Research Article

A Linear Method to Derive 3D Projective Invariants from 4 Uncalibrated Images

YuanBin Wang, 1 XingWei Wang, 1 Bin Zhang, 1 and Ying Wang 2

1 College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
2 Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA

Correspondence should be addressed to XingWei Wang; wangxingwei@ise.neu.edu.cn

Received 14 August 2013; Accepted 14 November 2013; Published 29 January 2014

Academic Editors: J. Shu and F. Yu

Copyright © 2014 YuanBin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A well-known method proposed by Quan to compute projective invariants of 3D points uses six points in three 2D images. The method is nonlinear and complicated. It usually produces three possible solutions. It is noted previously that the problem can be solved directly and linearly using six points in five images. This paper presents a method to compute projective invariants of 3D points from four uncalibrated images directly. For a set of six 3D points in general position, we choose four of them as the reference basis and represent the other two points under this basis. It is known that the cross ratios of the coefficients of these representations are projective invariant. After a series of linear transformations, a system of four bilinear equations in the three unknown projective invariants is derived. Systems of nonlinear multivariable equations are usually hard to solve. We show that this form of equations can be solved linearly and uniquely. This finding is remarkable. It means that the natural configuration of the projective reconstruction problem might be six points and four images. The solutions are given in explicit formulas.

1. Introduction

The recovery of the geometric structure of 3D points from 2D images is fundamental in computer vision. After decades of research, most of the mathematical aspect of this problem is well understood. It is proved that the geometric information of a 3D point configuration cannot be recovered from a single image, unless the configuration is further constrained [1]. When two or more images are available, the 3D structure of a scene can be recovered up to an unknown projective transformation. The projective reconstruction of camera parameters and 3D scene structure from multiple uncalibrated views is also called projective structure and motion [1–5].

A camera is a device that transforms properties of a 3D scene onto an image plane. A pinhole camera model is used to represent the linear projection from 3D space onto each image plane. In this paper, 3D world points are represented by homogeneous 4-vector \( \mathbf{X}_i = (x_i, y_i, z_i, 1)^T \). The projection of the \( i \)th 3D point is represented by a homogenous 3-vector \( \mathbf{x}_i = (u_i, v_i, 1)^T \). The relationships among the 3D points \( \mathbf{X}_i \) and their 2D projections are

\[
    k_i^j \mathbf{x}_i = \mathbf{P}^j \mathbf{X}_i, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m,
\]

where \( \mathbf{P}^j \) is the projection matrix (which is \( 3 \times 4 \) and is also called the camera matrix) of the \( j \)th camera, \( k_i^j \) is a nonzero scale factor called projective depth, and \( \mathbf{x}_i^j = (u_i^j, v_i^j, 1)^T \) is the \( j \)th projection of the \( i \)th 3D point. Suppose that \( m \) perspective images of a set of \( n \) 3D points are given. The structure and motion problem is to recover the 3D point locations and camera locations from the image measurements. When the cameras are uncalibrated and no additional geometric information of the point set is available, the reconstruction is determined only up to an unknown projective transformation. For any 3D projective transformation matrix \( \mathbf{H} \), \( \mathbf{P}, \mathbf{H}^{-1} \) and \( \mathbf{H} \mathbf{X}_i \) produce an equally valid reconstruction.

Existing methods for projective reconstruction are usually indirect. They rely on an a priori estimation of some tensors of multiple images of the scene to estimate the 3D point structure. A second-order tensor usually called the fundamental matrix captures the geometry between two views of a 3D scene. A third-order tensor usually called trifocal tensor captures the geometry among three views of a 3D scene. When these tensors of multiple views of a scene are known, there are many algorithms to recover the 3D geometric structure of the scene from them [6–17].
We can also compute 3D projective invariants of a point set from its 2D images directly. In the famous paper [9], Quan proposed a method to compute 3D projective invariants of six 3D points from three uncalibrated images. However, the method proposed by Quan is rather complicated and hard to use in real applications.

This paper presents a fast linear method for computing projective invariants of six 3D points from four 2D view images. A 3D point structure can be configured by first choosing four reference points as a basis and then representing the other two points under this basis. The cross ratios of the coordinates of the other two points under this basis are given in explicit formulas. The six unknown points in 3D space are projective equivalent to the following normalized points:

\[
X_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad X_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},
\]

\[
X_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad X_5 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad X_6 = \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}.
\]

The known point locations in the three 2D images are first normalized according to the projective basis. After this, the known point locations in the jth image are then corresponding to

\[
x_j^i = \begin{pmatrix} u_j^i \\ v_j^i \\ w_j^i \end{pmatrix}, \quad j = 1, 2, 3.
\]

From these correspondence relations, a homogeneous non-linear equation of the form

\[
i_x^1 X + i_x^2 X Z + i_x^3 X T + i_x^4 Y Z + i_x^5 Y T + i_x^6 Z T = 0
\]

can be derived for the jth image, where

\[
i_x^1 = u_j^i (w_j^i - v_j^i), \quad i_x^2 = v_j^i (w_j^i - u_j^i),
\]

\[
i_x^3 = u_j^i (v_j^i - w_j^i), \quad i_x^4 = v_j^i (v_j^i - w_j^i),
\]

\[
i_x^5 = v_j^i (w_j^i - u_j^i), \quad i_x^6 = w_j^i (u_j^i - v_j^i), \quad j = 1, 2, 3.
\]

It is also noticed that

\[
i_x^i + i_x^2 + i_x^3 + i_x^4 + i_x^5 + i_x^6 = 0, \quad j = 1, 2, 3.
\]

Since six 3D points have 18 degrees of freedom and a 3D projective transformation has 15 degrees of freedom, six points in 3D space can have \(18 - 15 = 3\) independent projective invariants. There are many forms of projective invariants. It is noticed that the ratios of \(X, Y, Z,\) and \(T\) in (7) are projective invariant. The three independent such invariants can be

\[
\alpha = \frac{X}{T}, \quad \beta = \frac{Y}{T}, \quad \gamma = \frac{Z}{T}.
\]
We begin by considering a set of 3D points which are seen from three of the 2D images. Quan tried to solve the system of bilinear equations (7) using the classical resultant technique. After eliminating the variable $Z$, he obtained two homogeneous polynomial equations of the third degree in three variables

$$G_1 = e_1 X^2 Y + e_2 X Y^2 + e_3 X Y T + e_4 T^2 = 0,$$

$$G_2 = e_5 X^2 Y + e_6 X Y^2 + e_4 X Y T + e_7 T^2 = 0.$$  \hspace{1cm} (11)

Eliminating $Y$ again will result in a homogeneous polynomial equation in $X$ and $T$ of degree eight. After that, a third degree polynomial equation can be derived numerically through polynomial factorization of the following form:

$$X^T (X - T) \left( b_1 X^2 + b_2 X T + b_3 T^3 \right) \times \left( a_4 X^3 + a_2 X^2 T + a_3 X T^2 + a_4 T^3 \right) = 0.$$  \hspace{1cm} (12)

As we can see from the procedure described above, the method proposed by Quan is hard to implement by ordinary users and inconvenient for real applications. In [13], the author proposed a method to eliminate variable $y$ and variable $beta$ in a single step. A third degree polynomial equation in single variable $alpha$ was given explicitly.

3. A Linear Method to Compute Projective Invariants from 4 Images

A novel direct method for computing projective invariants of six 3D points from four images is presented in this section. We begin by considering a set of 3D points which are seen from four views.

Suppose that a set of six 3D points labeled $X_i$ are given, the geometric structure of which is unknown. The point set is projected into view images by four unknown camera matrices $P_1^i$, $P_2^i$, $P_3^i$, and $P_4^i$. The relationships between them are

$$k_i^j X_i^j = P_i^j X_i, \quad i = 1, \ldots, 6, \quad j = 1, 2, 3, 4. \hspace{1cm} (13)$$

The only information available is the point locations in the four images and point correspondences between the four projections

$$X_i^1 = \begin{pmatrix} u_i^1 \\ v_i^1 \end{pmatrix} \longrightarrow X_i^2 = \begin{pmatrix} u_i^2 \\ v_i^2 \end{pmatrix} \longrightarrow X_i^3 = \begin{pmatrix} u_i^3 \\ v_i^3 \end{pmatrix} \longrightarrow X_i^4 = \begin{pmatrix} u_i^4 \\ v_i^4 \end{pmatrix}, \hspace{1cm} (14)$$

where $i = 1, \ldots, 6$. It is often supposed that no four points in space are coplanar and no three points in the images are collinear. Otherwise the problem is much simpler.

Points $X_5$ and $X_6$ can be represented as linear combinations of $X_1$, $X_2$, $X_3$, and $X_4$

$$X_5 = \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \alpha_4 X_4,$$

$$X_6 = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4.$$  \hspace{1cm} (15)

Since points $X_1$, $X_2$, $X_3$, and $X_4$ are linearly independent, this representation is unique and all the $alpha$ and $beta$ are nonzero. There are many forms of projective invariants. It is observed that the cross ratios of coefficients in (15) are projective invariant. Six 3D points have 18 degrees of freedom and 3D projective transformation has 15 degrees of freedom. So, six 3D points can have 3 independent projective invariants. A set of functional independent projective invariants of this form are

$$I_1 = \frac{\alpha_1 \beta_2}{\alpha_3 \beta_1}, \quad I_2 = \frac{\alpha_2 \beta_3}{\alpha_4 \beta_2}, \quad I_3 = \frac{\alpha_3 \beta_4}{\alpha_1 \beta_3}. \quad \hspace{1cm} (16)$$

The projective invariance of $I_1$, $I_2$, and $I_3$ can be proved easily. Suppose that the six points $Y_1$, $Y_2$, $Y_3$, $Y_4$, $Y_5$, and $Y_6$ are transformed into $Y_1$, $Y_2$, $Y_3$, $Y_4$, $Y_5$, and $Y_6$ by a 3D projective transformation $A$, where $A$ is a $4 \times 4$ full rank matrix. That is, $\lambda_i Y_i = AX_i$, $i = 1, 2, 3, 4, 5, 6$. \hspace{1cm} (17)

where $\lambda_i$ is a nonzero real number. Let

$$Y_5 = \alpha_1 Y_1 + \alpha_2 Y_2 + \alpha_3 Y_3 + \alpha_4 Y_4,$$

$$Y_6 = \beta_1 Y_1 + \beta_2 Y_2 + \beta_3 Y_3 + \beta_4 Y_4.$$  \hspace{1cm} (18)

be the linear representations of $Y_5$ and $Y_6$ in $Y_1$, $Y_2$, $Y_3$, and $Y_4$. Multiplying each side of each equation in (18) by matrix $A^{-1}$, we have

$$X_5 = \frac{\alpha_1}{\lambda_1} \lambda_5 X_1 + \frac{\alpha_2}{\lambda_2} \lambda_5 X_2 + \frac{\alpha_3}{\lambda_3} \lambda_5 X_3 + \frac{\alpha_4}{\lambda_4} \lambda_5 X_4,$$

$$X_6 = \frac{\beta_1}{\lambda_1} \lambda_6 X_1 + \frac{\beta_2}{\lambda_2} \lambda_6 X_2 + \frac{\beta_3}{\lambda_3} \lambda_6 X_3 + \frac{\beta_4}{\lambda_4} \lambda_6 X_4.$$  \hspace{1cm} (19)

Since vectors $X_1$, $X_2$, $X_3$, and $X_4$ are linearly independent, the linear representations in (15) and (19) are exactly the same. So we have

$$I_1 = \frac{\alpha_1 \beta_2}{\alpha_3 \beta_1} = \frac{\alpha_1 \beta_2}{\alpha_3 \beta_1} = I_1,$$

$$I_2 = \frac{\alpha_2 \beta_3}{\alpha_4 \beta_2} = \frac{\alpha_2 \beta_3}{\alpha_4 \beta_2} = I_2,$$

$$I_3 = \frac{\alpha_3 \beta_4}{\alpha_1 \beta_3} = \frac{\alpha_3 \beta_4}{\alpha_1 \beta_3} = I_3.$$  \hspace{1cm} (20)

This proved the invariance of $I_1$, $I_2$, and $I_3$.

The set of projective invariants in (16) have the property that when an invariant equals one, four of the 3D points are
coplanar. This can be proved easily. For example, if \( I_1 = 1 \), then 
\[
\alpha_1 \beta_2 = \alpha_2 \beta_1.
\]
From (15), we have
\[
\begin{align*}
\beta_1 X_5 &= \alpha_1 \beta_1 X_1 + \alpha_2 \beta_2 X_2 + \alpha_3 \beta_1 X_3 + \alpha_4 \beta_2 X_4, \\
\alpha_1 X_6 &= \alpha_1 \beta_1 X_1 + \alpha_2 \beta_2 X_2 + \alpha_3 \beta_3 X_3 + \alpha_4 \beta_2 X_4.
\end{align*}
\]
(21)

Subtracting one equation from the other equation in (21), we get
\[
(\alpha_1 \beta_3 - \alpha_3 \beta_3) X_3 + (\alpha_1 \beta_4 - \alpha_4 \beta_4) X_4 + \beta_1 X_3 - \alpha_1 X_6 = 0.
\]
(22)

Since \( \alpha_1 \) and \( \beta_1 \) are not zero, we have a nontrivial linear combination of points \( X_5, X_6, X_4, \) and \( X_6 \). So they are coplanar.

On the other hand, if points \( X_2, X_3, X_5, \) and \( X_6 \) are coplanar, then there are numbers \( \eta_5, \eta_4, \eta_3, \) and \( \eta_6 \) which are not all zero such that
\[
\eta_3 X_3 + \eta_4 X_4 + \eta_5 X_5 + \eta_6 X_6 = 0.
\]
(23)

Substituting \( X_5 \) and \( X_6 \) using (15) into (23), we obtain
\[
(\alpha_1 \eta_5 + \beta_1 \eta_6) X_1 + (\alpha_2 \eta_5 + \beta_2 \eta_6) X_2 \\
+ (\alpha_3 \eta_5 + \beta_3 \eta_6) X_3 + (\alpha_4 \eta_5 + \beta_4 \eta_6) X_4 = 0.
\]
(24)

Since points \( X_1, X_2, X_3, \) and \( X_4 \) are not coplanar, the coefficients in (24) have to be exactly zero. From this condition we have
\[
\begin{align*}
\alpha_1 \eta_5 + \beta_1 \eta_6 &= 0, \\
\alpha_2 \eta_5 + \beta_2 \eta_6 &= 0.
\end{align*}
\]
(25)

From (25), we obtain
\[
I_1 = \frac{\alpha_2 \beta_2}{\alpha_2 \beta_1} = 1.
\]
(26)

This proved the claim that the necessary and sufficient condition for four of the six points to be coplanar is that one of the projective invariants equals one.

Our next objective is to derive these invariants from image point correspondences. Multiplying each side of (15) by the projection matrices \( P^1, P^2, P^3, \) and \( P^4 \), we have
\[
\begin{align*}
\kappa_5 x_5^i &= \alpha_1 k_1^i x_1 + \alpha_2 k_2^i x_2 + \alpha_3 k_3^i x_3 + \alpha_4 k_4^i x_4, \\
\kappa_6 x_6^i &= \beta_1 k_1^i x_1 + \beta_2 k_2^i x_2 + \beta_3 k_3^i x_3 + \beta_4 k_4^i x_4,
\end{align*}
\]
(27)

That is,
\[
\begin{align*}
\kappa_5 x_5^i &= \alpha_1 k_1^i u_1 + \alpha_2 k_2^i u_2 + \alpha_3 k_3^i u_3 + \alpha_4 k_4^i u_4, \\
\kappa_6 x_6^i &= \beta_1 k_1^i u_1 + \beta_2 k_2^i u_2 + \beta_3 k_3^i u_3 + \beta_4 k_4^i u_4,
\end{align*}
\]
\[
\begin{align*}
\kappa_5 x_5^i &= \alpha_1 k_1^i v_1 + \alpha_2 k_2^i v_2 + \alpha_3 k_3^i v_3 + \alpha_4 k_4^i v_4, \\
\kappa_6 x_6^i &= \beta_1 k_1^i v_1 + \beta_2 k_2^i v_2 + \beta_3 k_3^i v_3 + \beta_4 k_4^i v_4,
\end{align*}
\]
\[
\begin{align*}
\kappa_5 x_5^i &= \alpha_1 k_1^i w_1 + \alpha_2 k_2^i w_2 + \alpha_3 k_3^i w_3 + \alpha_4 k_4^i w_4, \\
\kappa_6 x_6^i &= \beta_1 k_1^i w_1 + \beta_2 k_2^i w_2 + \beta_3 k_3^i w_3 + \beta_4 k_4^i w_4,
\end{align*}
\]
(28)

Applying variable eliminations to (28), we get
\[
\begin{align*}
\alpha_1 k_1^i a_1^i + \alpha_2 k_2^i a_2^i + \alpha_3 k_3^i a_3^i + \alpha_4 k_4^i a_4^i &= 0, \\
\alpha_1 k_1^i b_1^i + \alpha_2 k_2^i b_2^i + \alpha_3 k_3^i b_3^i + \alpha_4 k_4^i b_4^i &= 0, \\
\beta_1 k_1^i c_1^i + \beta_2 k_2^i c_2^i + \beta_3 k_3^i c_3^i + \beta_4 k_4^i c_4^i &= 0, \\
\beta_1 k_1^i d_1^i + \beta_2 k_2^i d_2^i + \beta_3 k_3^i d_3^i + \beta_4 k_4^i d_4^i &= 0,
\end{align*}
\]
(29)

where
\[
\begin{align*}
& a_j^i = u_j^i - u_s^i, \\
& b_j^i = v_j^i - v_s^i, \\
& c_j^i = u_j^i - u_t^i, \\
& d_j^i = v_j^i - v_t^i,
\end{align*}
\]
(30)

Rewriting (29) in another form, we have
\[
\begin{align*}
& a_1^i + a_2^i K_1 + a_3^i K_2 + a_4^i K_3 = 0, \\
& b_1^i + b_2^i K_1 + b_3^i K_2 + b_4^i K_3 = 0, \\
& c_1^i + c_2^i I_1 K_1 + c_3^i I_2 K_2 + c_4^i I_3 K_3 = 0, \\
& d_1^i + d_2^i I_1 K_1 + d_3^i I_2 K_2 + d_4^i I_3 K_3 = 0,
\end{align*}
\]
(31)

Since we have known in advance that the systems of equations in (31) have nontrivial solutions, the coefficients matrices in (31) must be rank deficient. That is
\[
\text{det}
\begin{pmatrix}
\begin{align*}
a_1^i & a_2^i & a_3^i & a_4^i \\
b_1^i & b_2^i & b_3^i & b_4^i \\
c_1^i & c_2^i & c_3^i & c_4^i \\
d_1^i & d_2^i & d_3^i & d_4^i
\end{align*}
\end{pmatrix}
= 0,
\]
(34)

where
\[
\begin{align*}
K_1 &= \frac{\alpha_2 k_2^i}{\alpha_1 k_1^i}, \\
K_2 &= \frac{\alpha_3 k_3^i}{\alpha_1 k_1^i}, \\
K_3 &= \frac{\alpha_4 k_4^i}{\alpha_1 k_1^i},
\end{align*}
\]
(32)

\[
\begin{align*}
I_1 &= \frac{\alpha_1 \beta_2}{\alpha_2 \beta_1}, \\
I_2 &= \frac{\alpha_1 \beta_3}{\alpha_3 \beta_1}, \\
I_3 &= \frac{\alpha_1 \beta_4}{\alpha_4 \beta_1},
\end{align*}
\]
(33)

Using these constraints, we can obtain a system of four bilinear equations in variables \( I_1, I_2, \) and \( I_3 \) of the following form:
\[
\begin{align*}
& \hat{t}_1 I_1 + \hat{t}_2 I_2 + \hat{t}_3 I_3 + \hat{t}_4 I_2 I_3 + \hat{t}_5 I_1 I_3 + \hat{t}_6 I_2 I_3 = 0,
\end{align*}
\]
(35)
where

\[ t_i' = a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4 + a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4, \]

\[ t_i' = a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4 + a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4, \]

\[ t_i' = a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4 + a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4, \]

\[ t_i' = a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4 + a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4, \]

\[ t_i' = a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4 + a_i^4b_i^4c_i^4d_i^4 - a_i^4b_i^4c_i^4d_i^4, \]

\[ i = 1, 2, 3, 4. \]

Applying constraints (39) to (43), we obtain the following

\[ \begin{vmatrix} t_1' + t_2'I_1 & t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' \\ t_2' + t_3'I_2 & t_2' + t_4'I_2 & t_2' + t_5'I_2 & t_2' \\ t_3' + t_4'I_3 & t_3' + t_5'I_3 & t_3' + t_6'I_3 & t_3' \\ t_6' + t_5'I_6 & t_6' + t_4'I_6 & t_6' + t_3'I_6 & t_6' \end{vmatrix} = 0. \]  
\[ (36) \]

The solution of (40) are \( I_1 = 1 \) and

\[ I_1 = -\frac{\begin{vmatrix} t_1' + t_2'I_1 & t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' \\ t_2' + t_3'I_2 & t_2' + t_4'I_2 & t_2' + t_5'I_2 & t_2' \\ t_3' + t_4'I_3 & t_3' + t_5'I_3 & t_3' + t_6'I_3 & t_3' \\ t_6' + t_5'I_6 & t_6' + t_4'I_6 & t_6' + t_3'I_6 & t_6' \end{vmatrix}}{\begin{vmatrix} t_1' + t_2'I_1 & t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' \\ t_2' + t_3'I_2 & t_2' + t_4'I_2 & t_2' + t_5'I_2 & t_2' \\ t_3' + t_4'I_3 & t_3' + t_5'I_3 & t_3' + t_6'I_3 & t_3' \\ t_6' + t_5'I_6 & t_6' + t_4'I_6 & t_6' + t_3'I_6 & t_6' \end{vmatrix}}. \]  
\[ (41) \]

The solution \( I_1 = 1 \) corresponds to the condition that four of the 3D points are coplanar. We neglect this solution according to the assumption that no four points are coplanar. In this way a unique linear solution of the projective invariant \( I_1 \) is obtained.

Now we derive the solution of \( I_2 \). From (35), we can obtain

\[ \begin{vmatrix} t_1' + t_4'I_1 & t_1' + t_6'I_1 & t_1' + t_5'I_2 & t_1' + t_2'I_2 \\ t_2' + t_4'I_2 & t_2' + t_6'I_2 & t_2' + t_5'I_3 & t_2' + t_3'I_3 \\ t_3' + t_4'I_3 & t_3' + t_6'I_3 & t_3' + t_5'I_4 & t_3' + t_4'I_4 \\ t_4' + t_6'I_4 & t_4' + t_5'I_4 & t_4' + t_3'I_5 & t_4' + t_2'I_5 \end{vmatrix} = 0. \]  
\[ (42) \]

Since \( I_1, I_2, \) and \( I_3 \) are nonzero, we have

\[ \begin{vmatrix} t_1' + t_6'I_1 & t_1' + t_5'I_2 & t_1' + t_2'I_3 & t_1' + t_3'I_4 \\ t_2' + t_6'I_2 & t_2' + t_5'I_3 & t_2' + t_3'I_4 & t_2' + t_4'I_5 \\ t_3' + t_6'I_3 & t_3' + t_5'I_4 & t_3' + t_2'I_5 & t_3' + t_4'I_6 \\ t_4' + t_6'I_4 & t_4' + t_5'I_5 & t_4' + t_3'I_6 & t_4' + t_2'I_7 \end{vmatrix} = 0. \]  
\[ (43) \]

Applying constraints (39) to (43), we obtain

\[ \begin{vmatrix} (t_1' + t_6')(I_2 - 1) & (t_1' + t_5'I_2) & (t_1' + t_2'I_3) & (t_1' + t_3'I_4) \\ (t_2' + t_6')(I_2 - 1) & (t_2' + t_5'I_3) & (t_2' + t_3'I_4) & (t_2' + t_4'I_5) \\ (t_3' + t_6')(I_2 - 1) & (t_3' + t_5'I_4) & (t_3' + t_2'I_5) & (t_3' + t_4'I_6) \\ (t_4' + t_6')(I_2 - 1) & (t_4' + t_5'I_5) & (t_4' + t_2'I_6) & (t_4' + t_3'I_7) \end{vmatrix} = 0. \]  
\[ (44) \]

Then the unique solution of \( I_2 \) is

\[ I_2 = -\frac{\begin{vmatrix} t_1' + t_2'I_1 & t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' \\ t_2' + t_3'I_2 & t_2' + t_4'I_2 & t_2' + t_5'I_2 & t_2' \\ t_3' + t_4'I_3 & t_3' + t_5'I_3 & t_3' + t_6'I_3 & t_3' \\ t_6' + t_5'I_6 & t_6' + t_4'I_6 & t_6' + t_3'I_6 & t_6' \end{vmatrix}}{\begin{vmatrix} t_1' + t_2'I_1 & t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' \\ t_2' + t_3'I_2 & t_2' + t_4'I_2 & t_2' + t_5'I_2 & t_2' \\ t_3' + t_4'I_3 & t_3' + t_5'I_3 & t_3' + t_6'I_3 & t_3' \\ t_6' + t_5'I_6 & t_6' + t_4'I_6 & t_6' + t_3'I_6 & t_6' \end{vmatrix}}. \]  
\[ (45) \]

Now we derive the solution of \( I_3 \). From (35), we can obtain

\[ \begin{vmatrix} t_1' + t_3'I_1 & t_1' + t_6'I_1 & t_1' + t_5'I_2 & t_1' + t_2'I_2 \\ t_2' + t_3'I_2 & t_2' + t_6'I_2 & t_2' + t_5'I_3 & t_2' + t_4'I_3 \\ t_3' + t_6'I_3 & t_3' + t_5'I_3 & t_3' + t_2'I_4 & t_3' + t_1'I_4 \\ t_4' + t_5'I_4 & t_4' + t_6'I_4 & t_4' + t_3'I_5 & t_4' + t_2'I_5 \end{vmatrix} = 0. \]  
\[ (46) \]
\text{X} = \text{RandomReal}[{-1000, 1000}, \{6, 4\}];
\text{M} = \text{RandomReal}[{-1, 1}, \{4, 3, 4\}];
\text{T} = \text{RandomReal}[{0, 1}, \{4, 6\}];
\text{x} = \text{RandomReal}[{0, 1}, \{4, 6, 4\}];
\text{a} = \text{RandomReal}[{0, 1}, \{4, 6, 4\}];
\text{b} = \text{RandomReal}[{0, 1}, \{4, 6, 4\}];
\text{u} = \text{RandomReal}[{0, 1}, \{4, 6\}];
\text{v} = \text{RandomReal}[{0, 1}, \{4, 6\}];
\text{X}[[1, 4]] = 1;
\text{X}[[2, 4]] = 1;
\text{X}[[3, 4]] = 1;
\text{X}[[4, 4]] = 1;
\text{X}[[5, 4]] = 1;
\text{X}[[6, 4]] = 1;
\text{XT} = \text{Transpose}[\{\text{X}[[1]], \text{X}[[2]], \text{X}[[3]], \text{X}[[4]]\}];
\text{A} = \text{LinearSolve}[\text{XT}, \text{X}[[5]]];
\text{B} = \text{LinearSolve}[\text{XT}, \text{X}[[6]]];
\text{Inv1} = (\text{A}[[1]] \text{B}[[2]])/(\text{A}[[2]] \text{B}[[1]]);
\text{Inv2} = (\text{A}[[1]] \text{B}[[3]])/(\text{A}[[3]] \text{B}[[1]]);
\text{Inv3} = (\text{A}[[1]] \text{B}[[4]])/(\text{A}[[4]] \text{B}[[1]]);
\text{Print}("\text{The three invariants computed from 3D point locations: }\)"	ext{, Inv1, " \ \ \text{" }, Inv2, " \ \ \text{" }, Inv3);%
\text{For}[i = 1, i < 4, i++, \text{For}[j = 1, j < 6, j++,
\text{x}[[i, j]] = \text{M}[[i]] \cdot \text{X}[[j]];
\text{u}[[i, j]] = \text{x}[[i, j, 1]]/\text{x}[[i, j, 3]];
\text{v}[[i, j]] = \text{x}[[i, j, 2]]/\text{x}[[i, j, 3]];
\text{]};
\text{For}[i = 1, i < 4, i++, \text{For}[j = 5, j < 6, j++,
\text{a}[[i, j, 1]] = \text{u}[[i, 1]] - \text{u}[[i, j]];
\text{a}[[i, j, 2]] = \text{u}[[i, 2]] - \text{u}[[i, j]];
\text{a}[[i, j, 3]] = \text{u}[[i, 3]] - \text{u}[[i, j]];
\text{a}[[i, j, 4]] = \text{u}[[i, 4]] - \text{u}[[i, j]];
\text{b}[[i, j, 1]] = \text{v}[[i, 1]] - \text{v}[[i, j]];
\text{b}[[i, j, 2]] = \text{v}[[i, 2]] - \text{v}[[i, j]];
\text{b}[[i, j, 3]] = \text{v}[[i, 3]] - \text{v}[[i, j]];
\text{b}[[i, j, 4]] = \text{v}[[i, 4]] - \text{v}[[i, j]];
\text{]};
\text{For}[i = 1, i < 4, i++, \text{For}[j = 5, j < 6, j++,
\text{T}[[i, 1]] = (\text{a}[[i, 5, 3]] \text{b}[[i, 5, 4]] - \text{a}[[i, 5, 4]] \text{b}[[i, 5, 3]])
(\text{a}[[i, 6, 1]] \text{b}[[i, 6, 2]] - \text{a}[[i, 6, 2]] \text{b}[[i, 6, 1]]);
\text{T}[[i, 2]] = (\text{a}[[i, 5, 4]] \text{b}[[i, 5, 2]] - \text{a}[[i, 5, 2]] \text{b}[[i, 5, 4]])
(\text{a}[[i, 6, 1]] \text{b}[[i, 6, 3]] - \text{a}[[i, 6, 3]] \text{b}[[i, 6, 1]]);
\text{T}[[i, 3]] = (\text{a}[[i, 5, 2]] \text{b}[[i, 5, 3]] - \text{a}[[i, 5, 3]] \text{b}[[i, 5, 2]])
(\text{a}[[i, 6, 1]] \text{b}[[i, 6, 4]] - \text{a}[[i, 6, 4]] \text{b}[[i, 6, 1]]);
\text{T}[[i, 4]] = (\text{a}[[i, 5, 1]] \text{b}[[i, 5, 4]] - \text{a}[[i, 5, 4]] \text{b}[[i, 5, 1]])
(\text{a}[[i, 6, 2]] \text{b}[[i, 6, 3]] - \text{a}[[i, 6, 3]] \text{b}[[i, 6, 2]]);
\text{T}[[i, 5]] = (\text{a}[[i, 5, 3]] \text{b}[[i, 5, 1]] - \text{a}[[i, 5, 1]] \text{b}[[i, 5, 3]])
(\text{a}[[i, 6, 2]] \text{b}[[i, 6, 4]] - \text{a}[[i, 6, 4]] \text{b}[[i, 6, 2]]);
\text{T}[[i, 6]] = (\text{a}[[i, 5, 1]] \text{b}[[i, 5, 2]] - \text{a}[[i, 5, 2]] \text{b}[[i, 5, 1]])
(\text{a}[[i, 6, 3]] \text{b}[[i, 6, 4]] - \text{a}[[i, 6, 4]] \text{b}[[i, 6, 3]]);
\text{]};
\text{II} = -\text{Det}\{\text{T}[[1, 1]], \text{T}[[1, 4]] + \text{T}[[1, 5]], \text{T}[[1, 3]], \text{T}[[1, 6]]\},
\text{T}[[2, 1]], \text{T}[[2, 4]] + \text{T}[[2, 5]], \text{T}[[2, 3]], \text{T}[[2, 6]]\},
\text{T}[[3, 1]], \text{T}[[3, 4]] + \text{T}[[3, 5]], \text{T}[[3, 3]], \text{T}[[3, 6]]\},
\text{T}[[4, 1]], \text{T}[[4, 4]] + \text{T}[[4, 5]], \text{T}[[4, 3]], \text{T}[[4, 6]]\}\)/\text{Det}\{\text{T}[[1, 1]], \text{T}[[1, 4]] + \text{T}[[1, 5]], \text{T}[[1, 3]], \text{T}[[1, 6]]\},
\text{T}[[2, 1]], \text{T}[[2, 4]] + \text{T}[[2, 5]], \text{T}[[2, 3]], \text{T}[[2, 6]]\},
\text{T}[[3, 1]], \text{T}[[3, 4]] + \text{T}[[3, 5]], \text{T}[[3, 3]], \text{T}[[3, 6]]\},
\text{T}[[4, 1]], \text{T}[[4, 4]] + \text{T}[[4, 5]], \text{T}[[4, 3]], \text{T}[[4, 6]]\}];
\text{ Algorithm 1: Continued.}
Since $I_1$, $I_2$, and $I_3$ are nonzero, we have

$$
\begin{vmatrix}
  t_1 + t_5 I_3 & t_1 + t_6 I_3 & t_1 + t_4 I_3 \\
  t_2 + t_5 I_3 & t_2 + t_6 I_3 & t_2 + t_4 I_3 \\
  t_4 + t_5 I_3 & t_4 + t_6 I_3 & t_4 + t_6 I_3 \\
  t_5 + t_6 I_3 & t_5 + t_6 I_3 & t_5 + t_6 I_3 \\
\end{vmatrix} = 0.
$$

(47)

Applying constraints (39) to (47), we obtain

$$
\begin{vmatrix}
  (t_5 + t_6) (I_3 - 1) & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  (t_5 + t_6) (I_3 - 1) & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  (t_5 + t_6) (I_3 - 1) & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  (t_5 + t_6) (I_3 - 1) & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
\end{vmatrix} = 0.
$$

(48)

Then the unique solution of $I_3$ is

$$
I_3 = -\frac{\begin{vmatrix}
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
\end{vmatrix}}{\begin{vmatrix}
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
  t_5 + t_6 I_3 & t_2 + t_6 I_3 & t_3 + t_6 I_3 \\
\end{vmatrix}}.
$$

(49)

### 4. Implementation of the Algorithm

We have validated the proposed method on the *mathematica* platform. The implementation is very simple. The code is given in Algorithm 1.

### 5. Conclusions

We have presented a direct and linear method for computing projective invariants of six 3D points from four 3D to 2D projection images. It can be used in 3D point pattern recognition from 2D images directly. Traditional methods for solving this problem are nonlinear and very complicated to use in real applications. The proposed formulas are clear and easy to implement by ordinary users. Another feature of our method is that we compute the projective invariants using only the original data. It is noticed that transformations of the original data can amplify the noise level of the data. This study provides a deeper understanding of the structure and motion problem. It seems that the natural configuration of the projective reconstruction problem is six points and four images.

Future directions of research include using this method in iterative or minimization schemas to solve the projective reconstruction problem with noising data, missing data, or outliers. It is also possible to develop similar methods for the cases of seven points in three images and eight points in two images.
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant nos. 61225012 and 71325002; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant no. 2012004130003; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 2011004210024; the Fundamental Research Funds for the Central Universities under Grant nos. N110204003 and N120104001.

References

[1] O. Faugeras, “What can be seen in three dimensions with an uncalibrated stereo rig?” in Proceedings of the 2nd European Conference on Computer Vision, pp. 563–578, Santa Margherita Ligure, Italy, May 1992.
[2] R. Hartley, “Estimation of relative camera positions for uncalibrated cameras,” in Proceedings of the 2nd European Conference on Computer Vision, pp. 579–587, Santa Margherita Ligure, Italy, May 1992.
[3] R. I. Hartley and P. Sturm, “Triangulation,” Computer Vision and Image Understanding, vol. 68, no. 2, pp. 146–157, 1997.
[4] A. Heyden, R. Berthilsson, and G. Sparr, “An iterative factorization method for projective structure and motion from image sequences,” Image and Vision Computing, vol. 17, no. 13, pp. 981–991, 1999.
[5] H. C. Longuet-higgins, “A computer algorithm for reconstructing a scene from two projections,” Nature, vol. 293, no. 10, pp. 133–135, 1981.
[6] Q. Luong and O. D. Faugeras, “The fundamental matrix: theory, algorithms, and stability analysis,” International Journal of Computer Vision, vol. 17, no. 1, pp. 43–76, 1996.
[7] V. Matiuakas and D. Miniotas, “Point cloud merging for complete 3D surface reconstruction,” Elektronika ir Elektrotechnika, no. 7, pp. 73–76, 2011.
[8] J. Oliensis and R. Hartley, "Iterative extensions of the sturm/triggs algorithm: convergence and nonconvergence," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2217–2233, 2007.
[9] L. Quan, “Invariants of six points and projective reconstruction from three uncalibrated images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 1, pp. 34–46, 1995.
[10] P. Sturm and W. Triggs, “A factorization based algorithm for multi-image projective structure and motion,” in Proceedings of the European Conference on Computer Vision, pp. 709–720, 1996.
[11] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” International Journal of Computer Vision, vol. 9, no. 2, pp. 137–154, 1992.
[12] Y. Wang, B. Zhang, and P. Hou, “Projective reconstruction of seven 3D points from two uncalibrated images,” in Proceedings of the 2nd International Conference on Future Computer and Communication (ICPCC '10), vol. 1, pp. V1659–V1663, May 2010.
[13] Y. Wang, B. Zhang, and F. Hou, “The projective invariants of six 3D points from three 2D uncalibrated images,” in Proceedings of the 3rd International Joint Conference on Computational Sciences and Optimization (CSO '10), vol. 1, pp. 486–487, May 2010.
[14] Y. B. Wang, B. Zhang, and T. S. Yao, “Projective invariants of co-moments of 2D images,” Pattern Recognition, vol. 43, no. 10, pp. 3233–3242, 2010.
[15] Y. B. Wang, X. W. Wang, B. Zhang, and Y. Wang, “A novel form of affine moment invariants of grayscale images,” Electronics and Electrical Engineering, vol. 19, no. 1, pp. 77–82, 2013.
[16] Y. B. Wang, X. W. Wang, and B. Zhang, “Affine differential invariants of functions on the plane,” Journal of Applied Mathematics, vol. 2013, Article ID 868725, 9 pages, 2013.
[17] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” Artificial Intelligence, vol. 78, no. 1-2, pp. 87–119, 1995.