\mathbf{q} = [Q_{11}, Q_{21}, Q_{22}, Q_{31}, Q_{32}, Q_{33}, \ldots, Q_{(p+1)(p+1)}]^T$, which is an m dimensional vector with $m = (p+1)(p+2)/2$. For simplicity, we define a linear operator $\mathcal{V}(\mathbf{Q}) = \mathbf{q}$, which maps a symmetric matrix $\mathbf{Q} \in \mathbb{R}^{(p+1)\times(p+1)}$ to a vector $\mathbf{q} \in \mathbb{R}^m$. Its inverse operator is $\mathcal{M}(\mathbf{q}) = \mathbf{Q}$ that maps a vector $\mathbf{q} \in \mathbb{R}^m$ to a symmetric matrix $\mathbf{Q} \in \mathbb{R}^{(p+1)\times(p+1)}$.

When there is noise, $\mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i = 0$ cannot be fulfilled by all data points. So it is natural to minimize the residual error $\mathbf{r} = [r_1, \ldots, r_n]^T$, where $r_i = \mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i$, $i = 1, \ldots, n$, measured in some norm. Note that $r_i = \mathbf{d}_i^T\mathbf{q}$, where $\mathbf{d}_i = [x_{i,1}^2, 2x_{i,1}x_{i,2}, x_{i,2}^2, 2x_{i,3}x_{i,1}, 2x_{i,3}x_{i,2}, x_{i,3}^2, \ldots, 1]^T \in \mathbb{R}^m$ is constructed from quadratic monomials of all entries in $\mathbf{x}_i$. So the ellipsoid fitting problem can be written as

$$\min_{\mathbf{q}} \|\mathbf{D}^T\mathbf{q}\|, \quad s.t. \quad \mathcal{S}_p(\mathcal{M}(\mathbf{q})) \succeq \mathbf{0}, \tag{2}$$

where $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ and $\|\cdot\|$ is some norm.

The simplest way to solve the ellipsoid fitting problem is the linear least-square (LLS) fitting [6]. LLS solves

$$\min_{\mathbf{q}} \|\mathbf{r}\|_2^2 = \mathbf{q}^T\mathbf{K}\mathbf{q}, \quad s.t. \quad \|\mathbf{q}\|_2 = 1, \tag{3}$$

without respecting the positive-semidefiniteness of $\mathcal{S}_p(\mathcal{M}(\mathbf{q}))$, where $\mathbf{K} = \mathbf{D}\mathbf{D}^T$ and $\|\cdot\|_2$ is the ℓ_2 norm of vectors. Thus the LLS solution is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{K}$.

As LLS does not enforce positive-semidefiniteness, its solution is not guaranteed to be an ellipsoidal surface when strong noise exist. To obtain a desired result, Yu et al. [25] suggested an outlier detection method to remove noise. But to guarantee an ellipsoid-specific result, one must impose an explicit semi-definiteness constraint. Fitzgibbon et al. [6] presented an efficient method by solving LLS with a quadratic semi-definiteness constraint using generalized eigenvalue decomposition (EVD), which is for 2D ellipses. Li and Griffiths [14] extended the method to the 3D case. However, these methods cannot be generalized for higher dimensional cases because as the dimension increases, the number of the constraint equations blows up exponentially.

• *The authors are with the Key Laboratory of Machine Perception (Ministry of Education), School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China, and Cooperative Medianet Innovation Center, Shanghai Jiaotong University, Shanghai 200240, China.*
E-mail: {zlin, huangyameng}@pku.edu.cn.

Manuscript received 1 Dec. 2014; revised 29 May 2015; accepted 3 Aug. 2015. Date of publication 17 Aug. 2015; date of current version 8 Apr. 2016.
Recommended for acceptance by R. Vidal.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2469283

1. Due to the semi-definiteness constraint $\mathcal{S}_p(\mathbf{Q}) \succeq \mathbf{0}$, the solution may degenerate to a paraboloid when the solution lies on the boundary of its feasible set. When this solution is not acceptable in real world applications, we may modify the constraint as $\mathcal{S}_p(\mathbf{Q}) \succeq \epsilon\mathbf{I}$ with a small ϵ. Our algorithm can still work with slight modification. Note that we cannot change the constraint to $\mathcal{S}_p(\mathbf{Q}) \succ \mathbf{0}$. Otherwise, the problem may have no solution when the optimal value is attained on the boundary of feasible set.

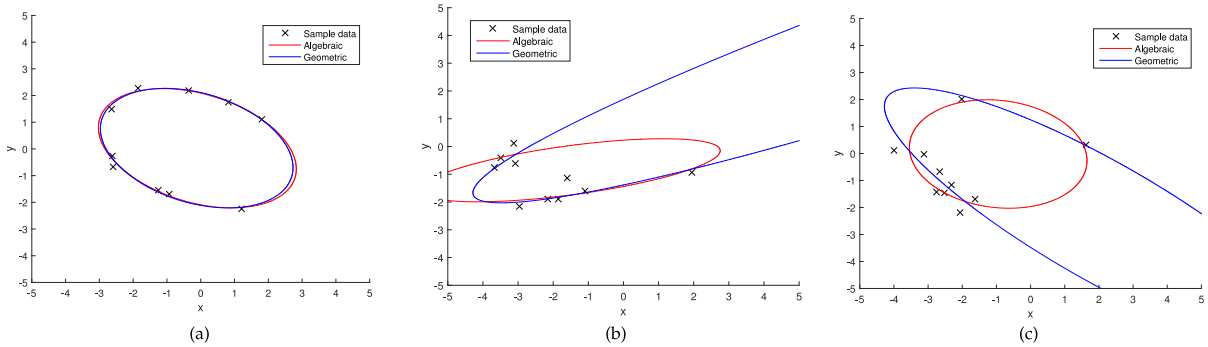

Fig. 1. A comparison between algebraic fitting [5] and geometric fitting [1]. (a) They produce close results if there is little noise and the sample points are uniform. (b)-(c) With heavy noise, non-uniform sampling, or bad initialization of the parameters, geometric fitting may produce a bad local minimum solution due to its non-convexity.

When enforcing the semi-definiteness constraint at the high dimensional case, the ellipsoid fitting problem is often formulated as semi-definite programs (SDP) [23]. Calafiore [5] first proposed a practicable SDP approach for multidimensional ellipsoid fitting, which minimizes the ℓ_2 norm of the residual error. The corresponding SDP is

$$\min_{\mathbf{q},\mathbf{t}} \sum_{i=1}^{n} t_i,$$
$$s.t. \quad \mathbf{A} \succeq \mathbf{0}, \quad \mathcal{S}_p(\mathcal{M}(\mathbf{q})) \succeq \mathbf{0}, \quad \mathrm{trace}(\mathcal{S}_p(\mathcal{M}(\mathbf{q})) = 1, \qquad (4)$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & r_1 & & & \\ r_1 & t_1 & & & \\ & & \ddots & & \\ & & & 1 & r_n \\ & & & r_n & t_n \end{bmatrix},$$

with the auxiliary variables $\mathbf{t} = [t_1, \ldots, t_n]^T$ to be optimized. Instead of the quadratic constraint $\|\mathbf{q}\|_2 = 1$ in LLS, here the linear constraint $\mathrm{trace}(\mathcal{S}_p(\mathcal{M}(\mathbf{q})) = 1$ is adopted to avoid the trivial solution $\mathbf{q} = \mathbf{0}$ and also make the problem convex. For different norms of the residual error, the objective function and the form of $\mathbf{A}$ vary [24], and the sizes of $\mathbf{A}$ and $\mathcal{S}_p(\mathcal{M}(\mathbf{q}))$ grow linearly with n and p, respectively. So it is intractable when dealing with a lot of data or very high dimensionality.

Since the computation complexity of SDP grows with the sizes of positive semi-definite matrices, reducing the size of $\mathbf{A}$ in (4) is an effective way to reduce the computation complexity. In [24], Ying et al. proposed a novel fast SDP method by defining a new norm for the residual error. Their SDP can be formulated as

$$\min_{\mathbf{q},t} t,$$
$$s.t. \quad \mathbf{A} \succeq \mathbf{0}, \quad \mathcal{S}_p(\mathcal{M}(\mathbf{q})) \succeq \mathbf{0}, \quad \mathrm{trace}(\mathcal{S}_p(\mathcal{M}(\mathbf{q})) = 1, \qquad (5)$$

where

$$\mathbf{A} = \begin{bmatrix} t & & & & r_1 & r_{d+1} & & r_n \\ & t & & & r_2 & r_{d+2} & & 0 \\ & & \ddots & & \vdots & \vdots & \cdots & \vdots \\ & & & t & r_d & r_{2d} & & 0 \\ r_1 & r_2 & \cdots & r_d & t & & & \\ r_{d+1} & r_{d+2} & \cdots & r_{2d} & & t & & \\ & & \vdots & & & & \ddots & \\ r_n & 0 & \cdots & 0 & & & & t \end{bmatrix},$$

with $d = \mathrm{ceil}(\sqrt{n})$. Correspondingly, the size of $\mathbf{A}$ is reduced from $O(n)$ to $O(\sqrt{n})$. The computation complexity of Ying et al.'s method is thus $O(p^6 + np^4 + n^{\frac{3}{2}}p^2)$ *for each iteration*, which is much lower than other SDP methods. However, it is still costly when the number of data points is large. The storage expense is at least $c_1 = m \times (2d) \times (2d) \geq 4nm \approx 2np^2$ units. It is an acceptable result comparing to Calafiore's method, but we will show in Section 2 that there is still potential for further reduction in memory cost.

In this paper, we apply the alternating direction method of multipliers (ADMM) [15] for solving the ellipsoid fitting problem directly, rather than reformulating it into SDP problems. By ADMM, the extra positive semi-definite matrix $\mathbf{A}$ in (4) and (5) is completely gone, hence further saving the complexity significantly. As a matter of fact, the complexity of our method is only $O(p^6 + np^4) + O(p^3)$, where $O(p^6 + np^4)$ results from preprocessing data **once**. Since the semi-definiteness is still enforced, our method still results in valid ellipsoid fittings even in the case of heavy noise (see Fig. 2). Furthermore, we also reduce the storage requirement significantly to about $\frac{1}{2}np^2$. Finally, the implementation of our method is much simpler than SDP based methods, as the latter require special software packages for SDP.

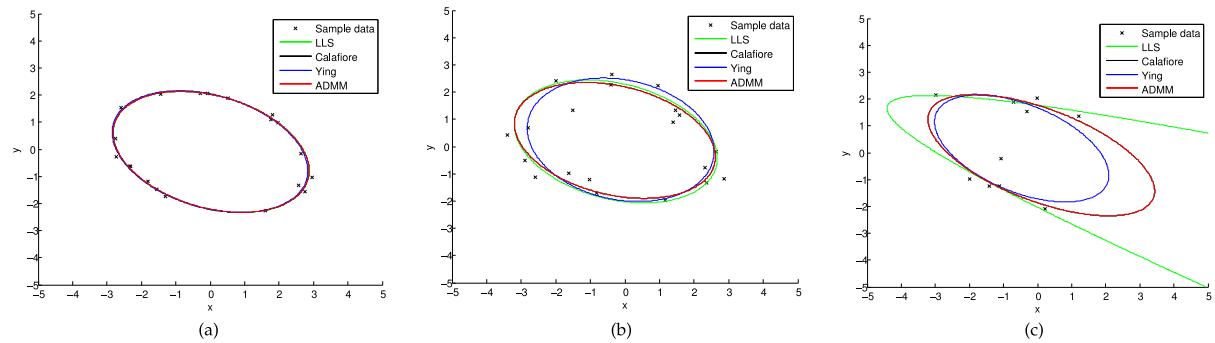

Fig. 2. An example of ellipse fitting with the presence of noise. (a)-(c) are the cases with 5, 15, and 25 percent Gaussian noise, respectively. The LLS solution may degenerate to a hyperbola when noise are strong (c), while other methods remain stable due to the semi-definiteness constraint.

2 Our Method

2.1 Problem Formulation

In our current effort, we aim at solving the ellipsoid-specific fitting problem (2) where the norm of residual error is the ℓ_2 norm, i.e.,

$$\min_{\mathbf{q}} \mathbf{q}^T\mathbf{K}\mathbf{q}, \tag{6}$$

$$s.t. \quad \mathcal{S}_p(\mathcal{M}(\mathbf{q})) \succeq \mathbf{0}, \quad \mathrm{trace}(\mathcal{S}_p(\mathcal{M}(\mathbf{q}))) = 1, \tag{7}$$

where as in SDP methods $\mathrm{trace}(\mathcal{S}_p(\mathcal{M}(\mathbf{q}))) = 1$ is added to prevent the trivial solution $\mathbf{q} = \mathbf{0}$. For other norms of residual error, which can be general and not limited to the norms proposed by Ying et al. [24], ADMM can be applied as well but the details would differ slightly.

Since $\|\mathbf{q}\|_2 \neq \|\mathcal{M}(\mathbf{q})\|_F$ ($\|\cdot\|_F$ is the Frobenius norm of matrices), which is needed when implementing ADMM (see the transition from (18) to (19)), we define new linear operators $\hat{\mathcal{V}}(\mathbf{Q}) = \hat{\mathbf{q}} = [Q_{11}, \sqrt{2}Q_{21}, Q_{22}, \sqrt{2}Q_{31}, \sqrt{2}Q_{32}, Q_{33}, \ldots, Q_{(p+1)(p+1)}]^T \in \mathbb{R}^m$ and its inverse operator $\hat{\mathcal{M}}(\hat{\mathbf{q}}) = \mathbf{Q}$. With the new linear operators, $\|\hat{\mathbf{q}}\|_2 = \|\hat{\mathcal{M}}(\hat{\mathbf{q}})\|_F$. Accordingly, the residual error $r_i = \hat{\mathbf{d}}_i^T\hat{\mathbf{q}}$, where $\hat{\mathbf{d}}_i = [x_{i,1}^2, \sqrt{2}x_{i,1}x_{i,2}, x_{i,2}^2, \sqrt{2}x_{i,3}x_{i,1}, \sqrt{2}x_{i,3}x_{i,2}, x_{i,3}^2, \ldots, 1]^T \in \mathbb{R}^m$.

Then problem (6)-(7) can be rewritten as:

$$\min_{\hat{\mathbf{q}}} \hat{\mathbf{q}}^T\hat{\mathbf{K}}\hat{\mathbf{q}}, \tag{8}$$

$$s.t. \quad \mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{q}})) \succeq \mathbf{0}, \quad \mathbf{c}^T\hat{\mathbf{q}} = 1, \tag{9}$$

where $\hat{\mathbf{K}} = \hat{\mathbf{D}}\hat{\mathbf{D}}^T, \hat{\mathbf{D}} = [\hat{\mathbf{d}}_1, \ldots, \hat{\mathbf{d}}_n]$, and $\mathbf{c} = [1, 0, 1, 0, 0, 1, \ldots, 0]^T \in \mathbb{R}^m$ is the binary mask vector such that $\mathbf{c}^T\hat{\mathbf{q}} = \mathrm{trace}(\mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{q}})))$.

2.2 ADMM for Solving (8)-(9)

Before applying ADMM we first introduce an auxiliary block of variables $\hat{\mathbf{s}}$ in order to make the update of every block of variables easy. Namely, we rewrite (8)-(9) as

$$\min_{\hat{\mathbf{q}},\hat{\mathbf{s}}} \hat{\mathbf{q}}^T\hat{\mathbf{K}}\hat{\mathbf{q}}, \tag{10}$$

$$s.t. \quad \hat{\mathbf{s}} = \hat{\mathbf{q}}, \quad \mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{s}})) \succeq \mathbf{0}, \quad \mathbf{c}^T\hat{\mathbf{q}} = 1. \tag{11}$$

ADMM [15] updates the variables alternately by minimizing over the augmented Lagrangian function of the problem and then updates the Lagrange multiplier. The partial augmented Lagrangian function of problem (10)-(11) is

$$L(\hat{\mathbf{q}}, \hat{\mathbf{s}}, \boldsymbol{\mu}) = \hat{\mathbf{q}}^T\hat{\mathbf{K}}\hat{\mathbf{q}} + \langle\boldsymbol{\mu}, \hat{\mathbf{s}} - \hat{\mathbf{q}}\rangle + \frac{\beta}{2}\|\hat{\mathbf{s}} - \hat{\mathbf{q}}\|_2^2,$$

where $\langle\mathbf{a}, \mathbf{b}\rangle = \mathbf{a}^T\mathbf{b}$ is the inner product, $\boldsymbol{\mu}$ is the Lagrange multiplier and $\beta > 0$ is the penalty parameter. The extra constraints

$$\mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{s}})) \succeq \mathbf{0} \text{ and } \mathbf{c}^T\hat{\mathbf{q}} = 1$$

will be enforced when minimizing L with respect to $\hat{\mathbf{s}}$ and $\hat{\mathbf{q}}$, respectively.

Then each iteration of ADMM consists of four steps:

1) Update $\hat{\mathbf{q}}$:

$$\hat{\mathbf{q}}_{k+1} = \underset{\hat{\mathbf{q}},\mathbf{c}^T\hat{\mathbf{q}}=1}{\mathrm{argmin}} L(\hat{\mathbf{q}}, \hat{\mathbf{s}}_k, \boldsymbol{\mu}_k). \tag{12}$$

2) Update $\hat{\mathbf{s}}$:

$$\hat{\mathbf{s}}_{k+1} = \underset{\hat{\mathbf{s}},\mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{s}}))\succeq\mathbf{0}}{\mathrm{argmin}} L(\hat{\mathbf{q}}_{k+1}, \hat{\mathbf{s}}, \boldsymbol{\mu}_k). \tag{13}$$

3) Update $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \beta_k(\hat{\mathbf{s}}_{k+1} - \hat{\mathbf{q}}_{k+1}). \tag{14}$$

4) Update β:

$$\beta_{k+1} = \min(\rho\beta_k, \beta_{\max}), \tag{15}$$

where we utilize the adaptive penalty scheme [15]:

$$\rho = \begin{cases} \rho_0, & \text{if } \max\{\|\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k+1}\|_\infty, \|\hat{\mathbf{s}}_{k+1} - \hat{\mathbf{s}}_k\|_\infty\} < \varepsilon_2, \\ 1, & \text{otherwise,} \end{cases} \tag{16}$$

and $\beta_{\max}$ is an upper bound of β, in which $\rho_0 \geq 1$, $\varepsilon_2 \in (0, 1)$, and $\|\cdot\|_\infty$ is the ℓ_∞ norm of vectors.

The iteration terminates when

$$\|\hat{\mathbf{s}}_{k+1} - \hat{\mathbf{q}}_{k+1}\|_\infty < \varepsilon_1, \text{and}$$

$$\max\{\|\hat{\mathbf{q}}_{k+1} - \hat{\mathbf{q}}_k\|_\infty, \|\hat{\mathbf{s}}_{k+1} - \hat{\mathbf{s}}_k\|_\infty\} < \varepsilon_2$$

are satisfied, where $\varepsilon_1 \in (0, 1)$.

In the following, we give details of updating $\hat{\mathbf{q}}$ and $\hat{\mathbf{s}}$.

2.3 Update $\hat{\mathbf{q}}$

The update of $\hat{\mathbf{q}}$ is a linearly constrained quadratic optimization problem:

$$\hat{\mathbf{q}}_{k+1} = \underset{\hat{\mathbf{q}},\mathbf{c}^T\hat{\mathbf{q}}=1}{\mathrm{argmin}} \hat{\mathbf{q}}^T\hat{\mathbf{K}}\hat{\mathbf{q}} + \langle\boldsymbol{\mu}_k, \hat{\mathbf{s}}_k - \hat{\mathbf{q}}\rangle + \frac{\beta_k}{2}\|\hat{\mathbf{s}}_k - \hat{\mathbf{q}}\|_2^2.$$

By introducing a Lagrange multiplier, one can easily deduce that the above problem has a closed form solution:

$$\hat{\mathbf{q}}_{k+1} = \mathbf{G}^{-1}\left(\frac{1 - \mathbf{c}^T\mathbf{G}^{-1}\mathbf{g}}{\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}}\mathbf{c} + \mathbf{g}\right), \tag{17}$$

where $\mathbf{G} = 2\hat{\mathbf{K}} + \beta_k\mathbf{I}$ and $\mathbf{g} = \boldsymbol{\mu}_k + \beta_k\hat{\mathbf{s}}_k$.

When computing $\hat{\mathbf{q}}_{k+1}$, one may calculate the eigenvalue decomposition $\mathbf{U}\Lambda\mathbf{U}^T$ of $\hat{\mathbf{K}}$ in advance, which is done only once. Then $\mathbf{G}^{-1}$ can be represented as $\mathbf{U}(2\Lambda + \beta_k\mathbf{I})^{-1}\mathbf{U}^T$. Without explicitly computing $\mathbf{G}^{-1}$, $\mathbf{G}^{-1}\mathbf{z}$ can be computed as $\mathbf{U}\Big((2\Lambda + \beta_k\mathbf{I})^{-1}(\mathbf{U}^T\mathbf{z})\Big)$. These tricks can save computation greatly when p is large.

2.4 Update $\hat{\mathbf{s}}$

The subproblem for updating $\hat{\mathbf{s}}$ is

$$\hat{\mathbf{s}}_{k+1} = \underset{\hat{\mathbf{s}},\mathcal{S}_p(\hat{\mathcal{M}}(\hat{\mathbf{s}}))\succeq\mathbf{0}}{\mathrm{argmin}} \frac{\beta_k}{2}\left\|\hat{\mathbf{s}} - \hat{\mathbf{q}}_{k+1} + \frac{\boldsymbol{\mu}_k}{\beta_k}\right\|_2^2. \tag{18}$$

By the equivalence between the ℓ_2 norm of $\hat{\mathbf{q}}$ and the Frobenius norm of $\hat{\mathcal{M}}(\hat{\mathbf{q}})$, we may rewrite the above in a matrix form:

$$\mathbf{S}_{k+1} = \underset{\mathbf{S},\mathcal{S}_p(\mathbf{S})\succeq\mathbf{0}}{\mathrm{argmin}} \frac{\beta_k}{2}\|\mathbf{S} - \mathbf{R}_{k+1}\|_F^2, \tag{19}$$

where $\mathbf{R}_{k+1} = \hat{\mathcal{M}}(\hat{\mathbf{q}}_{k+1} - \boldsymbol{\mu}_k/\beta_k)$.

Note that there is no constraint on the $(p+1)$th column and row of $\mathbf{S}$. So for the optimal solution $\mathbf{S}$, $S_{ij} = R_{ij}$ when $\max(i, j) = p + 1$, and we may just focus on the leading $p \times p$ principle submatrix $\tilde{\mathbf{S}}$ of $\mathbf{S}$:

$$\min_{\tilde{\mathbf{S}}} \|\tilde{\mathbf{S}} - \mathcal{S}_p(\mathbf{R}_{k+1})\|_F^2, \quad s.t. \quad \tilde{\mathbf{S}} \succeq \mathbf{0}. \tag{20}$$

Its solution is

$$\tilde{\mathbf{S}} = \mathbf{V}\Sigma_+\mathbf{V}^T, \tag{21}$$

TABLE 1
Average Runtime (in Seconds) at Different Numbers of Sample Points in Different Dimensions

	2D						3D						10D				
Point No.	10	10^2	10^3	10^4	10^5	10^6	10	10^2	10^3	10^4	10^5	10^6	10^2	10^3	10^4	10^5	10^6
Calafiore [5]	0.156	0.436	4.183	418.5	-	-	0.222	0.494	4.903	510.2	-	-	0.702	7.144	-	-	-
Ying [24]	0.168	0.181	0.251	1.411	10.09	202.9	0.206	0.227	0.276	2.401	11.95	249.5	0.386	0.692	3.834	57.40	1445
ADMM	**0.001**	**0.001**	**0.002**	**0.019**	**0.189**	**1.893**	**0.001**	**0.001**	**0.003**	**0.025**	**0.226**	**2.148**	**0.029**	**0.049**	**0.168**	**1.346**	**13.38**

where $\mathbf{V}\Sigma\mathbf{V}^T$ is the EVD of $\mathcal{S}_p(\mathbf{R}_{k+1})$, the leading $p \times p$ principle submatrix of $\mathbf{R}_{k+1}$, and $(\Sigma_+)_{ij} = \max(\Sigma_{ij}, 0)$ truncates the negative eigenvalues.

After obtaining $\mathbf{S}_{k+1}$, $\hat{\mathbf{s}}$ is updated as $\hat{\mathbf{s}}_{k+1} = \hat{\mathcal{V}}(\mathbf{S}_{k+1})$.

2.5 Complexity of Our Algorithm

As we have shown in the previous section, our algorithm is efficient due to closed form solutions in each iteration. One main bottleneck of our algorithm is the calculation of $\hat{\mathbf{K}} \in \mathbb{R}^{m\times m}$ and its EVD, yielding $O(nm^2 + m^3) = O(np^4 + p^6)$ computation complexity. **But these only need to be calculated once during the whole iteration.** The other major computation cost is updating $\hat{\mathbf{s}}$ *in every iteration*, where we have to compute the EVD of $\mathcal{S}_p(\mathbf{R}_{k+1})$ and then form $\tilde{\mathbf{S}}$ as (21), resulting in additional $O(p^3)$ complexity. So the overall computation complexity of our algorithm is $O(np^4 + p^6) + O(p^3)$, where the former does *not* depend on numerical precision. Note that this computation complexity is of the same order of the LLS method. This means that our algorithm is very efficient.

Our algorithm also has great advantage on storage complexity. The main expense of the memory in our algorithm is the storage of $\hat{\mathbf{D}}$, which costs for $c_2 = nm \approx \frac{1}{2}np^2$ units of memory. It is at least four times reduction comparing to Ying's SDP method.

Besides computation complexity and storage complexity, the implementation complexity of our method is also much lower than that of SDP based methods, as our method does not rely on special software packages for SDP.

3 Experimental Results

In this section, we conduct extensive experiments to verify the efficiency, robustness, and accuracy of the proposed method.[2] We compare our ADMM method with LLS [6], Calafiore's algorithm [5] where ℓ_2 norm is chosen for the residual error, and Ying et al.'s algorithm [24] where the newly invented $2_{r\times c}$ matrix norm is chosen to induce the vector norm for the residual error. We implement our method and other algorithms in Matlab R2013a, where the implementations of Calafiore's and Ying et al.'s algorithms are based on the SDPT3 software package [22]. For ADMM, we set $\beta_0 = 0.1$, $\rho_0 = 1.02$, $\varepsilon_1 = \varepsilon_2 = 10^{-3}$. The initial values of $\boldsymbol{\mu}$, $\mathbf{s}$, and $\mathbf{q}$ are simply set to $\mathbf{0}$. For other methods, we set default values as specified by respective authors. All the experiments are done on a PC with an Intel Pentium Dual core 3.40GHz CPU and 8GB RAM.

3.1 Synthetic Data

We first test with synthetic data. We sample data points uniformly from the surfaces of ellipsoids. Then we add Gaussian noise to each of the points, where the standard deviation of noise varies from 5 to 25 percent of the diameters of the ellipsoids. Finally, we fit ellipsoids to the data points. We repeat the experiment for 100 times on different levels of noise, numbers of data points, and dimensions of the data points.

2. Codes are available at *http://www.cis.pku.edu.cn/faculty/vision/zlin/zlin.htm*.

3.1.1 Ellipsoid-Specificity

Despite the semi-definiteness constraints, it is instructive to observe the performance on synthetic data to prove the ellipsoid-specificity of our algorithm. Fig. 2 shows an example. With the presence of noise, the LLS algorithm may result in a general conic curve (Fig. 2c), while other algorithms all produce ellipses. One should be reminded that our ADMM algorithm produces exactly the same results as Calafiore's because they solve exactly the same optimization problem.

3.1.2 Efficiency

Now we compare the computation efficiency of various algorithms. As we have shown, LLS cannot ensure ellipsoid-specific results. So we do not show the results of LLS in this section. For different data sizes in different dimensionality, we record the average runtime of the 100 trials of different algorithms and present it in Table 1.

From the Table 1, we can see that Ying et al.'s method is much faster than Calafiore's and the latter cannot scale up for a large amount of data points. Nonetheless, our ADMM method is further much faster than Ying et al.'s. Even with a million data samples in 10D, the computation time of our method is still affordable.

3.1.3 Accuracy

We now compare the fitting accuracy in terms of the residual error measured in ℓ_1 norm, ℓ_2 norm, and ℓ_∞ norm, respectively. In this experiment, we aim at fitting 2D ellipses. The results are shown in Fig. 3.

Again, one should be reminded that Calafiore's and ADMM produce the same result but ADMM is at least hundreds times faster than Calafiore's. From Fig. 3, we can see that both of them result in lower residual errors than Ying et al.'s, measured in whichever norm in consideration.

3.2 Real Data—Camera Calibration

We further compare the ellipsoid fitting algorithms with real data. As mentioned in Introduction, camera self-calibration also involves an ellipsoid-fitting problem. In this section, we focus on the calibration method proposed by Zhang [26], which is a highly cited flexible calibration technique. For completeness, we quote part of the deductions here.

The relationship between a 3D point $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$ represented in homogeneous coordinates and its image projection $\tilde{\mathbf{m}} = [x, y, 1]^T$ is

$$\lambda\tilde{\mathbf{m}} = \mathbf{P}[\mathbf{R}, \mathbf{t}]\tilde{\mathbf{M}}, \tag{22}$$

where $\mathbf{P}$ encodes the intrinsic parameters of the camera, $\mathbf{R}$ and $\mathbf{t}$ represents the rotation and translation which relates the world coordinate system to the camera coordinate system, respectively, and λ is the projective depth. Without loss of generality, we assume that the model plane is on $Z = 0$ of the world coordinate system. Then we have

$$\lambda\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{P}[\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{P}[\mathbf{R}_1, \mathbf{R}_2, t]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{23}$$

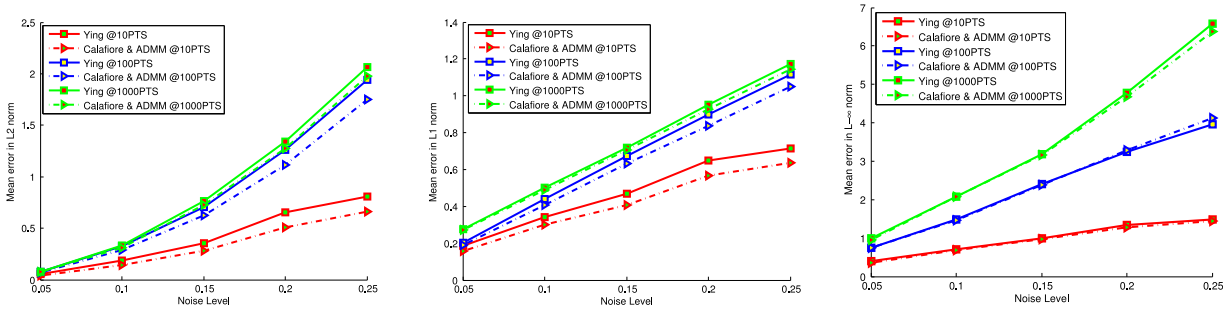

Fig. 3. Residual errors of Calafiore's, Yin et al.'s, and our ADMM methods, under different noise levels and different numbers of data points, measured in ℓ_2 norm (a), ℓ_1 norm (b), and ℓ_∞ norm (c).

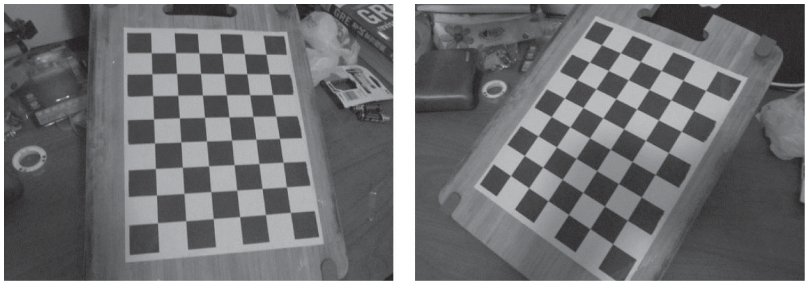

Fig. 4. Two images of a checkerboard used in our experiment.

TABLE 2
Success Rate, Mean Reprojection Error (in Pixels), and Mean Runtime (in Seconds) in Camera Intrinsic Parameter Estimation

Noise Level	0.5%			1%			1.5%		
	Succ. Rate	Ave. Error	Time	Succ. Rate	Ave. Error	Time	Succ. Rate	Ave. Error	Time
LLS [6]	**100%**	2.64	**0.003**	97%	5.26	**0.003**	90%	8.39	**0.003**
Calafiore [5]	**100%**	**2.52**	0.229	**100%**	**4.53**	0.232	**100%**	**6.60**	0.238
Ying [24]	**100%**	2.78	0.205	**100%**	4.85	0.206	**100%**	7.10	0.205
ADMM	**100%**	**2.52**	0.021	**100%**	**4.53**	0.017	**100%**	**6.60**	0.018

Denote $\mathbf{H} = \mathbf{P}[\mathbf{R}, \mathbf{t}] = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$, which can be estimated given an image of the model plane. Due to the orthogonality between $\mathbf{R}_1$ and $\mathbf{R}_2$, we have

$$\mathbf{h}_1^T \mathbf{B} \mathbf{h}_2 = 0 \text{ and } \mathbf{h}_1^T \mathbf{B} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{B} \mathbf{h}_2, \tag{24}$$

where $\mathbf{B} = \mathbf{P}^{-T}\mathbf{P}^{-1} \in \mathbb{R}^{3\times 3}$ is a positive semi-definite matrix, which can be represented by $\mathbf{b} = \mathcal{V}(\mathbf{B})$.

Then equations in (24) can be reformulated as a linear system for $\mathbf{b}$:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0, \tag{25}$$

where $\mathbf{v}_{ij} = [h_{i,1}h_{j,1}, h_{i,1}h_{j,2} + h_{i,2}h_{j,1}, h_{i,2}h_{j,2}, h_{i,1}h_{j,3} + h_{i,3}h_{j,1}, h_{i,3}h_{j,2} + h_{i,2}h_{j,3}, h_{i,3}h_{j,3}]^T$, $i, j \in \{1, 2\}$. If l images of the model plane are observed, by stacking l such equations as (25), we have

$$\mathbf{V}^T \mathbf{b} = \mathbf{0}, \tag{26}$$

where $\mathbf{V}$ collects all $\mathbf{v}_{ij}$'s as its columns.

In [26], Zhang simply used LLS to solve for $\mathbf{b}$, i.e. the eigenvector of $\mathbf{V}\mathbf{V}^T$ associated with the smallest eigenvalue. However, since the semi-definiteness is not enforced, when the noise are strong and samples are insufficient, the resulted matrix $\mathbf{B}$ may not be positive semi-definite, which makes the recovery of intrinsic parameters infeasible.

To ensure the positive semi-definiteness of $\mathbf{B}$, we may aim at solving

$$\min_{\mathbf{b}} \quad \|\mathbf{V}\mathbf{b}\|_2, \quad s.t. \quad \mathcal{M}(\mathbf{b}) \succeq \mathbf{0} \text{ and } \operatorname{trace}(\mathcal{M}(\mathbf{b})) = 1, \tag{27}$$

instead, which can be solved by Calafiore's and Ying et al.'s algorithms. When applying our ADMM, we may reformulate (27) as

$$\min_{\hat{\mathbf{b}}} \quad \|\hat{\mathbf{V}}\hat{\mathbf{b}}\|_2, \quad s.t. \quad \hat{\mathcal{M}}(\hat{\mathbf{b}}) \succeq \mathbf{0} \text{ and } \operatorname{trace}(\hat{\mathcal{M}}(\hat{\mathbf{b}})) = 1, \tag{28}$$

where $\hat{\mathbf{b}} = \hat{\mathcal{M}}(\mathbf{B})$ and $\hat{\mathbf{V}}$ collects all $\hat{\mathbf{v}}_{ij}$'s as its columns, in which $\hat{\mathbf{v}}_{ij} = [h_{i,1}h_{j,1}, \frac{1}{\sqrt{2}}(h_{i,1}h_{j,2} + h_{i,2}h_{j,1}), h_{i,2}h_{j,2}, \frac{1}{\sqrt{2}}(h_{i,1}h_{j,3} + h_{i,3}h_{j,1}), \frac{1}{\sqrt{2}}(h_{i,3}h_{j,2} + h_{i,2}h_{j,3}), h_{i,3}h_{j,3}]^T$.

To prepare real data, we take 8 ($l = 8$, $n = 2l = 16$) photos of a checkerboard (see Fig. 4) from different angles and label the feature points manually. To examine the influence of noise, we further add different levels of Gaussian noise to the image coordinates of the feature points. We then estimate the intrinsic and extrinsic parameters of the camera and re-project the ground-truth world coordinate to the image coordinate using the imaging model (23). We present the success rate (i.e., the percent of trials that produce positive semi-definite $\mathbf{B}$), the re-projection errors, and the running time of different approaches in Table 2.

From the table, we can see that LLS does not always guarantee a successful calibration, although its computation is the fastest.

Calafiore's, Ying et al.'s, and our ADMM methods always produce valid calibration results, in which ADMM (and Calafiore's) is the most accurate and is also much faster than Calafiore's and Ying et al.'s methods.

4 Conclusion

We have proposed using ADMM to solve the ellipsoid-specific fitting problem directly. Our approach does not introduce extra semi-definiteness constraint and each iteration consists of low-weight closed-form solutions. Thus our method is much faster than SDP based methods and still guarantee the positive semi-definiteness. Extensive experiments show that our method outperforms the state-of-the-art methods by a large margin, in both speed and accuracy. Our method enables at least four times storage complexity reduction over SDP based methods. The implementation of our method is also much simpler than SDP based methods, as the latter rely on special software packages for SDP.

Acknowledgments

Zhouchen Lin is supported by 973 Program of China (grant no. 2015CB352502), National Natural Science Foundation of China (grant nos. 61272341 and 61231002), and Microsoft Research Asia Collaborative Research Program.

References

[1] S. J. Ahn, W. Rauh, H. S. Cho, and H.-J. Warnecke, "Orthogonal distance fitting of implicit curves and surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 620–638, May 2002.

[2] S. J. Ahn, W. Rauh, and H.-J. Warnecke, "Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola," *Pattern Recognit.*, vol. 34, no. 12, pp. 2283–2303, 2001.

[3] M. M. Blane, Z. Lei, H. Çivi, and D. B. Cooper, "The 3L algorithm for fitting implicit polynomial curves and surfaces to data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 298–313, Mar. 2000.

[4] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[5] G. Calafiore, "Approximation of n-dimensional data using spherical and ellipsoidal primitives," *IEEE Trans. Syst., Man Cybern., Part A, Syst. Humans*, vol. 32, no. 2, pp. 269–278, Mar. 2002.

[6] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.

[7] A. Forbes, "Generalised regression problems in metrology," *Numerical Algorithms*, vol. 5, no. 10, pp. 523–533, 1993.

[8] W. Gander, G. H. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *BIT Numerical Math.*, vol. 34, no. 4, pp. 558–578, 1994.

[9] Z. Ge, B. Sahiner, H.-P. Chan, L. M. Hadjiiski, P. N. Cascade, N. Bogot, E. A. Kazerooni, J. Wei, and C. Zhou, "Computer-aided detection of lung nodules: False positive reduction using a 3D gradient field method and 3D ellipsoid fitting," *Med. Phys.*, vol. 32, no. 8, pp. 2443–2454, 2005.

[10] N. Grammalidis and M. G. Strintzis, "Head detection and tracking by 2-D and 3-D ellipsoid fitting," in *Proc. Comput. Graph. Int.*, 2000, pp. 221–226.

[11] M.-Y. Ju, J.-S. Liu, S.-P. Shiang, Y.-R. Chien, K.-S. Hwang, and W.-C. Lee, "A novel collision detection method based on enclosed ellipsoid," in *Proc. IEEE Int. Conf. Robot. Automation*, 2001, vol. 3, pp. 2897–2902.

[12] M. Kleinsteuber and K. Hüper, "Approximate geometric ellipsoid fitting: A CG-approach," in *Recent Advances in Optimization and its Applications in Engineering*. New York, NY, USA: Springer, 2010, pp. 73–82.

[13] K. K. Lee and W. C. Yoon, "Adaptive classification with ellipsoidal regions for multidimensional pattern classification problems," *Pattern Recognit. Lett.*, vol. 26, no. 9, pp. 1232–1243, 2005.

[14] Q. Li and J. G. Griffiths, "Least squares ellipsoid specific fitting," in *Proc. Geometric Model. Process.*, 2004, pp. 335–340.

[15] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.

[16] S. Mahdavi and S. E. Salcudean, "3D prostate segmentation based on ellipsoid fitting, image tapering and warping," in *Proc. 30th IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2008, pp. 2988–2991.

[17] I. Markovsky, A. Kukush, and S. Van Huffel, "Consistent least squares fitting of ellipsoids," *Numerische Mathematik*, vol. 98, no. 1, pp. 177–194, 2004.

[18] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," *Comput Graph. Forum*, vol. 22, pp. 775–792, 2003.

[19] E. Rimon and S. P. Boyd, "Obstacle collision detection using best ellipsoid fit," *J. Intell. Robot. Syst.*, vol. 18, no. 2, pp. 105–126, 1997.

[20] P. L. Rosin, "A note on the least squares fitting of ellipses," *Pattern Recognit. Lett.*, vol. 14, no. 10, pp. 799–808, 1993.

[21] S. Sivapalan, D. Chen, S. Denman, S. Sridharan, and C. Fookes, "3D ellipsoid fitting for multi-view gait recognition," in *Proc. IEEE Int. Conf. Adv. Video Signal-Based Surveillance*, 2011, pp. 355–360.

[22] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3-A MATLAB software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, no. 1-4, pp. 545–581, 1999.

[23] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[24] X. Ying, L. Yang, and H. Zha, "A fast algorithm for multidimensional ellipsoid-specific fitting by minimizing a new defined vector norm of residuals using semidefinite programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1856–1863, Sep. 2012.

[25] J. Yu, H. Zheng, S. R. Kulkarni, and H. V. Poor, "Outlier elimination for robust ellipse and ellipsoid fitting," in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2009, pp. 33–36.

[26] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.