Estimating Projective Transformation Matrix (Collineation, Homography)

Zhengyou Zhang
Microsoft Research
One Microsoft Way, Redmond, WA 98052, USA
E-mail: zhang@microsoft.com

November 1993; Updated May 29, 2010
Microsoft Research Techical Report MSR-TR-2010-63

Note: *The original version of this report was written in November 1993 while I was at INRIA. It was circulated among very few people, and never published. I am now publishing it as a tech report, adding Section 7, with the hope that it could be useful to more people.*

Abstract

In many applications, one is required to estimate the projective transformation between two sets of points, which is also known as collineation or homography. This report presents a number of techniques for this purpose.

Contents

1 Introduction

Projective Transformation is a concept used in projective geometry to describe how a set of geometric objects maps to another set of geometric objects in *projective space.* The basic intuition behind projective space is to add extra points (points at infinity) to Euclidean space, and the geometric transformation allows to move those extra points to traditional points, and vice versa.

Homogeneous coordinates are used in projective space much as Cartesian coordinates are used in Euclidean space. A point in two dimensions is described by a 3D vector. A point in three dimension is described by a 4D vector. If the homogeneous coordinates of a given point are multiplied by a non-zero scalar, the resulting homogeneous coordinates represent the same point. That is, $\lambda\mathbf{m}$ ($\lambda \neq 0$) and $\mathbf{m}$ represent the same point. Consider a point $\mathbf{p} = [u, v]^T$ on a plane in Euclidean space; its corresponding homogeneous coordinates are $\mathbf{m} = \lambda[u, v, 1]^T$. A point at infinity on the plane is represented by $[\alpha, \beta, 0]^T$, i.e., the last element is 0. A point at infinity in 2D space can be used to describe the direction of a line on the plane. Now consider a point $\mathbf{p} = [x, y, z]^T$ in 3D Euclidean space; its corresponding homogeneous coordinates are $\mathbf{m} = \lambda[x, y, z, 1]^T$. A point at infinity in 3D space is represented by $[\alpha, \beta, \gamma, 0]^T$, i.e., the last element is 0.

Projective linear transformations do not preserve sizes and angles. They do preserve incidence (e.g., points on a line remain on a line after transformation; two lines intersecting with each other will intersect after transformation) and cross-ratio. A projective linear transformation are also known as a *collineation* or *projectivity.* In the case of projective plane ($\mathcal{P}^2$), it is also known as a *homography* or *plane projectivity.* In computer vision, homography plays an important role because any two images of the same planar surface are related by a homography. It has been used for camera calibration [4], image rectification [2], among others.

In this report, I consider the collineation in projective 3D space. The methods described can be easily applied for estimating homography. Also, I have not discussed all the important methods such as robust methods to deal with outliers. The interested reader is referred to my tutorial on parameter estimation [3].

Let $\mathbf{m} = [m_1, m_2, m_3, m_4]^T$ and $\mathbf{m}' = [m'_1, m'_2, m'_3, m'_4]^T$ be two $\mathcal{P}^3$ points in homogeneous coordinates, related by the following collineation:

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} , \tag{1}$$

we have

$$\lambda\mathbf{m}' = \mathbf{P}\mathbf{m} , \tag{2}$$

where λ is an arbitrary scalar because $\mathbf{P}$ is only defined up to a scale.

We can now state the problem as follows:

Given n point correspondences $\{(\mathbf{m}_i,\ \mathbf{m}'_i)\}$, $i = 1, \ldots, n$ with $n \geq 5$, determine the projective transformation $\mathbf{P}$ (collineation matrix) between the two sets of points $\{\mathbf{m}_i\}$ and $\{\mathbf{m}'_i\}$.

2 Method 1: Five-correspondences case

To determine $\mathbf{P}$, we need at least five point correspondences. If we are given 5 point correspondences, to compute $\mathbf{P}$, we first compute the collineation $\mathbf{A}_1$ which maps the standard reference points ($\mathbf{e}_1 = [1,0,0,0]^T$, $\mathbf{e}_2 = [0,1,0,0]^T$, $\mathbf{e}_3 = [0,0,1,0]^T$, $\mathbf{e}_4 = [0,0,0,1]^T$, $\mathbf{e}_5 = [1,1,1,1]^T$) to the first set of points $\mathbf{m}_i$ ($i = 1, \ldots, 5$). This gives

$$\mathbf{A}_1 = [\lambda_1 \mathbf{m}_1, \lambda_2 \mathbf{m}_2, \lambda_3 \mathbf{m}_3, \lambda_4 \mathbf{m}_4] \ ,$$

where

$$[\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4]^{-1} \mathbf{m}_5 \ .$$

Similarly, we can compute the collineation $\mathbf{A}_2$ which maps the standard reference points $\mathbf{e}_i$ ($i = 1, \ldots, 5$) to the second set of points $\mathbf{m}'_i (i = 1, \ldots, 5)$. It is then easy to see that the collineation $\mathbf{P}$ which maps the first set of points $\mathbf{m}_i (i = 1, \ldots, 5)$ to the second set of points $\mathbf{m}'_i$ ($i = 1, \ldots, 5$) is

$$\mathbf{P} = \mathbf{A}_2 \mathbf{A}_1^{-1} \ .$$

The above method is simple. It is however unstable if the data are corrupted by the noise because only the minimum number of points are used. In the following, we will develop several robust methods when more data are available.

3 Method 2: Compute P together with the scalar factors

Given n point correspondences, we have n vector equations:

$$\lambda_i \mathbf{m}'_i = \mathbf{P} \mathbf{m}_i \ . \tag{3}$$

This implies that we have $4n$ equations in $16 + n$ unknowns: P_{ij} ($i = 1, \ldots, 4$; $j = 1, \ldots, 4$) and λ_i ($i = 1, \ldots, n$). If $n > 5$, the problem becomes over-determined.

As $\mathbf{P}$ is defined up to a scale, we can set one of P_{ij}'s to, say, 1. However, this is dangerous because the true value of the P_{ij} we choose could be zero. An alternative and safe way is to set one of λ_i's, say λ_1, to 1.

Let $\mathbf{x} = [P_{11}, P_{12}, \ldots, P_{44}, \lambda_2, \ldots, \lambda_n]^T$ be the vector of the $15 + n$ parameters to be computed. It is easy to say that (3) can be rewritten as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \ , \tag{4}$$

where $\mathbf{A}$ is a $4n \times (15+n)$ matrix[1]

$$\mathbf{A} = \left[\begin{array}{lllll} \mathbf{M}_1 & & & & \\ \mathbf{M}_2 & -\mathbf{m}'_2 & & & \\ \mathbf{M}_3 & & -\mathbf{m}'_3 & & \\ \multicolumn{5}{c}{\dotfill} \\ \mathbf{M}_n & & & & -\mathbf{m}'_n \end{array} \right]$$

with $\mathbf{M}_i = \left[\begin{array}{cccc} \mathbf{m}_i^T & & & \\ & \mathbf{m}_i^T & & \\ & & \mathbf{m}_i^T & \\ & & & \mathbf{m}_i^T \end{array} \right]$, and $\mathbf{b}$ is a $4n$ vector

$$\mathbf{b} = [\mathbf{m}_1'^T, 0, \ldots, 0]^T \ .$$

The least-squares solution to (4) is given by

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \ , \tag{5}$$

which minimizes the error function $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2$.

If the uncertainties in points $(\mathbf{m}_i, \mathbf{m}'_i)$ are known and represented by the covariance matrices $(\mathbf{\Lambda}_{\mathbf{m}_i}, \mathbf{\Lambda}_{\mathbf{m}'_i})$, a weighted least-squares solution can be carried out to obtain a better estimation of $\mathbf{x}$. The error function to be minimized can be defined as

$$\mathcal{F} = (\mathbf{b} - \mathbf{A}\mathbf{x})^T\mathbf{W}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}) \ , \tag{6}$$

where $\mathbf{W}$ is a $4n \times 4n$ matrix:

$$\mathbf{W} = \left[\begin{array}{ccc} \mathtt{W}_1 & & \\ & \ddots & \\ & & \mathtt{W}_n \end{array} \right] ,$$

and $\mathtt{W}_i$ is a 4×4 matrix, the covariance matrix corresponding to the four equations of the ith point pair $(\mathbf{m}_i, \mathbf{m}'_i)$. The $\mathtt{W}_i$'s are computed as follows:

$$\mathtt{W}_i = \mathbf{P}\mathbf{\Lambda}_{\mathbf{m}_i}\mathbf{P}^T + \lambda_i^2\mathbf{\Lambda}_{\mathbf{m}'_i} \ , \tag{7}$$

which requires an estimation of $\mathbf{P}$ and λ_i. The initial estimates of $\mathbf{P}$ and λ_i can be the non-weighted least-squares estimation given by (5). The solution to (6) is given by

$$\mathbf{x} = (\mathbf{A}^T\mathbf{W}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^{-1}\mathbf{b} \ . \tag{8}$$

As the weights are approximately computed by (7), a few iterations are necessary.

One disadvantage of this method is that the system becomes prohibitive when a large number of points, say 1000, are available and we want to take all into account.

[1] In this paper, any element of a matrix which is not explicitly specified is identified to be zero.

4 Method 3: Compute P only (a batch approach)

From (2), the two vectors $\mathbf{m}'$ and $\mathbf{Pm}$ are parallel. Let $\mathbf{v} = \mathbf{Pm} = [v_1, v_2, v_3, v_4]^T$, then we have the following three independent equations:

$$\begin{aligned} m'_2 v_1 - m'_1 v_2 &= 0 \\ m'_3 v_1 - m'_1 v_3 &= 0 \\ m'_4 v_1 - m'_1 v_4 &= 0 \;. \end{aligned} \tag{9}$$

Let $\mathbf{x} = [P_{11}, P_{12}, \ldots, P_{44}]^T$ be the 16 parameter vector to be computed. It is easy to show that (9) is equivalent to the following equation:

$$\mathbf{Bx} = 0 \;, \tag{10}$$

where

$$\mathbf{B} = \begin{bmatrix} m'_2\mathbf{m}^T & -m'_1\mathbf{m}^T & & \\ m'_3\mathbf{m}^T & & -m'_1\mathbf{m}^T & \\ m'_4\mathbf{m}^T & & & -m'_1\mathbf{m}^T \end{bmatrix} . \tag{11}$$

Given n correspondences $(\mathbf{m}_i, \mathbf{m}'_i)$, we have n equations of the type (10). The problem is then to estimate $\mathbf{x}$ by minimizing the following error function:

$$\mathcal{F} = \sum_{i=1}^{n} (\mathbf{B}_i\mathbf{x})^2 = \mathbf{x}^T \left(\sum_{i=1}^{n} \mathbf{B}_i^T \mathbf{B}_i \right) \mathbf{x} \;. \tag{12}$$

Let $\mathbf{A} = \sum_{i=1}^{n} \mathbf{B}_i^T \mathbf{B}_i$, which is a symmetric matrix. As $\mathbf{P}$ is only defined up to a scale, we can normalize $\mathbf{x}$ with $\|\mathbf{x}\| = 1$. It is well known that the solution to (12) is the eigenvector of $\mathbf{A}$ corresponding to the smallest eigenvalue of $\mathbf{A}$.

If the uncertainties in points $(\mathbf{m}_i, \mathbf{m}'_i)$ are known and represented by the covariance matrices $(\mathbf{\Lambda}_{\mathbf{m}_i}, \mathbf{\Lambda}_{\mathbf{m}'_i})$, we can estimate $\mathbf{x}$ by minimizing the following weighted error function

$$\mathcal{F} = \mathbf{x}^T \left(\sum_{i=1}^{n} \mathbf{B}_i^T \mathbf{W}_i^{-1} \mathbf{B}_i \right) \mathbf{x} \equiv \mathbf{x}^T \mathbf{A} \mathbf{x} \;. \tag{13}$$

The solution is still the eigenvector of $\mathbf{A}$ corresponding to the smallest eigenvalue of $\mathbf{A}$. The 3×3 matrix $\mathbf{W}_i$ is the covariance matrix of $\mathbf{B}_i\mathbf{x}$, given by

$$\mathbf{W}_i = J_{\mathbf{m}_i} \mathbf{\Lambda}_{\mathbf{m}_i} J_{\mathbf{m}_i}^T + J_{\mathbf{m}'_i} \mathbf{\Lambda}_{\mathbf{m}'_i} J_{\mathbf{m}'_i}^T \;, \tag{14}$$

where $J_{\mathbf{m}_i}$ and $J_{\mathbf{m}'_i}$ are the Jacobian matrices of $\mathbf{B}_i\mathbf{x}$ with respect to $\mathbf{m}_i$ and $\mathbf{m}'_i$:

$$J_{\mathbf{m}_i} = \begin{bmatrix} m'_2\mathbf{p}_1^T - m'_1\mathbf{p}_2^T \\ m'_3\mathbf{p}_1^T - m'_1\mathbf{p}_3^T \\ m'_4\mathbf{p}_1^T - m'_1\mathbf{p}_4^T \end{bmatrix}$$

$$J_{\mathbf{m}'_i} = \begin{bmatrix} -v_2 & v_1 & & \\ -v_3 & & v_1 & \\ -v_4 & & & v_1 \end{bmatrix} ,$$

where $\mathbf{m}'_i = [m'_1, m'_2, m'_3, m'_4]^T$, $\mathbf{v} = [v_1, v_2, v_3, v_4]^T = \mathbf{P}\mathbf{m}_i$, and $\mathbf{p}_i$ is the ith row of the matrix $\mathbf{P}$, i.e., $\mathbf{p}_i = [P_{i1}, P_{i2}, P_{i3}, P_{i4}]^T$. Thus the computation of the weights, (14), requires an estimation of $\mathbf{P}$, which can be obtained without considering the weights by (12). If the initial estimate of $\mathbf{P}$ is not close enough to the true one, several iterations are required.

Computation of the scalar factors. The scalars, λ_i's, are usually not useful. However, they can be computed either from a simple least-squares solution

$$\lambda_i = \mathbf{m}'^T_i \mathbf{P}\mathbf{m}_i / \mathbf{m}'^T_i \mathbf{m}_i$$

or from a weighted least-squares solution based on (3)

$$\lambda_i = \mathbf{m}'^T_i \mathbf{W}_i^{-1} \mathbf{P}\mathbf{m}_i / \mathbf{m}'^T_i \mathbf{W}_i^{-1} \mathbf{m}_i \ ,$$

where $\mathbf{W}_i$ is given by (7), and requires an initial estimate of λ_i.

5 Method 4: Compute P only (an iterative approach)

The method described just above can be considered as an iterative one. Let $\mathbf{A}^{(n)}$ be the matrix $\mathbf{A}$ computed with n point correspondences. When one more correspondence is available, one only need update $\mathbf{A}$ by

$$\mathbf{A}^{(n+1)} = \mathbf{A}^{(n)} + \mathbf{B}^T_{n+1} \mathbf{W}^{-1}_{n+1} \mathbf{B}_{n+1} \ .$$

The solution after incorporating the new correspondence is then the eigenvector of $\mathbf{A}^{(n+1)}$ corresponding to the smallest eigenvalue of $\mathbf{A}^{(n+1)}$.

Here we present an iterative method based on the Kalman filtering technique. Since $\mathbf{P}$ is defined up to a scale, we should set one element to a nonzero value. Of course, the true value of the element chosen should not be zero. As this method will need an initial estimate of $\mathbf{P}$, we can apply either of the previous methods to the first m ($m \geq 5$) correspondences to obtain such an estimate, and then normalize $\mathbf{P}$ by one of the nonzero elements, say the largest element.

In the following, without loss of generality, we assume the last element of $\mathbf{P}$, P_{44}, is nonzero, and we set it to 1. Let $\mathbf{x} = [P_{11}, P_{12}, \ldots, P_{43}]^T$ be the parameter vector to be estimated. Before processing the ith correspondence, we have an estimate of $\mathbf{x}$, denoted by $\hat{\mathbf{x}}_{i-1}$, and its associated covariance matrix $\mathbf{\Lambda}_{\hat{\mathbf{x}}_{i-1}}$. Given the ith correspondence, we have three scalar equations (9) relating the measurements $(\mathbf{m}_i, \mathbf{m}'_i)$ to the parameter vector $\mathbf{x}$. They are called the measurement/observation equation, denoted by $\mathbf{f}(\mathbf{m}_i, \mathbf{m}'_i, \mathbf{x})$, i.e.,

$$\mathbf{f}(\mathbf{m}_i, \mathbf{m}'_i, \mathbf{x}) = \begin{bmatrix} m'_2 v_1 - m'_1 v_2 \\ m'_3 v_1 - m'_1 v_3 \\ m'_4 v_1 - m'_1 v_4 \end{bmatrix} \ . \qquad (15)$$

As $\mathbf{f}$ is nonlinear, we should apply the extended Kalman filtering technique, that is, we should first linearize the measurement equation at the previous estimate $\hat{\mathbf{x}}_{i-1}$. The linearized measurement equation is:

$$\mathbf{y}_i = \mathbf{M}_i \mathbf{x}_i + \boldsymbol{\xi}_i \ , \tag{16}$$

where $\mathbf{y}_i$ is the new measurement vector, $\boldsymbol{\xi}_i$ is the noise vector of the new measurement, and $\mathbf{M}_i$ is the linearized observation matrix. They are given by

$$\begin{aligned}\mathbf{M}_i &= \frac{\partial \mathbf{f}(\mathbf{m}_i, \mathbf{m}'_i, \hat{\mathbf{x}}_{i-1})}{\partial \mathbf{x}} \\ \mathbf{y}_i &= -\mathbf{f}(\mathbf{m}_i, \mathbf{m}'_i, \hat{\mathbf{x}}_{i-1}) + \mathbf{M}_i \hat{\mathbf{x}}_{i-1} \ ,\end{aligned}$$

and $\boldsymbol{\xi}_i$ has mean zero and covariance as computed by (14). In fact, $\mathbf{M}_i$ is the matrix $\mathbf{B}$ given in (11) with the last column dropped, and $\mathbf{y}_i$ is simply $[0, 0, m'_1 m_4]^T$. Now we can apply the standard Kalman filter to the linearized measurement equation, and an updated estimate $\hat{\mathbf{x}}_i$ can be obtained.

If the initial estimate $\hat{\mathbf{x}}_{i-1}$ is very different from the true one, the first-order approximation we have made is not good anymore. One approach to reducing the influence of the nonlinearities is to apply iteratively the Kalman filter, the so-called *iterative extended Kalman filter*.

6 Method 5: Compute P through normalization

Kanatani [1] proposes a method to compute the collineation $\mathbf{P}$ from $\mathcal{P}^2$ point correspondences $(\mathbf{m}_i, \mathbf{m}'_i)$ where $\mathbf{m}_i$ and $\mathbf{m}'_i$ are normalized so that their norms are 1, and $\mathbf{P}$ is normalized so that $\det \mathbf{P} = 1$. In the following, we generalize the method in order to deal with the $\mathcal{P}^3$ case.

Let h_i be the distance of the endpoint of vector $\mathbf{P}\mathbf{m}_i$ to the line which passes through the starting point of vector $\mathbf{P}\mathbf{m}_i$ and extends in the direction of $\mathbf{m}'_i$. Note that $\|\mathbf{m}_i\| = 1$, $\|\mathbf{m}'_i\| = 1$, and $\det \mathbf{P} = 1$. We have

$$h_i^2 = \|\mathbf{P}\mathbf{m}_i\|^2 - ({\mathbf{m}'_i}^T \mathbf{P}\mathbf{m}_i)^2 \ .$$

The problem is to compute $\mathbf{P}$ by minimizing the following error function

$$\mathcal{F} = \sum_{i=1}^{n} h_i^2 = \sum_{i=1}^{n} \left(\mathbf{m}_i^T \mathbf{P}^T \mathbf{P} \mathbf{m}_i - ({\mathbf{m}'_i}^T \mathbf{P}\mathbf{m}_i)^2 \right) \ . \tag{17}$$

However, it is difficult to compute the solution under the constraint $\det \mathbf{P} = 1$. Instead, the solution under the constraint $\|\mathbf{P}\| = 1$ is computed as an approximation, as to be described below, and then $\mathbf{P}$ is rescaled so that $\det \mathbf{P} = 1$.

Like in Sect. 4, let $\mathbf{x} = [P_{11}, P_{12}, \ldots, P_{44}]^T$ be the 16 parameter vector to be computed. It is easy to show that (17) is equivalent to the following:

$$\mathcal{F} = \mathbf{x}^T \sum_{i=1}^{n} \mathbf{A}_i \mathbf{x} \ , \tag{18}$$

where $\mathbf{A}_i$ is a symmetric matrix given by

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{m}_i & & & \\ & \mathbf{m}_i & & \\ & & \mathbf{m}_i & \\ & & & \mathbf{m}_i \end{bmatrix} \left(\mathbf{I}_4 - \mathbf{m}'_i \mathbf{m}'^T_i\right) \begin{bmatrix} \mathbf{m}_i^T & & & \\ & \mathbf{m}_i^T & & \\ & & \mathbf{m}_i^T & \\ & & & \mathbf{m}_i^T \end{bmatrix} ,$$

where $\mathbf{I}_4$ is the 4×4 identity matrix. Let $\mathbf{A} = \sum_{i=1}^{n} \mathbf{A}_i$, then the solution $\mathbf{x}$ is the eigenvector of $\mathbf{A}$ corresponding to the smallest eigenvalue of $\mathbf{A}$.

7 Method 6: Maximum Likelihood Estimation

For each point $\mathbf{m}_i$, its uncertainty is assumed to be known. To resolve the scalar ambiguity, we assume the last element is set to 1, and we use $\underline{\mathbf{m}}_i$ to denote the first 3 elements of $\mathbf{m}_i$, i.e., $\mathbf{m}_i = [\underline{\mathbf{m}}_i^T, 1]^T$. The uncertainty of $\underline{\mathbf{m}}_i$ is assumed to be represented by covariance matrix $\mathbf{\Lambda}_{\underline{\mathbf{m}}_i}$. If the uncertainty can be considered as identical and isotropic for all points, then it can be set to an identity matrix.

Furthermore, for each given point pair $(\mathbf{m}_i, \mathbf{m}'_i)$, let $\widehat{\mathbf{m}}_i$ be the corresponding *ideal* point in the same coordinate system as $\mathbf{m}_i$. Then, we can formulate the problem as maximum likelihood estimation of both $\mathbf{P}$ and $\{\underline{\widehat{\mathbf{m}}}_i\}$ by solving the following problem:

$$\min_{\mathbf{P}, \{\underline{\widehat{\mathbf{m}}}_i\}} \sum_i [(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i)^T \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1} (\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i) + (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})] \tag{19}$$

This is potentially a large minimization problem. Note, however, each ideal point is only involved in the two corresponding terms. We can rewrite the above problem as

$$\min_{\mathbf{P}} \sum_i \min_{\underline{\widehat{\mathbf{m}}}_i} [(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i)^T \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1} (\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i) + (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})] \tag{20}$$

Now the estimation of each individual $\underline{\widehat{\mathbf{m}}}_i$ is a small problem, and is embedded inside the estimation of $\mathbf{P}$. This makes the overall estimation very efficient. The nonlinear minimization can be performed with the Levenberg-Marquardt algorithm.

In the following, we show that the estimation of $\underline{\widehat{\mathbf{m}}}_i$, i.e.,

$$\min_{\underline{\widehat{\mathbf{m}}}_i} [(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i)^T \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1} (\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i) + (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} (\underline{\mathbf{m}}'_i - \underline{\mathbf{P}\widehat{\mathbf{m}}_i})] \tag{21}$$

can be performed through linearlization. Let

$$\underline{\widehat{\mathbf{m}}}'_i \equiv \underline{\mathbf{P}\widehat{\mathbf{m}}_i} = \frac{1}{\mathbf{p}_4^T \widehat{\mathbf{m}}_i} \begin{bmatrix} \mathbf{p}_1^T \widehat{\mathbf{m}}_i \\ \mathbf{p}_2^T \widehat{\mathbf{m}}_i \\ \mathbf{p}_3^T \widehat{\mathbf{m}}_i \end{bmatrix} \tag{22}$$

where $\mathbf{p}_j$ $(j = 1, 2, 3, 4)$ is the j-th *row* vector of matrix $\mathbf{P}$. Given an initial estimate of $\widehat{\mathbf{m}}_i$, denoted by $\widehat{\mathbf{m}}_i^-$, we can linearize it around $\underline{\widehat{\mathbf{m}}}_i^-$ as

$$\underline{\widehat{\mathbf{m}}}'_i = \underline{\widehat{\mathbf{m}}}_i^{'-} + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-} (\underline{\widehat{\mathbf{m}}}_i - \underline{\widehat{\mathbf{m}}}_i^-) \tag{23}$$

where

$$\underline{\widehat{\mathbf{m}}}_i'^{-} \equiv \frac{1}{\mathbf{p}_4^T \widehat{\mathbf{m}}_i^-} \begin{bmatrix} \mathbf{p}_1^T \widehat{\mathbf{m}}_i^- \\ \mathbf{p}_2^T \widehat{\mathbf{m}}_i^- \\ \mathbf{p}_3^T \widehat{\mathbf{m}}_i^- \end{bmatrix} \tag{24}$$

$$\mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-} \equiv \frac{\partial \underline{\mathbf{P}\widehat{\mathbf{m}}_i}}{\partial \underline{\widehat{\mathbf{m}}}_i}\Big|_{\underline{\widehat{\mathbf{m}}}_i = \underline{\widehat{\mathbf{m}}}_i^-} \tag{25}$$

After some simple algebra, we get the Jacobian matrix $\mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}$ as

$$\mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-} = \frac{1}{\mathbf{p}_4^T \widehat{\mathbf{m}}_i^-}(\mathbf{P}_{3\times 3} - \underline{\widehat{\mathbf{m}}}_i'^{-} \begin{bmatrix} P_{41} & P_{42} & P_{43} \end{bmatrix}) \tag{26}$$

where $\mathbf{P}_{3\times 3}$ is the upper left 3×3 submatrix of $\mathbf{P}$, and P_{kl} is the (k,l)-element of $\mathbf{P}$.

With the above linearization, the objective function in (21) can be rewritten as

$$\begin{aligned} \mathcal{F}(\underline{\widehat{\mathbf{m}}}_i) = {} & (\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i)^T \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1} (\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i) \\ & + (\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-} - \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}(\underline{\widehat{\mathbf{m}}}_i - \underline{\widehat{\mathbf{m}}}_i^-))^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} (\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-} - \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}(\underline{\widehat{\mathbf{m}}}_i - \underline{\widehat{\mathbf{m}}}_i^-)) \end{aligned} \tag{27}$$

The derivative of $\mathcal{F}(\underline{\widehat{\mathbf{m}}}_i)$ with respect to $\underline{\widehat{\mathbf{m}}}_i$ (after ignoring the constant factor) is given by

$$\frac{\partial \mathcal{F}(\underline{\widehat{\mathbf{m}}}_i)}{\partial \underline{\widehat{\mathbf{m}}}_i} = \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1}(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i) + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1}(\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-} - \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}(\underline{\widehat{\mathbf{m}}}_i - \underline{\widehat{\mathbf{m}}}_i^-)) \tag{28}$$

Setting it to zero yields the solution to $\underline{\widehat{\mathbf{m}}}_i$, which is

$$\underline{\widehat{\mathbf{m}}}_i = \mathbf{W}_i^{-1}[\mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1}\underline{\mathbf{m}}_i + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1}(\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-} + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}\underline{\widehat{\mathbf{m}}}_i^-)] \tag{29}$$

$$\mathbf{W}_i \equiv \mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1} + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-} \tag{30}$$

Now, substituting $\underline{\widehat{\mathbf{m}}}_i$ with the above solution, we can easily get

$$\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i = \mathbf{W}_i^{-1}[\mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1} \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i^-) - \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1}(\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-})] \tag{31}$$

$$\underline{\widehat{\mathbf{m}}}_i - \underline{\widehat{\mathbf{m}}}_i^- = \mathbf{W}_i^{-1}[\mathbf{\Lambda}_{\underline{\mathbf{m}}_i}^{-1}(\underline{\mathbf{m}}_i - \underline{\widehat{\mathbf{m}}}_i^-) + \mathbf{J}_{\underline{\widehat{\mathbf{m}}}_i^-}^T \mathbf{\Lambda}_{\underline{\mathbf{m}}'_i}^{-1}(\underline{\mathbf{m}}'_i - \underline{\widehat{\mathbf{m}}}_i'^{-})] \tag{32}$$

The objective function (27) can then be computed explicitly.

Finally, let's discuss how to choose the initial guess $\underline{\widehat{\mathbf{m}}}_i^-$. During the iteration process, it can be computed from the previous iteration using (29). For the very first iteration, a good initial guess is obviously $\underline{\widehat{\mathbf{m}}}_i^- = \underline{\mathbf{m}}_i$. Yet a better initial guess is to also consider $\mathbf{m}'_i$. We first transform $\mathbf{m}'_i$ to the first coordinate system with the initial guess of $\mathbf{P}$, and then average them. That is:

$$\underline{\widehat{\mathbf{m}}}_i^- = \frac{1}{2}(\underline{\mathbf{m}}_i + \underline{\mathbf{P}^{-1}\mathbf{m}'_i}) \tag{33}$$

References

[1] K. Kanatani. *Geometric computation for machine vision.* Oxford university press, 1992.

[2] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado, June 1999. IEEE Computer Society.

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

[4] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.