

PAPER

# Optimization without Minimization Search: Constraint Satisfaction by Orthogonal Projection with Applications to Multiview Triangulation

Kenichi KANATANI<sup>†a)</sup>, Yasuyuki SUGAYA<sup>††</sup>, *Members*, and Hiroataka NIITSUMA<sup>†</sup>, *Nonmember*

**SUMMARY** We present an alternative approach to what we call the “standard optimization”, which minimizes a cost function by searching a parameter space. Instead, our approach “projects” in the joint observation space onto the manifold defined by the “consistency constraint”, which demands that any minimal subset of observations produce the same result. This approach avoids many difficulties encountered in the standard optimization. As typical examples, we apply it to line fitting and multiview triangulation. The latter produces a new algorithm far more efficient than existing methods. We also discuss the optimality of our approach.

**key words:** consistency constraint satisfaction, orthogonal projection, line fitting, multiview triangulation, trifocal tensor

## 1. Introduction

For extracting a geometric structure from noisy images, numerical optimization is vital. A widely adopted approach is the *standard optimization*, minimizing a cost function by searching a parameter space. In this paper, we present an alternative approach of *orthogonally projecting* the input in the joint input space onto the manifold defined by the *consistency constraint*, which demands that any minimal subset of observations produce the same result. We show how this approach avoids many difficulties encountered in the standard optimization.

We first describe the standard optimization in Sect. 2 and summarize existing global optimization techniques in Sect. 3. We describe our orthogonal projection approach in Sect. 4 and apply it to line fitting in Sect. 5. Then, we apply our approach to multiview triangulation in Sect. 6 and demonstrate by experiments in Sect. 7 that the resulting algorithm is far more efficient than existing methods. In Sect. 8, we conclude and discuss the optimality of our approach.

## 2. Standard Optimization

Given  $M$  observations  $\mathbf{x}_\kappa$ ,  $\kappa = 0, \dots, M - 1$ , we want to estimate a parameter  $\boldsymbol{\theta}$  that specifies the structure

Manuscript received August 24, 2009.

Manuscript revised April 13, 2010.

<sup>†</sup>The authors are with the Department of Computer Science, Okayama University, Okayama-shi, 700-8530 Japan.

<sup>††</sup>The author is with the Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi-shi, 441-8580 Japan.

a) E-mail: kanatani@suri.cs.okayama-u.ac.jp

DOI: 10.1587/transinf.E93.D.1

that should exist in the input images. The estimation is based on the knowledge that  $\mathbf{x}_\kappa$  should ideally satisfy some constraint parameterized by  $\boldsymbol{\theta}$ , but it is violated in the presence of noise. Let  $E(\boldsymbol{\theta}; \mathbf{x}_0, \dots, \mathbf{x}_{M-1})$  measure the cost (also called “energy”) of this violation. We compute the value of  $\boldsymbol{\theta}$  that minimizes  $E$ . A popular approach is to express the ideal value of  $\mathbf{x}_\kappa$ , given  $\boldsymbol{\theta}$ , in the form  $\bar{\mathbf{x}}_\kappa(\boldsymbol{\theta})$  and minimize

$$E = \sum_{\kappa=0}^{M-1} \|\bar{\mathbf{x}}_\kappa(\boldsymbol{\theta}) - \mathbf{x}_\kappa\|^2. \quad (1)$$

This includes what is known as *bundle adjustment*, where Eq. (1) is called the *reprojection error*.

## 3. Global Optimization

The major problem of the standard optimization as defined above is the difficulty of finding an absolute minimum of the cost  $E$ . The parameter space of  $\boldsymbol{\theta}$  is usually infinitely large and high-dimensional. Well known search techniques include Newton iterations and conjugate gradient search. For bundle adjustment, the Levenberg-Marquardt method is the standard tool. However, such gradient-based search may fall into local minima. In recent years, intensive efforts have been made to minimize  $E$  globally [4].

**Algebraic methods.** We let the derivatives of  $E$  with respect to the parameters be zero, compute all solutions exhaustively, and choose the one for which  $E$  is the smallest. In many vision applications, we obtain a set of algebraic equations, which reduces, via the Gröbner basis, to a single polynomial. However, its degree is usually very high, and numerical evaluation is unstable and inefficient.

**Branch and bound.** Introducing a function that gives a lower bound of  $E$  locally, we partition the parameter space into small cells, evaluating a representative value and a lower bound in each cell. Those cells for which the lower bound is larger than the values already evaluated in other cells are discarded; the remaining cells are recursively subdivided. However, the lower bounding process is often complicated, requiring a large amount of computation.

**Matrix inequality optimization.** Changing variables, we reduce the problem to polynomial minimization subject to matrix inequalities [10]. This has

the form of SDP (semidefinite program), which can be solved by a Matlab tool called GloptiPoly. The resulting solution is approximate, but it is theoretically proved to approach the true solution as the number of variables and the size of the accompanying matrices are increased. A complicate analysis and a large amount of computation are necessary if we want to reach a high accuracy.

$L_\infty$  **optimization.** Minimizing Eq. (1) is regarded as maximum likelihood (ML) if the noise is independent and identical Gaussian, which is appropriate in many applications. However, this makes global optimization difficult, so we replace the  $L_2$  norm in Eq. (1) by  $L_\infty$  [5], [8], [13]. Then, the cost  $E$  usually becomes quasi-convex. We gradually increase the threshold from 0 (or using binary search) and check if there exists a value of  $E$  above the threshold. This problem usually has the form of SOCP (second-order cone program), which can be solved by a Matlab tool called SeDuMi.

All these approaches need a complicated analysis and a large amount of computation, requiring various optimization tools, whose performance is not always guaranteed.

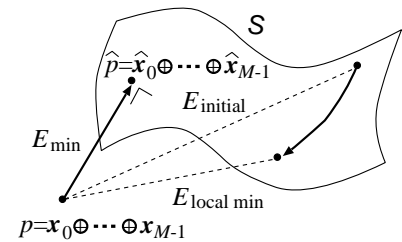
#### 4. Orthogonal Projection Approach

We now present a complementary approach: we do not minimize any function in the *parameter space*; we reach an optimal solution in the *data space* by *imposing constraints*. Our approach is motivated by the following observations:

1. If there were no noise, the parameter  $\theta$  could immediately be computed. In fact, algebraic procedures have been intensively studied for computing geometric properties from exact image data [7].
2. There exists a minimum number of (nondegenerate) observations (call it a *minimal set*) that can uniquely determine the value of  $\theta$ . For example, a line is uniquely determined by two points, an ellipse by five points, and a fundamental matrix by seven pairs of corresponding points.
3. Given redundant observations, we can choose from among them any minimal set for determining  $\theta$ . In the presence of noise, however, the solution depends on which minimal set is chosen.

The standard optimization overcomes this dependence on the choice of the minimal set by minimizing a cost  $E$  over the parameter space of  $\theta$ . Our alternative does *not* introduce any cost but *minimally corrects* the observations  $\mathbf{x}_0, \dots, \mathbf{x}_{M-1}$  into  $\hat{\mathbf{x}}_0, \dots, \hat{\mathbf{x}}_{M-1}$  to enforce the condition, which we call the *consistency constraint*, that *any choice of the minimal set result in the same solution*. Once this constraint is satisfied, we can choose any minimal set to determine  $\theta$ . By *minimally*, we mean that the correction is done in such a way that

$$E = \sum_{\kappa=0}^{M-1} \|\hat{\mathbf{x}}_\kappa - \mathbf{x}_\kappa\|^2, \quad (2)$$



**Fig. 1** Orthogonal projection of  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$  onto the manifold  $\mathcal{S}$  gives the minimum projection error  $E_{\min}$ . The standard optimization starts from some point in  $\mathcal{S}$  and searches the “inside” of  $\mathcal{S}$ , where the consistency constraint is always satisfied, but may stop at a local minimum  $E_{\text{local min}}$ .

which we call the *reprojection error*, is the smallest. Evidently, the solution is the same as the standard optimization using the cost of Eq. (1).

Our approach can be geometrically interpreted as follows. Let

$$\mathbf{F}(\mathbf{x}_0, \dots, \mathbf{x}_{M-1}) = \mathbf{0}, \quad (3)$$

be the consistency constraint, which may be a set of equations. This defines a manifold  $\mathcal{S}$  in the joint space of  $\mathbf{x}_0, \dots, \mathbf{x}_{M-1}$ . Our goal is to find a point  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1} \in \mathcal{S}$  closest to the observation  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$ ; the square distance  $|p\hat{p}|^2$  between  $p$  and  $\hat{p}$  is the reprojection error in Eq. (2). Thus, the solution is obtained by orthogonally projecting  $p$  onto  $\mathcal{S}$  (Fig. 1). The fact that minimum Euclidean distance is equivalent to orthogonal projection is well known [15], and this principle has been used in many geometric problems including curve/surface fitting [1], [2] and 3-D reconstruction from multiple images [3].

The standard optimization using Eq. (1) can be interpreted as follows. Introducing the parameter  $\theta$  is equivalent to *parameterizing*  $\mathcal{S}$ . We search the *inside* of  $\mathcal{S}$  to find a location (i.e., its “coordinates”  $\theta$ ) closest to  $p$ . Gradient-based search may fall into local minima, but finding a global minimum is difficult, as described earlier.

Our approach can be viewed as *geometric correction*, as described in [11], which optimally corrects observations so that they satisfy a given constraint. The first contribution of this paper is to show that parametric minimization problems can be converted into geometric correction problems via “consistency constraint”. Although first order expressions for geometric correction are given in [11], we need iterations to obtain a strictly optimal solution. The second contribution of this paper is to derive an iteration scheme for this, always projecting *original observations*, not intermediate solutions. We also give a geometric interpretation as to how our scheme works.

#### 5. Line Fitting

We now apply our approach to a simplest example: fitting a line to points. Although this does not produce practical benefits, since the problem is immediately solved by the standard optimization, this will illustrate

how our approach works. Given  $M$  points  $(x_0, y_0), \dots, (x_{M-1}, y_{M-1})$ , the standard optimization goes like this. We first *parameterize* the line, say  $Ax + By + C = 0$ , the parameter being  $\theta = (A, B, C)^\top$ . The distance  $d_\kappa$  of point  $(x_\kappa, y_\kappa)$  from this line is

$$d_\kappa = \frac{|Ax_\kappa + By_\kappa + C|}{\sqrt{A^2 + B^2}}. \quad (4)$$

We minimize the sum of square distances:

$$E(A, B, C) = \sum_{\kappa=0}^{M-1} \frac{(Ax_\kappa + By_\kappa + C)^2}{A^2 + B^2}. \quad (5)$$

The solution is analytically obtained by solving a  $2 \times 2$  eigenvalue problem.

Our approach is different. We represent the point  $(x_\kappa, y_\kappa)$  by the 3-D vector

$$\mathbf{x}_\kappa = \begin{pmatrix} x_\kappa/f_0 \\ y_\kappa/f_0 \\ 1 \end{pmatrix}, \quad (6)$$

where  $f_0$  is a scale normalization constant of approximately the image size<sup>†</sup>. Three points  $\mathbf{x}_\kappa, \mathbf{x}_\lambda$ , and  $\mathbf{x}_\mu$  are collinear if and only if their scalar triple product  $|\mathbf{x}_\kappa, \mathbf{x}_\lambda, \mathbf{x}_\mu|$  is zero. Hence, the consistency (or *collinearity* in this case) constraint is given by

$$|\mathbf{x}_\kappa, \mathbf{x}_{\kappa+1}, \mathbf{x}_{\kappa+2}| = 0, \quad (7)$$

for  $\kappa = 0, \dots, M-3$ . These  $M-2$  equations define an  $(M+2)$ -D manifold  $\mathcal{S}$  in the  $2M$ -D joint space of  $(x_0, y_0, \dots, x_{M-1}, y_{M-1})$ . Here, consecutive triplets of points are considered, but the order is irrelevant as long as the  $M$  points are constrained to be collinear. The dimension of  $\mathcal{S}$  corresponds to the two degrees of freedom of the line to be fitted and the positions of the  $M$  points on it. We now orthogonally project the observation  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$ , which does not necessarily satisfy Eq. (7), onto  $\mathcal{S}$ . Note that we neither introduce any *parameterization* to the line to be fitted nor define any cost function to minimize.

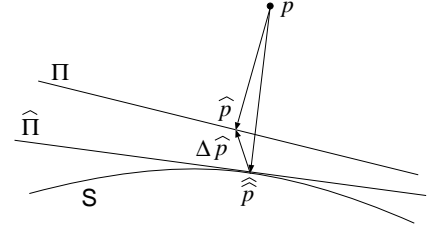
Let  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  be the desired projection, and let  $\hat{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \Delta\mathbf{x}_\kappa$ . Substituting  $\hat{\mathbf{x}}_\kappa$  into Eq. (7) and expanding it to a first order in  $\Delta\mathbf{x}_\kappa$ , we obtain

$$\begin{aligned} &(\Delta\mathbf{x}_\kappa, \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2}) + (\Delta\mathbf{x}_{\kappa+1}, \mathbf{x}_{\kappa+2} \times \mathbf{x}_\kappa) \\ &+ (\Delta\mathbf{x}_{\kappa+2}, \mathbf{x}_\kappa \times \mathbf{x}_{\kappa+1}) = |\mathbf{x}_\kappa, \mathbf{x}_{\kappa+1}, \mathbf{x}_{\kappa+2}|, \end{aligned} \quad (8)$$

where and throughout this paper we denote by  $(\mathbf{a}, \mathbf{b})$  the inner product of  $\mathbf{a}$  and  $\mathbf{b}$ . Since Eq. (8) is linear in  $\Delta\mathbf{x}_\kappa$ , the  $M-2$  equations in this form, when viewed as equations of free variables  $\hat{\mathbf{x}}_\kappa$  through  $\Delta\mathbf{x}_\kappa = \mathbf{x}_\kappa - \hat{\mathbf{x}}_\kappa$ , define a plane<sup>††</sup>,  $\Pi$  that approximates the manifold  $\mathcal{S}$ . We now compute a projection direction  $\Delta\mathbf{x}_0 \oplus \dots \oplus \Delta\mathbf{x}_{M-1}$  orthogonal to  $\Pi$ . The direction of this projection is given by

$$\begin{aligned} \Delta\mathbf{x}_\kappa = &\lambda_\kappa \mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2}) + \lambda_{\kappa-1} \mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa-1}) \\ &+ \lambda_{\kappa-2} \mathbf{P}_\kappa(\mathbf{x}_{\kappa-2} \times \mathbf{x}_{\kappa-1}), \end{aligned} \quad (9)$$

where  $\lambda_\kappa$  are unknown parameters and  $\mathbf{P}_\kappa \equiv$



**Fig. 2** Successive orthogonal projection onto  $\mathcal{S}$ . The orthogonal projection from  $p$  to  $\Pi$  minimizes  $\|p - \hat{p}\|^2$ , while the orthogonal projection from  $p$  to  $\hat{\Pi}$  minimizes  $\|p - \hat{p}\|^2 = \|p - \hat{p} + \Delta\hat{p}\|^2$ , where  $\Delta\hat{p} = \Delta\hat{\mathbf{x}}_0 \oplus \dots \oplus \Delta\hat{\mathbf{x}}_{M-1}$ .

$\text{diag}(1, 1, 0)$  and (see Appendix A.1 for the derivation). We adjust the parameters  $\lambda_\kappa$  so that the projection reaches the plane  $\Pi$ . Substitution of Eq. (9) into Eq. (8) results in simultaneous linear equations in  $\lambda_\kappa$  in the form

$$A_\kappa \lambda_{\kappa-2} + B_\kappa \lambda_{\kappa-1} + C_\kappa \lambda_\kappa + D_\kappa \lambda_{\kappa+1} + E_\kappa \lambda_{\kappa+2} = F_\kappa, \quad (10)$$

where

$$\begin{aligned} A_\kappa &= (\mathbf{P}_\kappa(\mathbf{x}_{\kappa-2} \times \mathbf{x}_{\kappa-1}), \mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2})), \\ B_\kappa &= (\mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa-1}), \mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2})) \\ &\quad + (\mathbf{P}_\kappa(\mathbf{x}_{\kappa-1} \times \mathbf{x}_\kappa), \mathbf{P}_\kappa(\mathbf{x}_{\kappa+2} \times \mathbf{x}_\kappa)), \\ C_\kappa &= \|\mathbf{P}_\kappa(\mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2})\|^2 + \|\mathbf{P}_\kappa(\mathbf{x}_{\kappa+2} \times \mathbf{x}_\kappa)\|^2 \\ &\quad + \|\mathbf{P}_\kappa(\mathbf{x}_\kappa \times \mathbf{x}_{\kappa+1})\|^2, \\ D_\kappa &= (\mathbf{P}_\kappa(\mathbf{x}_{\kappa+2} \times \mathbf{x}_{\kappa+3}), \mathbf{P}_\kappa(\mathbf{x}_{\kappa+2} \times \mathbf{x}_\kappa)) \\ &\quad + (\mathbf{P}_\kappa(\mathbf{x}_{\kappa+3} \times \mathbf{x}_{\kappa+1}), \mathbf{P}_\kappa(\mathbf{x}_\kappa \times \mathbf{x}_{\kappa+1})), \\ E_\kappa &= (\mathbf{P}_\kappa(\mathbf{x}_{\kappa+3} \times \mathbf{x}_{\kappa+4}), \mathbf{P}_\kappa(\mathbf{x}_\kappa \times \mathbf{x}_{\kappa+1})), \\ F_\kappa &= |\mathbf{x}_\kappa, \mathbf{x}_{\kappa+1}, \mathbf{x}_{\kappa+2}|. \end{aligned} \quad (11)$$

Solving Eq. (10) and substituting the resulting  $\lambda_\kappa$  into Eq. (9), we can determine  $\hat{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \Delta\mathbf{x}_\kappa$ . The resulting projection  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  is orthogonal to the plane  $\Pi$  by construction, but not necessarily  $\hat{p} \in \mathcal{S}$  or the projection is orthogonal to  $\mathcal{S}$ . So, we correct  $\hat{p}$  to  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  so that  $\hat{p} \in \mathcal{S}$  and the projection is orthogonal to  $\mathcal{S}$ . Letting  $\hat{\mathbf{x}}_\kappa = \hat{\mathbf{x}}_\kappa - \Delta\hat{\mathbf{x}}_\kappa$ , we substitute  $\hat{\mathbf{x}}_\kappa$  for  $\mathbf{x}_\kappa$  in Eq. (7). The first order expansion at  $\hat{\mathbf{x}}_\kappa$  in the higher order term  $\Delta\hat{\mathbf{x}}_\kappa$  is

$$\begin{aligned} &(\Delta\hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) + (\Delta\hat{\mathbf{x}}_{\kappa+1}, \hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_\kappa) \\ &+ (\Delta\hat{\mathbf{x}}_{\kappa+2}, \hat{\mathbf{x}}_\kappa \times \hat{\mathbf{x}}_{\kappa+1}) = |\hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1}, \hat{\mathbf{x}}_{\kappa+2}|. \end{aligned} \quad (12)$$

The  $M-2$  equations in this form define a plane  $\hat{\Pi}$  that approximates  $\mathcal{S}$  better than  $\Pi$ . We compute a new orthogonal projection to it. Note that the projection always starts from the observation  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$ , not from  $\hat{p}$  (Fig. 2). The new direction of this projection is given by

$$\begin{aligned} \Delta\hat{\mathbf{x}}_\kappa = &\lambda_\kappa \mathbf{P}_\kappa(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) + \lambda_{\kappa-1} \mathbf{P}_\kappa(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa-1}) \\ &+ \lambda_{\kappa-2} \mathbf{P}_\kappa(\hat{\mathbf{x}}_{\kappa-2} \times \hat{\mathbf{x}}_{\kappa-1}) - \hat{\mathbf{x}}_\kappa, \end{aligned} \quad (13)$$

where we define

<sup>†</sup>This is merely for making the three components to have the same order of magnitude for numerical stability. Theoretically, it can be set to any value, e.g., 1.

<sup>††</sup>Strictly speaking, this is an  $(M+2)$ -D *affine space* but we call this a “plane” for short.

$$\tilde{\mathbf{x}}_{\kappa} = \mathbf{x}_{\kappa} - \hat{\mathbf{x}}_{\kappa}. \quad (14)$$

(See Appendix A.2 for the derivation.) The parameters  $\lambda_{\kappa}$  are determined so that that the projection reaches the plane  $\hat{\Pi}$ . Substitution of Eq. (13) into Eq. (12) results in simultaneous linear equations in  $\lambda_{\kappa}$  in the same form as Eq. (10) except that  $A_{\kappa}, \dots, F_{\kappa}$  are now replaced by

$$\begin{aligned} A_{\kappa} &= (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa-2} \times \hat{\mathbf{x}}_{\kappa-1}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2})), \\ B_{\kappa} &= (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa-1}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2})) \\ &\quad + (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa-1} \times \hat{\mathbf{x}}_{\kappa}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_{\kappa})), \\ C_{\kappa} &= \|\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2})\|^2 + \|\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_{\kappa})\|^2 \\ &\quad + \|\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa} \times \hat{\mathbf{x}}_{\kappa+1})\|^2, \\ D_{\kappa} &= (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_{\kappa+3}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_{\kappa})) \\ &\quad + (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+3} \times \hat{\mathbf{x}}_{\kappa+1}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa} \times \hat{\mathbf{x}}_{\kappa+1})), \\ E_{\kappa} &= (\mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+3} \times \hat{\mathbf{x}}_{\kappa+4}), \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa} \times \hat{\mathbf{x}}_{\kappa+1})), \\ F_{\kappa} &= |\hat{\mathbf{x}}_{\kappa}, \hat{\mathbf{x}}_{\kappa+1}, \hat{\mathbf{x}}_{\kappa+2}| + |\tilde{\mathbf{x}}_{\kappa}, \tilde{\mathbf{x}}_{\kappa+1}, \tilde{\mathbf{x}}_{\kappa+2}| \\ &\quad + |\hat{\mathbf{x}}_{\kappa}, \tilde{\mathbf{x}}_{\kappa+1}, \tilde{\mathbf{x}}_{\kappa+2}| + |\tilde{\mathbf{x}}_{\kappa}, \hat{\mathbf{x}}_{\kappa+1}, \tilde{\mathbf{x}}_{\kappa+2}|. \end{aligned} \quad (15)$$

Solving Eq. (10) and substituting the resulting  $\lambda_{\kappa}$  in Eq. (13), we can determine  $\hat{\mathbf{x}}_{\kappa} = \mathbf{x}_{\kappa} - \Delta\hat{\mathbf{x}}_{\kappa}$ . The resulting projection is orthogonal to  $\hat{\Pi}$  by construction but still not strictly on  $\mathcal{S}$  or orthogonal to it. So, we let  $\hat{\mathbf{x}}_{\kappa} \leftarrow \hat{\mathbf{x}}_{\kappa}$  and repeat this correction. In the end,  $\Delta\hat{\mathbf{x}}_{\kappa} = \mathbf{0}$  and  $\hat{p} \in \mathcal{S}$ . Because the plane  $\hat{\Pi}$  is a first order expansion of  $\mathcal{S}$  at  $\hat{p} \in \mathcal{S}$ , it is *tangent* to  $\mathcal{S}$  at  $\hat{p}$ . Since  $p\hat{p}$  is orthogonal to  $\hat{\Pi}$ , it is orthogonal to  $\mathcal{S}$  itself. Our algorithm is summarized as follows:

**Input:** Observed positions  $\mathbf{x}_{\kappa}, \kappa = 0, \dots, M-1$ .  
**Output:** Corrected positions  $\hat{\mathbf{x}}_{\kappa}, \kappa = 0, \dots, M-1$ , and the reprojection error  $E$ .

**Procedure:**

1. Let  $E_0 = \infty, \hat{\mathbf{x}}_{\kappa} = \mathbf{x}_{\kappa}, \tilde{\mathbf{x}}_{\kappa} = \mathbf{0}, \kappa = 0, \dots, M-1$ , where  $\infty$  is a sufficiently large number.
2. Compute  $A_{\kappa}, \dots, F_{\kappa}, \kappa = 0, \dots, M-3$ , by Eqs. (14).
3. Solve the following set of linear equations in  $\lambda_{\kappa}, \kappa = 0, \dots, M-3$ :

$$\begin{aligned} & \left( \begin{array}{ccccccc} C_0 & D_0 & E_0 & & & & \\ B_1 & C_1 & D_1 & E_1 & & & \\ A_2 & B_2 & C_2 & D_2 & E_2 & & \\ & A_3 & B_3 & C_3 & D_3 & E_3 & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & & A_{M-3} & B_{M-3} & C_{M-3} \end{array} \right) \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_{M-3} \end{pmatrix} \\ &= \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ \vdots \\ F_{M-3} \end{pmatrix}. \end{aligned} \quad (16)$$

4. Update  $\tilde{\mathbf{x}}_{\kappa}$  and  $\hat{\mathbf{x}}_{\kappa}, \kappa = 0, \dots, M-1$ , as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{\kappa} & \leftarrow \lambda_{\kappa} \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) \\ & \quad + \lambda_{\kappa-1} \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa-1}) \\ & \quad + \lambda_{\kappa-2} \mathbf{P}_{\mathbf{k}}(\hat{\mathbf{x}}_{\kappa-2} \times \hat{\mathbf{x}}_{\kappa-1}), \\ \hat{\mathbf{x}}_{\kappa} & \leftarrow \mathbf{x}_{\kappa} - \tilde{\mathbf{x}}_{\kappa}. \end{aligned} \quad (17)$$

5. Compute the reprojection error  $E$  as follows:

$$E = \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_{\kappa}\|^2. \quad (18)$$

6. If  $|E - E_0| \approx 0$ , return  $E$  and  $\hat{\mathbf{x}}_{\kappa}, \kappa = 0, \dots, M-1$ , and stop. Else, let  $E_0 \leftarrow E$ , and go back to Step 2.

Once we have imposed the consistency constraint, we can pick out any two (distinct) points and compute the line connecting them.

## 6. Triangulation from Multiple Views

The above line fitting algorithm is only for *illustration* of our approach. In practice, as pointed out earlier, it is much simpler to minimize Eq. (5) by solving a  $2 \times 2$  eigenvalue problem; the solution is identical to the one obtained by the above procedure. Using the same principle, however, we can derive a new algorithm for triangulation from multiple views.

### 6.1 Problem

Suppose we have  $M$  cameras. We assume that their intrinsic and extrinsic parameters are known. Let  $(x_{\kappa}, y_{\kappa})$  be the image of a 3-D point  $(X, Y, Z)$  in the  $\kappa$ th view. Our task is to compute  $(X, Y, Z)$  from  $(x_{\kappa}, y_{\kappa}), \kappa = 0, \dots, M-1$ . If  $(x_{\kappa}, y_{\kappa})$  are exact, we can pick out any two (nondegenerate) views and compute  $(X, Y, Z)$  by elementary triangulation. If  $(x_{\kappa}, y_{\kappa})$  are not exact, however, the rays, call them the *lines of sight*, starting from the projection centers and passing through the points in the images, do not meet at a single point in the scene. Traditionally, this has been dealt with by the standard optimization [4]: we search the 3-D space for  $\boldsymbol{\theta} = (X, Y, Z)^T$  that minimizes the reprojection error of Eq. (1). Our approach is as follows. As line fitting, we represent each point  $(x_{\kappa}, y_{\kappa})$  by the 3-D vector in Eq. (6). The consistency constraint we adopt is

$$\epsilon_{ljp} \epsilon_{mkq} T_{(\kappa)i}^{lm} x_{(\kappa)}^i x_{(\kappa+1)}^j x_{(\kappa+2)}^k = 0, \quad (19)$$

for  $\kappa = 0, \dots, M-3$ , where  $T_{(\kappa)i}^{lm}$  is the trifocal tensor for the  $\kappa$ th, the  $(\kappa+1)$ st, and the  $(\kappa+2)$ nd views;  $\epsilon_{ijk}$  is the permutation symbol. We use Einstein's convention for omitting the summation symbol over repeated upper and lower indices. Equation (19) is known as the *trilinear constraint*, which provides a necessary and sufficient condition that the lines of sight from the three cameras meet at a single point in the scene [7]. The  $M-2$  equations in Eq. (19) guarantee that the  $M$  lines of sight have a common intersection. Here, consecutive triplets of frames are considered, but as in the case of

line fitting the order is irrelevant as long as the  $M$  lines of sight are constrained to meet at a point.

The  $M - 2$  equations in Eq. (19) define a manifold  $\mathcal{S}$  in the joint  $2M$ -D space of  $(x_0, y_0, \dots, x_{M-1}, y_{M-1})$ . Our task is to orthogonally project the observation  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$  onto  $\mathcal{S}$ . Let  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  be the desired projection, and let  $\hat{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \Delta\mathbf{x}_\kappa$ . Replacing  $\mathbf{x}_\kappa$  in Eq. (19) by  $\hat{\mathbf{x}}_\kappa$  and expanding it to a first order in  $\Delta\mathbf{x}_\kappa$ , we obtain (the summation symbol is omitted by Einstein's convention)

$$\begin{aligned} & \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( \Delta x_\kappa^i x_{\kappa+1}^j x_{\kappa+2}^k + x_\kappa^i \Delta x_{\kappa+1}^j x_{\kappa+2}^k \right) \\ & + x_\kappa^i x_{\kappa+1}^j \Delta x_{\kappa+2}^k = \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} x_\kappa^i x_{\kappa+1}^j x_{\kappa+2}^k. \end{aligned} \quad (20)$$

The resulting  $9(M - 2)$  equations for  $p, q = 1, 2, 3$  and  $\kappa = 0, \dots, M - 3$ , when viewed as equations of free variables  $\hat{\mathbf{x}}_\kappa$  through  $\Delta\mathbf{x}_\kappa = \mathbf{x}_\kappa - \hat{\mathbf{x}}_\kappa$ , define a set  $\Pi$  of  $9(M - 2)$  hyperplanes in  $2M$ -D that approximate  $\mathcal{S}$ . They would intersect at a 3-D affine space if  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1} \in \mathcal{S}$ . Otherwise, they may not have a common intersection.

We now determine a projection direction  $\Delta\mathbf{x}_0 \oplus \dots \oplus \Delta\mathbf{x}_{M-1}$  orthogonal to *all* the hyperplanes in  $\Pi$ . Such a direction may not exist, but we proceed as if there is; the nonexistence condition emerges later. If there are such  $\Delta\mathbf{x}_\kappa$ , they should have form

$$\Delta\mathbf{x}_\kappa^s = P_{(\kappa)pq}^i \lambda_{(\kappa)}^{pq} + Q_{(\kappa)pq}^i \lambda_{(\kappa-1)}^{pq} + R_{(\kappa)pq}^i \lambda_{(\kappa-2)}^{pq}, \quad (21)$$

where  $\lambda_{(\kappa)}^{pq}$  are unknown parameters (Appendix A.3 for the derivation). Here, we define

$$\begin{aligned} P_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} P_{\mathbf{k}}^{si} x_{\kappa+1}^j x_{\kappa+2}^k, \\ Q_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa-1)i}^{lm} x_{\kappa-1}^i P_{\mathbf{k}}^{sj} x_{\kappa+1}^k, \\ R_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa-2)i}^{lm} x_{\kappa-2}^i x_{\kappa-1}^j P_{\mathbf{k}}^{sk}. \end{aligned} \quad (22)$$

The symbol  $P_{\mathbf{k}}^{ij}$  denotes the  $(ij)$  element of the matrix  $\mathbf{P}_{\mathbf{k}}$  ( $= \text{diag}(1, 1, 0)$ ). We adjust the parameters  $\lambda_{(\kappa)}^{pq}$  so that the projection  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  be on all the hyperplanes in  $\Pi$ . Such a solution may not exist, but we proceed as if there is. Substituting Eq. (21) into Eq. (20), we obtain

$$\begin{aligned} & A_{(\kappa)pqrs} \lambda_{(\kappa-2)}^{rs} + B_{(\kappa)pqrs} \lambda_{(\kappa-1)}^{rs} + C_{(\kappa)pqrs} \lambda_{(\kappa)}^{rs} \\ & + D_{(\kappa)pqrs} \lambda_{(\kappa+1)}^{rs} + E_{(\kappa)pqrs} \lambda_{(\kappa+2)}^{rs} = F_{(\kappa)pq}, \end{aligned} \quad (23)$$

where we define

$$\begin{aligned} A_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} R_{(\kappa)rs}^i x_{\kappa+1}^j x_{\kappa+2}^k, \\ B_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( Q_{(\kappa)rs}^i x_{\kappa+1}^j x_{\kappa+2}^k \right. \\ & \quad \left. + x_\kappa^i R_{(\kappa+1)rs}^j x_{\kappa+2}^k \right), \\ C_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( P_{(\kappa)rs}^i x_{\kappa+1}^j x_{\kappa+2}^k \right. \\ & \quad + x_\kappa^i R_{(\kappa+1)rs}^j x_{\kappa+2}^k + x_\kappa^i Q_{(\kappa+1)rs}^j x_{\kappa+2}^k \\ & \quad \left. + x_\kappa^i R_{(\kappa+1)rs}^j x_{\kappa+2}^k + x_\kappa^i x_{\kappa+1}^j R_{(\kappa+2)rs}^k \right), \\ D_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( x_\kappa^i P_{(\kappa+1)rs}^j x_{\kappa+2}^k \right. \\ & \quad \left. + x_\kappa^i R_{(\kappa+1)rs}^j x_{\kappa+2}^k + x_\kappa^i x_{\kappa+1}^j Q_{(\kappa+2)rs}^k \right), \end{aligned}$$

$$\begin{aligned} E_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} x_\kappa^i x_{\kappa+1}^j P_{(\kappa+2)rs}^k, \\ F_{(\kappa)pq} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} x_\kappa^i x_{\kappa+1}^j x_{\kappa+2}^k. \end{aligned} \quad (24)$$

Equation (23) provides  $9(M - 2)$  linear equations ( $r, s = 1, 2, 3, \kappa = 0, \dots, M - 3$ ) in the  $9(M - 2)$  unknowns  $\lambda_{(\kappa)}^{pq}$  ( $p, q = 1, 2, 3, \kappa = 0, \dots, M - 3$ ). However, Eq. (23) is derived on the assumption that there is a solution, but it has a unique solution only when  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1} \in \mathcal{S}$ ; the solvability condition is gradually violated as  $p$  departs from  $\mathcal{S}$ . We discuss how to cope with this shortly. Once  $\lambda_{(\kappa)}^{pq}$  are obtained, we can determine  $\Delta\mathbf{x}_\kappa$  by Eq. (21) and compute  $\hat{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \Delta\mathbf{x}_\kappa$ .

Now, we go the second round. Replacing  $\mathbf{x}_\kappa$  in Eq. (19) by  $\hat{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \Delta\mathbf{x}_\kappa$  and expanding it to a first order in  $\Delta\hat{\mathbf{x}}_\kappa$ , we obtain

$$\begin{aligned} & \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( \Delta\hat{x}_\kappa^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k + \hat{x}_\kappa^i \Delta\hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k \right. \\ & \quad \left. + \hat{x}_\kappa^i \hat{x}_{\kappa+1}^j \Delta\hat{x}_{\kappa+2}^k \right) = \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \hat{x}_\kappa^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k, \end{aligned} \quad (25)$$

which defines a set  $\hat{\Pi}$  of hyperplanes that should approximate  $\mathcal{S}$  better than those in  $\Pi$ , since  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  is expected to be closer to  $\mathcal{S}$  than  $p = \mathbf{x}_0 \oplus \dots \oplus \mathbf{x}_{M-1}$ . The hyperplanes in  $\hat{\Pi}$  would intersect at a 3-D affine space if  $\hat{p} \in \mathcal{S}$ . We determine a new projection direction  $\Delta\hat{\mathbf{x}}_0 \oplus \dots \oplus \Delta\hat{\mathbf{x}}_{M-1}$  orthogonal to all the hyperplanes in  $\hat{\Pi}$ , although such a direction may not exist. If there are such  $\Delta\hat{\mathbf{x}}_\kappa$ , they should have the form

$$\begin{aligned} \Delta\hat{\mathbf{x}}_\kappa^s &= \sum_{p,q=1}^3 \hat{P}_{(\kappa)pq}^s \lambda_{(\kappa)}^{pq} + \sum_{p,q=1}^3 \hat{Q}_{(\kappa)pq}^s \lambda_{(\kappa-1)}^{pq} \\ & \quad + \sum_{p,q=1}^3 \hat{R}_{(\kappa)pq}^s \lambda_{(\kappa-2)}^{pq} - \tilde{x}_\kappa^i, \end{aligned} \quad (26)$$

where we define

$$\tilde{\mathbf{x}}_\kappa = \mathbf{x}_\kappa - \hat{\mathbf{x}}_\kappa, \quad (27)$$

and

$$\begin{aligned} \hat{P}_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} P_{\mathbf{k}}^{si} \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k, \\ \hat{Q}_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa-1)i}^{lm} \hat{x}_{\kappa-1}^i P_{\mathbf{k}}^{sj} \hat{x}_{\kappa+1}^k, \\ \hat{R}_{(\kappa)pq}^s &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa-2)i}^{lm} \hat{x}_{\kappa-2}^i \hat{x}_{\kappa-1}^j P_{\mathbf{k}}^{sk}. \end{aligned} \quad (28)$$

(See Appendix A.4 for the derivation.) We adjust the parameters  $\lambda_{(\kappa)}^{pq}$  so that the resulting projection be on all the hyperplanes in  $\hat{\Pi}$ , although such a solution may not exist. Substituting Eq. (26) into Eq. (25), we obtain simultaneous linear equations in  $\lambda_{(\kappa)}^{pq}$  in the same form as Eq. (23) except that  $A_{(\kappa)pqrs}, \dots, F_{(\kappa)pq}$  are now replaced by

$$\begin{aligned} A_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \hat{R}_{(\kappa)rs}^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k, \\ B_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( \hat{Q}_{(\kappa)rs}^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k \right. \\ & \quad \left. + \hat{x}_\kappa^i \hat{R}_{(\kappa+1)rs}^j \hat{x}_{\kappa+2}^k \right), \\ C_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm} \left( \hat{P}_{(\kappa)rs}^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k \right. \\ & \quad \left. + \hat{x}_\kappa^i \hat{Q}_{(\kappa+1)rs}^j \hat{x}_{\kappa+2}^k + \hat{x}_\kappa^i \hat{x}_{\kappa+1}^j \hat{R}_{(\kappa+2)rs}^k \right), \end{aligned}$$

$$\begin{aligned}
D_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm}\left(\hat{x}_{\kappa}^i\hat{P}_{(\kappa+1)rs}^j\hat{x}_{\kappa+2}^k\right. \\
&\quad \left. + \hat{x}_{\kappa}^i\hat{x}_{\kappa+1}^j\hat{Q}_{(\kappa+2)rs}^k\right), \\
E_{(\kappa)pqrs} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm}\hat{x}_{\kappa}^i\hat{x}_{\kappa+1}^j\hat{P}_{(\kappa+2)rs}^k, \\
F_{(\kappa)pq} &= \epsilon_{ljp}\epsilon_{mkq}T_{(\kappa)i}^{lm}\left(\hat{x}_{\kappa}^i\hat{x}_{\kappa+1}^j\hat{x}_{\kappa+2}^k + \tilde{x}_{\kappa}^i\hat{x}_{\kappa+1}^j\hat{x}_{\kappa+2}^k\right. \\
&\quad \left. + \hat{x}_{\kappa}^i\tilde{x}_{\kappa+1}^j\hat{x}_{\kappa+2}^k + \hat{x}_{\kappa}^i\hat{x}_{\kappa+1}^j\tilde{x}_{\kappa+2}^k\right). \tag{29}
\end{aligned}$$

Solving Eqs. (23) and substituting the resulting  $\lambda_{(\kappa)}^{pq}$  in Eq. (26), we can determine  $\Delta\hat{\mathbf{x}}_{\kappa}$  and compute  $\hat{\mathbf{x}}_{\kappa} = \hat{\mathbf{x}}_{\kappa} - \Delta\hat{\mathbf{x}}_{\kappa}$ . Letting  $\hat{\mathbf{x}}_{\kappa} \leftarrow \hat{\mathbf{x}}_{\kappa}$ , we repeat this procedure. In the end,  $\Delta\hat{\mathbf{x}}_{\kappa} = \mathbf{0}$  and  $\hat{p} = \hat{p} \in \mathcal{S}$ . Because the hyperplanes in  $\hat{\Pi}$  are first order expansions of  $\mathcal{S}$  at  $\hat{p} \in \mathcal{S}$ , they are *tangent* to  $\mathcal{S}$  at  $\hat{p}$ , defining the tangent space  $T_{\hat{p}}(\mathcal{S})$  at  $\hat{p}$  as their intersection. Since  $p\hat{p}$  is now orthogonal to all the hyperplanes in  $\hat{\Pi}$ , it is orthogonal to  $\mathcal{S}$  itself.

## 6.2 Rank deficiency

There is one added complexity due to the fact that Eq. (19) has *redundancies*: each trilinear constraint consists of nine equalities for  $p, q = 1, 2, 3$ , and only four are *linearly* independent due to the skew properties of  $\epsilon_{ijk}$ . Hence, Eq. (19) consists of  $9(M-2)$  equalities,  $4(M-2)$  of which are linearly independent. Among them, however, only  $2M-3$  are *algebraically* independent. This is because *the manifold  $\mathcal{S}$  should be homeomorphic to  $\mathcal{R}^3$* , since a point in  $\mathcal{S}$  is in one-to-one correspondence to a point to be reconstructed in  $\mathcal{R}^3$  as the unique intersection of the lines of sight. Thus,  $\mathcal{S}$  is 3-D and can be defined as an intersection of  $2M-3$  hypersurfaces; *the remaining hypersurfaces automatically pass through it*.

We enumerate the index pairs  $(p, q) = (1,1), (1,2), \dots, (3,3)$  with a serial number  $\alpha = 1, \dots, 9$  and  $(r, s) = (1,1), (1,2), \dots, (3,3)$  with  $\beta = 1, \dots, 9$ , and identify  $\hat{A}_{(\kappa)pqrs}, \hat{B}_{(\kappa)pqrs}$ , etc. with  $9 \times 9$  matrices. Likewise,  $\hat{F}_{(\kappa)pq}$  and  $\lambda_{(\kappa)}^{pq}$  are identified with 9-D vectors  $\hat{\mathbf{f}}_{(\kappa)}$  and  $\boldsymbol{\lambda}_{(\kappa)}$ , respectively. Then, Eq. (23) now takes the form

$$\begin{aligned}
&\begin{pmatrix} \mathbf{C}_0 & \mathbf{D}_0 & \mathbf{E}_0 & & & \\ \mathbf{B}_1 & \mathbf{C}_1 & \mathbf{D}_1 & \mathbf{E}_1 & & \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & \mathbf{D}_2 & \mathbf{E}_2 & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{A}_{M-3} & \mathbf{B}_{M-3} & \mathbf{C}_{M-3} \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{M-3} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{M-3} \end{pmatrix}, \tag{30}
\end{aligned}$$

where  $\mathbf{A}_{\kappa}, \dots, \mathbf{E}_{\kappa}$  are  $9 \times 9$  matrices, and  $\boldsymbol{\lambda}_{\kappa}$  and  $\mathbf{f}_{\kappa}$  are 9-D vectors. The coefficient matrix has a band of width 45. Equation (30) does not have a unique solution unless  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1} \in \mathcal{S}$ . The  $9(M-2) \times 9(M-2)$  coefficient matrix generally has rank  $2M$ , because the underlying unknowns are  $2M$  variables  $\Delta x_0, \Delta y_0, \dots, \Delta x_{M-1}, \Delta y_{M-1}$ . However, the rank drops to  $2M-3$  at

the moment the projection  $\hat{p} = \hat{\mathbf{x}}_0 \oplus \dots \oplus \hat{\mathbf{x}}_{M-1}$  reaches  $\mathcal{S}$ , which is our goal. Hence, we can select appropriate  $2M-3$  equations from among the  $9(M-2)$  equations in Eq. (30), which is mathematically equivalent to using the pseudoinverse of rank  $2M-3$ . Our algorithm is summarized as follows:

**Input:** Observed positions  $\mathbf{x}_{\kappa}$ ,  $\kappa = 0, \dots, M-1$ , and the trifocal tensors  $T_{(\kappa)i}^{jk}$ ,  $\kappa = 0, \dots, M-2$ .

**Output:** Corrected positions  $\hat{\mathbf{x}}_{\kappa}$ ,  $\kappa = 0, \dots, M-1$ , and the reprojection error  $E$ .

## Procedure:

1. Let  $E_0 = \infty$ ,  $\hat{\mathbf{x}}_{\kappa} = \mathbf{x}_{\kappa}$ ,  $\tilde{\mathbf{x}}_{\kappa} = \mathbf{0}$ ,  $\kappa = 0, \dots, M-1$ , where  $\infty$  is a sufficiently large number.
2. Compute  $\hat{P}_{pq}^s, \hat{Q}_{pq}^s$ , and  $\hat{R}_{pq}^s$  in Eqs. (28).
3. Compute  $A_{pqrs}, \dots, F_{pq}$  in Eqs. (29).
4. Solve Eq. (30) for  $\lambda_{(\kappa)}^{pq}$ , using pseudoinverse of rank  $2M-3$ .
5. Update  $\tilde{\mathbf{x}}_{\kappa}$  and  $\hat{\mathbf{x}}_{\kappa}$ ,  $\kappa = 0, \dots, M-1$ , as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}_{\kappa} &\leftarrow \sum_{p,q=1}^3 P_{(\kappa)pq}^i \lambda_{(\kappa)}^{pq} + \sum_{p,q=1}^3 Q_{(\kappa)pq}^i \lambda_{(\kappa-1)}^{pq} \\
&\quad + \sum_{p,q=1}^3 R_{(\kappa)pq}^i \lambda_{(\kappa-2o)}^{pq}, \\
\hat{\mathbf{x}}_{\kappa} &\leftarrow \mathbf{x}_{\kappa} - \tilde{\mathbf{x}}_{\kappa}. \tag{31}
\end{aligned}$$

6. Compute the reprojection error  $E$  as follows:

$$E = \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_{\kappa}\|^2. \tag{32}$$

7. If  $|E - E_0| \approx 0$ , return  $E$  and  $\hat{\mathbf{x}}_{\kappa}$ ,  $\kappa = 0, \dots, M-1$ , and stop. Else, let  $E_0 \leftarrow E$ , and go back to Step 2.

## 6.3 Efficient computation

In the above algorithm, we frequently encounter expressions in the form

$$T_{pq} = \epsilon_{ljp}\epsilon_{mkq}T_i^{lm}x^i y^j z^k, \tag{33}$$

where  $T_i^{lm}$  takes  $T_{i(\kappa)}^{lm}$ , and  $x^i, y^j$ , and  $z^k$  take, respectively, the  $i, j, k$  components of  $\mathbf{x}_{\kappa}, \hat{\mathbf{x}}_{\kappa}$ , or  $\tilde{\mathbf{x}}_{\kappa}$ . The right-hand side of Eq. (33) is a sum over  $i, j, k, l, m = 1, 2, 3$  (the summation symbol omitted), so we need to add  $3^5 = 243$  terms. These summations cost a considerable computation time. It can be significantly reduced if we note that Eq. (33) can be equivalently rewritten as

$$\begin{aligned}
T_{pq} &= x^i \left( T_i^{p\boxplus 1, q\boxplus 1} y^{p\boxplus 2} z^{q\boxplus 2} - T_i^{p\boxplus 2, q\boxplus 1} y^{p\boxplus 1} z^{q\boxplus 2} \right. \\
&\quad \left. - T_i^{p\boxplus 1, q\boxplus 2} y^{p\boxplus 2} z^{q\boxplus 1} + T_i^{p\boxplus 2, q\boxplus 2} y^{p\boxplus 1} z^{q\boxplus 1} \right), \tag{34}
\end{aligned}$$

where  $\boxplus$  denotes addition modulo 3. The right-hand side is a sum over  $i = 1, 2, 3$ , so we need to add only  $3 \times 4 = 12$  terms. This makes the computation about  $243/12 (\approx 20)$  times more efficient.

## 7. Experiment

### 7.1 Accuracy

We created synthetic images of a cylindrical grid surface viewed by cameras surrounding it. Fig. 3 shows some of them. The image size is  $1000 \times 1000$  pixels, and the focal length is  $f = 600$  pixels. Independent Gaussian noise of mean 0 and standard deviation  $\sigma$  pixels is added to the  $x$  and the  $y$  coordinates of each grid point, and our algorithm<sup>†</sup> is applied. We stopped when the update of the reprojection error  $E$  is less than  $10^{-6}$ .

The solid line in Fig. 4(a) shows the average reprojection error per point over 1000 trials for each  $\sigma$ . The dotted line shows the first order theoretical expectation  $(2M - 3)(\sigma/f_0)^2$  for maximum likelihood (ML); we can see that the ML solution is indeed computed by our method. As a comparison, the dashed line shows the result of least squares; we linearize the perspective projection equation of the 3-D position  $(X, Y, Z)$  onto  $(x, y)$  in each frame and solve the resulting set of simultaneous linear equations by least squares. This is computationally the simplest but is known to be not optimal.

The solid line in Fig. 4(b) shows the average RMS error per point of the reconstructed 3-D position. We see that although the reprojection error is not much different between least squares and our method, the 3-D reconstruction accuracy is markedly distinct between them.

Here, we are assuming that cameras are calibrated in advance. In practice, accurate camera calibration is crucial for triangulation from images. Camera calibration is usually done off line, using reference patterns, but it is also possible to compute the intrinsic and extrinsic camera parameters and the 3-D positions simultaneously; this problem is called *structure from motion* [7], and various algebraic schemes for *self-*

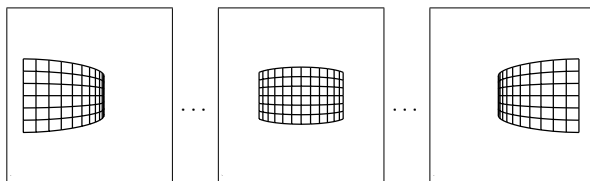


Fig. 3 Simulated images of a cylindrical grid surface.

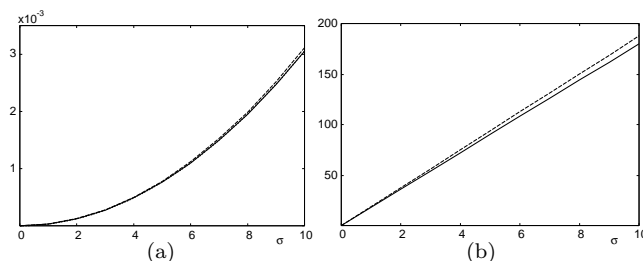


Fig. 4 (a) Average reprojection error. (b) RMS error of 3-D. Solid lines: our method. Dashed lines: least squares. Dotted lines: theoretical expectation  $(2M - 3)(\sigma/f_0)^2$ .

<sup>†</sup><http://www.iim.cs.tut.ac.jp/~sugaya/public.php>

*calibration* and numerical schemes for bundle adjustment have been proposed. We do not go into the details of this problem in this paper.

### 7.2 Computation time

Figure 5 shows the average computation time per point (average over 10 trials) for  $M = 3, 4, \dots, 31$  views with noise of  $\sigma = 5$  pixels. We used C++ with Intel Core2Duo E6850, 3.0GHz, using the efficient expression of Eq. (34). Most of the execution time is spent on the pseudoinverse computation<sup>†</sup> for solving Eq. (30). The dotted line is the curve  $aM^e$  fitted to the result; we found that the complexity is  $O(M^{3.22})$  as compared to  $O(M)$  for least squares. This should not be a problem for most applications, since usually feature points can be tracked only over a relatively small number of views.

We also used the tracking data provided by Oxford University<sup>††</sup> with 36 views (Fig. 6(a)), where 4983 points are tracked over 2 to 21 views<sup>†††</sup>. One view of the 3-D reconstruction is shown in Fig. 6(b). As a comparison, we tested the algorithm<sup>††††</sup> of Kahl et al. [9], and found that our reprojection error is slightly smaller than theirs for all points<sup>†††††</sup>. We surmise that this has

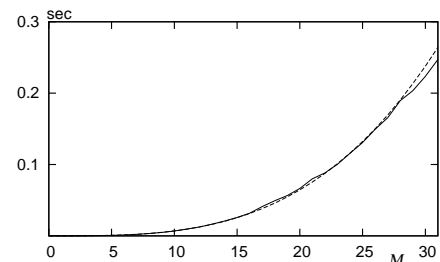


Fig. 5 Computation time (sec) vs. the number  $M$  of views. The dotted line  $aM^e$  is fitted with  $a = 4.17 \times 10^{-6}$  and  $e = 3.22$ .

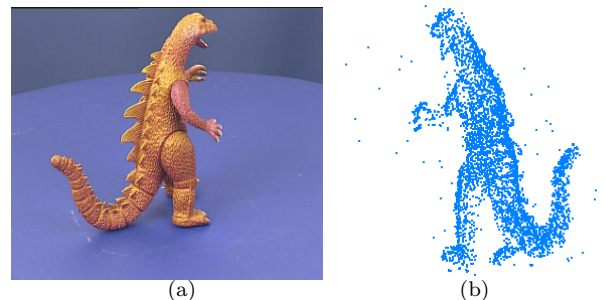


Fig. 6 (a) Feature point tracking. (b) Resulting 3-D reconstruction.

<sup>†</sup>We did not take the band structure into consideration; for a large  $M$ , further speedup will be possible by exploiting the sparseness.

<sup>††</sup><http://www.robots.ox.ac.uk/~vgg/data.html>

<sup>†††</sup>For 2 view correspondences, we used the method described in [12].

<sup>††††</sup><http://www.cs.washington.edu/homes/sagarwal/code.html>

<sup>†††††</sup>For most points, the reprojection error they reached

something to do with the iteration stopping criteria of the SeDuMi tool they used.

The total computation time of our method, including the preprocessing of the trifocal tensor computation and the post processing of 3-D reconstruction, is 2.22 sec, i.e., 0.000446 sec per point. The algorithm of Kahl et al. [9] took 5030 sec, i.e., 1.01 sec per point. Lu and Hartley [14] reported that their C++ program of branch and bound applied for a different data set took 0.02 sec per point. Fair and definitive comparison is difficult; existing methods all use complicated algorithms involving black box software tools and are difficult to implement from scratch. Also, the codes offered by the authors are written in different environments. Still, the above observations suggest that our algorithm is far faster than all existing standard optimization methods.

## 8. Conclusions

We presented an alternative approach to the standard optimization, which minimizes a cost function by searching a parameter space. We showed that our approach can lead to a new algorithm of multiview triangulation. While the standard optimization is generic in nature, applicable to any problem for which the cost function can be defined, our approach is limited only to those problems for which the consistency constraint can be defined in a tractable form. Another drawback of our approach is the dependence of efficiency on the number  $M$  of observations: Since we solve linear equations in the joint data space, the complexity is about  $O(M^3)$ , as we showed in Sect. 7.2, while it is  $O(M)$  for the standard optimization. However, this is not a problem depending applications. For example, our approach for multiview triangulation with a moderate  $M$  is far more efficient, as we demonstrated in the preceding section. On the other hand, our approach can avoid many difficulties encountered in the standard optimization.

- The standard optimization requires clever parameterization of the problem. Poor parameterization results in an intractable cost function which is difficult to minimize by any method. In contrast, our approach does not require any parameterization.
- The standard optimization requires a good initial value to start the search, which is often hard to guess. Global optimization can reach a solution independent of initial guesses but requires complicated analysis and a large amount of computation, often relying on various software tools. Our approach does not need any initial guess.

In theory, there can be pathological cases where our approach does not produce an exactly global optimum. This can arise when in the presence of extremely

---

was, when converted to pixel/frame, less than 1.005 times ours. Among the 4983 points, however, there were 11 whose reprojection error was more than that, the maximum ratio being 1.33.

large noise the joint input  $p$  is as far away from the manifold  $\mathcal{S}$  as compared to its radius of curvature; two points  $\hat{p}, \hat{p}' \in \mathcal{S}$  may exist such that  $p\hat{p}$  and  $p\hat{p}'$  are both orthogonal to  $\mathcal{S}$  and  $|p\hat{p}| \approx |p\hat{p}'|$ . Then, our projection may fall into either of them. Consider line fitting, for example. Suppose the input points are so disturbed by noise that they spread almost uniformly in a circular region. Then, two or more lines can fit almost equally well (or equally poorly, to be precise).

In an extremely noisy situation, however, the distinction between exactly optimal (in the sense of ML) and nearly optimal solutions does not make much sense; both are reasonable estimates in view of such noise, and accepting the solution produced by our approach seems a sensible choice. In the standard optimization, on the other hand, a local minimum can arise even in the absence of noise if the initial guess is bad. The cost  $E$  at a local minimum can be very high. In contrast, a non-optimal solution of our approach could result only from large noise (the input  $p$  being far apart from the consistency manifold  $\mathcal{S}$ ), and its reprojection error would be nearly the same as the optimal one.

From a theoretical viewpoint, however, it is desirable to obtain a criterion by analyzing the curvature of  $\mathcal{S}$  to give a noise threshold for guaranteeing exact optimality of orthogonal projection, in the same spirit as Hartley and Seo [6]. This remains as a future task.

**Acknowledgments.** The authors thank Fredrik Kahl and Martin Byröd of Lund University, Sweden, Fangfang Lu of Australian National University, and Yongduek Seo of Sogang University, Korea for various information. This work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, under a Grant in Aid for Scientific Research (C 21500172).

## References

- [1] S.J. Ahn, W. Rauh, H.S. Cho, and H.-J. Warecke, "Orthogonal distance fitting of implicit curves and surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, no.5, pp.620–638, May 2002.
- [2] A. Atieg and G.A. Watson, "A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances," *J. Comp. Appl. Math.*, vol.158, no.2, pp.277–296, Sept. 2003.
- [3] W. Föstner, "On weighting and choosing constraints for optimally reconstructing the geometry of image triplets," *Proc. 6th Euro. Conf. Computer Vision*, vol.2, pp.669–701, Dublin, Ireland, June/July 2000.
- [4] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," *Proc. 8th Asian Conf. Computer Vision*, vol.1, pp.13–34, Tokyo, Japan, Nov. 2007.
- [5] R. Hartley and F. Schaffalitzky, " $L_\infty$  minimization in geometric reconstruction problems," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, vol.1, pp.504–509, Washington DC, U.S.A., June/July 2004.
- [6] R. Hartley and Y. Seo, "Verifying global minima for  $L_2$  minimization problems," *Proc. IEEE Conf. Computer Vision Pattern Recog.*, Anchorage, AK, U.S.A., June 2008.
- [7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press,



Cambridge, U.K., 2004.

- [8] F. Kahl, "Multiple view geometry and the  $L_\infty$ -norm," Proc. 10th Int. Conf. Computer Vision, vol.2, pp.1002–1009, Beijing, China, Oct. 2005
- [9] F. Kahl, S. Agarwal, M.K. Chandraker, D. Kriegman, and S. Belongie, "Practical global optimization for multiview geometry," Int. J. Comput. Vis., vol.79, no.3, pp.271–284, Sept. 2008.
- [10] F. Kahl and D. Henrion, "Global optimal estimates for geometric reconstruction problems," Int. J. Comput. Vis., vol.74, no.1, pp.3–15, Aug. 2007.
- [11] K. Kanatani, Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier Science, Amsterdam, the Netherlands, 1996; Reprinted, Dover, New York, NY, U.S.A., 2005.
- [12] K. Kanatani, Y. Sugaya, and H. Niitsuma, "Triangulation from two views revisited: Hartley-Sturm vs. optimal correction," Proc. 19th British Machine Vision Conf., pp.173–182, Leeds, U.K., Sept. 2008.
- [13] Q. Ke and T. Kanade, "Quasiconvex optimization for robust geometric reconstruction," IEEE Trans. Patt. Anal. Mach. Intell., vol.29, no.10, pp.1834–1847, Oct. 2007.
- [14] F. Lu and R. Hartley, "A fast optimal algorithm for  $L_2$  triangulation," Proc. 8th Asian Conf. Computer Vision, vol.2, pp.279–288, Tokyo, Japan, Nov. 2007.
- [15] E.M. Mikhail and F. Ackermann, Observations and Least Squares, University Press of America, Lanham, MD., U.S.A., 1976.

### Appendix A: Derivation of Eq. (9)

The projection direction  $\Delta \mathbf{x}_0 \oplus \dots \oplus \Delta \mathbf{x}_{M-1}$  orthogonal to the  $(M+2)$ -D plane  $\Pi$  in 2M-D defined by Eq. (8) is computed by minimizing

$$E = \sum_{\kappa=0}^{M-1} \|\Delta \mathbf{x}_\kappa\|^2, \quad (\text{A}\cdot 1)$$

subject to Eq. (8). The third component of  $\mathbf{x}_\kappa$  is identically 1, so the third component of  $\Delta \mathbf{x}_\kappa$  is also identically 0. This constraint is written as

$$(\mathbf{k}, \Delta \mathbf{x}_\kappa) = 0, \quad (\text{A}\cdot 2)$$

where we define  $\mathbf{k} \equiv (0, 0, 1)^\top$ . Introducing the Lagrange multiplies to Eqs. (8) and (A·2), we let

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\Delta \mathbf{x}_\kappa\|^2 - \sum_{\kappa=0}^{M-1} \lambda_\kappa \left( (\Delta \mathbf{x}_\kappa, \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2}) \right. \\ & \left. + (\Delta \mathbf{x}_{\kappa+1}, \mathbf{x}_{\kappa+2} \times \mathbf{x}_\kappa) + (\Delta \mathbf{x}_{\kappa+2}, \mathbf{x}_\kappa \times \mathbf{x}_{\kappa+1}) \right) \\ & - \sum_{\kappa=0}^{M-1} \mu_\kappa (\mathbf{k}, \Delta \mathbf{x}_\kappa). \end{aligned} \quad (\text{A}\cdot 3)$$

This can be rewritten as

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\Delta \mathbf{x}_\kappa\|^2 - \sum_{\kappa=0}^{M-1} \lambda_\kappa (\Delta \mathbf{x}_\kappa, \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2}) \\ & - \sum_{\kappa=0}^{M-1} \lambda_{\kappa-1} (\Delta \mathbf{x}_\kappa, \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa-1}) \\ & - \sum_{\kappa=0}^{M-1} \lambda_{\kappa-2} (\Delta \mathbf{x}_\kappa, \mathbf{x}_{\kappa-2} \times \mathbf{x}_{\kappa-1}) \end{aligned}$$

$$- \sum_{\kappa=0}^{M-1} \mu_\kappa (\mathbf{k}, \Delta \mathbf{x}_\kappa), \quad (\text{A}\cdot 4)$$

where terms with subscript  $k$  outside the range of  $0, \dots, M-1$  are regarded as 0. Differentiating Eq. (A·4) with respect to  $\Delta \mathbf{x}_\kappa$  and setting the result to  $\mathbf{0}$ , we obtain

$$\begin{aligned} \Delta \mathbf{x}_\kappa &= \lambda_\kappa \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa+2} + \lambda_{\kappa-1} \mathbf{x}_{\kappa+1} \times \mathbf{x}_{\kappa-1} \\ & + \lambda_{\kappa-2} \mathbf{x}_{\kappa-2} \times \mathbf{x}_{\kappa-1} + \mu_\kappa \mathbf{k}. \end{aligned} \quad (\text{A}\cdot 5)$$

Multiplying on both sides  $\mathbf{P}_\mathbf{k} = \text{diag}(1, 1, 0)$ , which makes the third component 0, and noting that  $\mathbf{P}_\mathbf{k} \Delta \mathbf{x}_\kappa = \Delta \mathbf{x}_\kappa$  and  $\mathbf{P}_\mathbf{k} \mathbf{k} = \mathbf{0}$ , we obtain Eq. (9).

### Appendix B: Derivation of Eq. (13)

The projection direction  $\Delta \hat{\mathbf{x}}_0 \oplus \dots \oplus \Delta \hat{\mathbf{x}}_{M-1}$  orthogonal to the  $(M+2)$ -D plane  $\Pi$  in 2M-D defined by Eq. (12) is computed by minimizing

$$E = \sum_{\kappa=0}^{M-1} \|\mathbf{x}_\kappa - \hat{\mathbf{x}}_\kappa + \Delta \hat{\mathbf{x}}_\kappa\|^2 = \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_\kappa + \Delta \hat{\mathbf{x}}_\kappa\|^2. \quad (\text{A}\cdot 6)$$

The third component of  $\Delta \hat{\mathbf{x}}_\kappa$  should be 0, so we have the constraint

$$(\mathbf{k}, \Delta \hat{\mathbf{x}}_\kappa) = 0. \quad (\text{A}\cdot 7)$$

Introducing the Lagrange multiplies to Eqs. (12) and (A·7), we let

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_\kappa + \Delta \hat{\mathbf{x}}_\kappa\|^2 - \sum_{\kappa=0}^{M-1} \lambda_\kappa \left( (\Delta \hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) \right. \\ & \left. + (\Delta \hat{\mathbf{x}}_{\kappa+1}, \hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_\kappa) + (\Delta \hat{\mathbf{x}}_{\kappa+2}, \hat{\mathbf{x}}_\kappa \times \hat{\mathbf{x}}_{\kappa+1}) \right) \\ & - \sum_{\kappa=0}^{M-1} \mu_\kappa (\mathbf{k}, \Delta \hat{\mathbf{x}}_\kappa), \end{aligned} \quad (\text{A}\cdot 8)$$

which can be rewritten as

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_\kappa + \Delta \hat{\mathbf{x}}_\kappa\|^2 - \sum_{\kappa=0}^{M-1} \lambda_\kappa (\Delta \hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) \\ & - \sum_{\kappa=0}^{M-1} \lambda_\kappa (\Delta \hat{\mathbf{x}}_{\kappa+1}, \hat{\mathbf{x}}_{\kappa+2} \times \hat{\mathbf{x}}_\kappa) \\ & - \sum_{\kappa=0}^{M-1} \lambda_\kappa (\Delta \hat{\mathbf{x}}_{\kappa+2}, \hat{\mathbf{x}}_\kappa \times \hat{\mathbf{x}}_{\kappa+1}) - \sum_{\kappa=0}^{M-1} \mu_\kappa (\mathbf{k}, \Delta \hat{\mathbf{x}}_\kappa) \\ & = \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_\kappa + \Delta \hat{\mathbf{x}}_\kappa\|^2 - \sum_{\kappa=0}^{M-1} \lambda_\kappa (\Delta \hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2}) \\ & - \sum_{\kappa=0}^{M-1} \lambda_{\kappa-1} (\Delta \hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa-1}) \\ & - \sum_{\kappa=0}^{M-1} \lambda_{\kappa-2} (\Delta \hat{\mathbf{x}}_\kappa, \hat{\mathbf{x}}_{\kappa-2} \times \hat{\mathbf{x}}_{\kappa-1}) \end{aligned}$$

$$- \sum_{\kappa=0}^{M-1} \mu_{\kappa}(\mathbf{k}, \Delta \hat{\mathbf{x}}_{\kappa}). \quad (\text{A}\cdot 9)$$

Differentiating this with respect to  $\Delta \hat{\mathbf{x}}_{\kappa}$  and setting the result to  $\mathbf{0}$ , we obtain

$$\Delta \hat{\mathbf{x}}_{\kappa} = \lambda_{\kappa} \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa+2} + \lambda_{\kappa-1} \hat{\mathbf{x}}_{\kappa+1} \times \hat{\mathbf{x}}_{\kappa-1} + \lambda_{\kappa-2} \hat{\mathbf{x}}_{\kappa-2} \times \hat{\mathbf{x}}_{\kappa-1} + \mu_{\kappa} \mathbf{k} - \tilde{\mathbf{x}}_{\kappa}. \quad (\text{A}\cdot 10)$$

Multiplying  $\mathbf{P}_{\mathbf{k}} = \text{diag}(1, 1, 0)$  on both sides and noting that  $\mathbf{P}_{\mathbf{k}} \Delta \hat{\mathbf{x}}_{\kappa} = \Delta \hat{\mathbf{x}}_{\kappa}$  and  $\mathbf{P}_{\mathbf{k}} \mathbf{k} = \mathbf{0}$ , we obtain Eq. (13).

### Appendix C: Derivation of Eq. (21)

The projection direction  $\Delta \mathbf{x}_0 \oplus \cdots \oplus \Delta \mathbf{x}_{M-1}$  orthogonal to all the hyperplanes in  $\Pi$ , if exists, is determined by minimizing

$$E = \sum_{\kappa=0}^{M-1} \|\Delta \mathbf{x}_{\kappa}\|^2, \quad (\text{A}\cdot 11)$$

subject to Eq. (20) and

$$k_i \Delta x_{\kappa}^i = 0, \quad (\text{A}\cdot 12)$$

which state that the third component of  $\Delta \mathbf{x}_{\kappa}$  be zero ( $\mathbf{k} \equiv (0, 0, 1)^{\top}$  as before). Introducing Lagrange multiplies to Eqs. (20) and (A.12), we write

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\Delta \mathbf{x}_{\kappa}\|^2 - \sum_{\kappa=0}^{M-3} \lambda_{(\kappa)}^{pq} \epsilon_{ljp} \epsilon_{mkq} T_{(\kappa)i}^{lm} \left( \Delta x_{\kappa}^i x_{\kappa+1}^j x_{\kappa+2}^k \right. \\ & \quad \left. + x_{\kappa}^i \Delta x_{\kappa+1}^j x_{\kappa+2}^k + x_{\kappa}^i x_{\kappa+1}^j \Delta x_{\kappa+2}^k \right) \\ & \quad - \sum_{\kappa=0}^{M-1} \mu_{(\kappa)} k_i \Delta x_{\kappa}^i. \end{aligned} \quad (\text{A}\cdot 13)$$

Differentiating this with respect to  $\Delta x_{\kappa}^n$  and letting the result be 0, we obtain

$$\begin{aligned} \Delta x_{\kappa}^n &= \epsilon_{ljp} \epsilon_{mkq} \lambda_{(\kappa)}^{pq} T_{(\kappa)n}^{lm} x_{\kappa+1}^j x_{\kappa+2}^k \\ & \quad + \epsilon_{lnp} \epsilon_{mkq} \lambda_{(\kappa-1)}^{pq} T_{(\kappa-1)i}^{lm} x_{\kappa-1}^i x_{\kappa+1}^k \\ & \quad + \epsilon_{ljp} \epsilon_{mq} \lambda_{(\kappa-2)}^{pq} T_{(\kappa-2)i}^{lm} x_{\kappa-2}^i x_{\kappa-1}^j \\ & \quad + \mu_{(\kappa)} k_n, \end{aligned} \quad (\text{A}\cdot 14)$$

where terms with subscript  $\kappa$  outside the range of  $0, \dots, M-3$  are regarded as 0. Multiplying  $\mathbf{P}_{\mathbf{k}} = \text{diag}(1, 1, 0)$  on both sides and noting that  $\mathbf{P}_{\mathbf{k}} \Delta \mathbf{x}_{\kappa} = \Delta \mathbf{x}_{\kappa}$  and  $\mathbf{P}_{\mathbf{k}} \mathbf{k} = \mathbf{0}$ , we obtain Eq. (21).

### Appendix D: Derivation of Eq. (26)

The projection direction  $\Delta \mathbf{x}_0 \oplus \cdots \oplus \Delta \mathbf{x}_{M-1}$  orthogonal to all the hyperplanes in  $\tilde{\Pi}$ , if exists, is determined by minimizing

$$E = \sum_{\kappa=0}^{M-1} \|\mathbf{x}_{\kappa} - \hat{\mathbf{x}}_{\kappa} + \Delta \hat{\mathbf{x}}_{\kappa}\|^2 = \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_{\kappa} + \Delta \hat{\mathbf{x}}_{\kappa}\|^2. \quad (\text{A}\cdot 15)$$

Introducing Lagrange multipliers to Eqs. (25) and to

$$k_i \Delta \hat{x}_{\kappa}^i = 0, \quad (\text{A}\cdot 16)$$

we write

$$\begin{aligned} & \frac{1}{2} \sum_{\kappa=0}^{M-1} \|\tilde{\mathbf{x}}_{\kappa} + \Delta \hat{\mathbf{x}}_{\kappa}\|^2 - \sum_{\kappa=0}^{M-3} \lambda_{(\kappa)}^{pq} \epsilon_{ljp} \epsilon_{mkq} T_{(\kappa)i}^{lm} \\ & \quad \times \left( \Delta \hat{x}_{\kappa}^i \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k + \hat{x}_{\kappa}^i \Delta \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k + \hat{x}_{\kappa}^i \hat{x}_{\kappa+1}^j \Delta \hat{x}_{\kappa+2}^k \right) \\ & \quad - \sum_{\kappa=0}^{M-1} \mu_{(\kappa)} k_i \Delta \hat{x}_{\kappa}^i. \end{aligned} \quad (\text{A}\cdot 17)$$

Differentiating this with respect to  $\Delta \hat{x}_{\kappa}^n$  and setting the result to 0, we obtain

$$\begin{aligned} \Delta \hat{x}_{\kappa}^n &= \epsilon_{ljp} \epsilon_{mkq} \lambda_{(\kappa)}^{pq} T_{(\kappa)n}^{lm} \hat{x}_{\kappa+1}^j \hat{x}_{\kappa+2}^k \\ & \quad + \epsilon_{lnp} \epsilon_{mkq} \lambda_{(\kappa-1)}^{pq} T_{(\kappa-1)i}^{lm} \hat{x}_{\kappa-1}^i \hat{x}_{\kappa+1}^k \\ & \quad + \epsilon_{ljp} \epsilon_{miq} \lambda_{(\kappa-2)}^{pq} T_{(\kappa-2)i}^{lm} \hat{x}_{\kappa-2}^i \hat{x}_{\kappa-1}^j \\ & \quad + \mu_{(\kappa)} k_n - \tilde{x}_{\kappa}^i. \end{aligned} \quad (\text{A}\cdot 18)$$

Multiplying  $\mathbf{P}_{\mathbf{k}}$  on both sides and noting that  $\mathbf{P}_{\mathbf{k}} \Delta \hat{\mathbf{x}}_{\kappa} = \Delta \hat{\mathbf{x}}_{\kappa}$ ,  $\mathbf{P}_{\mathbf{k}} \mathbf{k} = \mathbf{0}$ , and  $\mathbf{P}_{\mathbf{k}} \tilde{\mathbf{x}}_{\kappa} = \mathbf{0}$  (see Eq. (27)), we obtain Eq. (26).



**Kenichi Kanatani** received his B.E., M.S., and Ph.D. in applied mathematics from the University of Tokyo in 1972, 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Gunma, Japan, he is currently Professor of computer science at Okayama University, Okayama, Japan. He is the author of many books on computer vision and received many awards including the best paper awards from IPSJ (1987) and IEICE (2005). He is an IEEE Fellow.



**Yasuyuki Sugaya** received his B.E., M.S., and Ph.D. in computer science from the University of Tsukuba, Ibaraki, Japan, in 1996, 1998, and 2001, respectively. From 2001 to 2006, he was Assistant Professor of computer science at Okayama University, Okayama, Japan. Currently, he is Associate Professor of computer science and engineering at Toyohashi University of Technology, Toyohashi, Aichi, Japan. His research interests include image processing and computer vision. He received the IEICE best paper award in 2005.



**Hirotaka Niitsuma** received his B.E. an M.S. from Osaka University, Japan, in 1993 and 1995, respectively, and his Ph.D. from NAIST, Japan, in 1999. He was a researcher at TOSHIBA, at JST Corporation, at Denso IT Laboratory, Inc., at Kwansai Gakuin University, Japan, at Kyungpook National University, Korea, and at AIST, Japan. From April 2007, he is Assistant Professor of computer science at Okayama University, Japan. His research interests include computer vision, machine learning, and neural networks.