In this paper, a new geometric structure of projective invariants is proposed. Compared with the traditional invariant calculation method based on 3D reconstruction, this method is comparable in the reliability of invariant calculation. According to this method, the only thing needed to find out is the geometric relationship between 3D points and 2D points, and the invariant can be obtained by using a single frame image. In the method based on 3D reconstruction, the basic matrix of two images is estimated first, and then, the 3D projective invariants are calculated according to the basic matrix. Therefore, in terms of algorithm complexity, the method proposed in this paper is superior to the traditional method. In this paper, we also study the projection transformation from a 3D point to a 2D point in space. According to this relationship, the geometric invariant relationships of other point structures can be easily derived, which have important applications in model-based object recognition. At the same time, the experimental results show that the eight-point structure invariants proposed in this paper can effectively describe the essential characteristics of the 3D structure of the target, without the influence of view, scaling, lighting, and other link factors, and have good stability and reliability.

1. Introduction

3D object recognition is one of the important contents of computer vision research, but the object imaging will be affected by the shooting angle, illumination, camera parameters, and other factors, which will make the same object in different shooting conditions get different images, leading to recognition difficulties [1]. Therefore, it is very important to find an invariant feature from the object which is not affected by these external factors. With the development of recognition technology, people not only find this invariant feature, geometric invariant [2–5], but also give some application examples. For example, Wei and Qiu studied the recognition method of a man-made object with a geometric structure line segment feature [6], Cao and Yan et al. studied deep learning with geometric invariance, [7], and Ding and Feng et al. studied the geometric correction of 3D human face with geometric invariance [8]. The geometric invariants were obtained by Cao et al. from another point of view by using the geometric algebra method [9–12]. At present, although many geometric invariants have been proposed by researchers and applied to 3D object recognition, face recognition, medical image recognition, and other fields, these invariants are still difficult to effectively eliminate the adverse effects caused by the difference of object parameters [13–16].

The recognition of 3D objects by a computer is to describe the features of the object by extracting the points of interest in the image and to recognize the object according to these features. Because the point features generally exist in any image, it is easy to extract, and the derivation and calculation of invariants based on point features is relatively simple, so it has been favored by more and more researchers. At the same time, the use of a single frame image to calculate
its invariant has also been a widespread concern and promotion [17–19], but the previous calculation of the point, line, curve feature invariant requires that the extracted elements are coplanar, but the reality is that most of the elements obtained are not in a plane; thus, the previous methods have great limitations.

In view of this kind of present situation, based on the theory of Conformal Geometric Algebra (CGA), this paper proposes an eight-point geometric structure invariant in three-dimensional space, which has the following advantages.

First of all, because the invariant calculation mainly relies on the extracted point features and line features, which are commonly found in any image, the extraction method is relatively mature. Compared with the five, six, and seven-point structure invariants, the calculation of invariants based on point features and line features is relatively accurate and easy; second, the calculation of the invariant only needs a single frame image, which avoids the calculation of the fundamental matrix between multiple images and camera calibration, compared with the invariant obtained by using multiple frames, and is simple and convenient; finally, the invariants of other geometric structures can be easily deduced by using the algorithm proposed in this paper, and it is also beneficial to find more invariants of unknown structures, which shows that the invariant algorithm proposed in this paper has good generality and simplicity. At the same time, the simulation and real experiments also prove that the invariants extracted in this paper are not affected by the camera viewpoint and other factors and basically remain unchanged in different perspectives, which has good reliability and stability.

2. The Projective Transformation

Relationship between 3D Space Points and Two-Dimensional Plane Points

Geometric invariant: Let \( G \) be a transformation group, \( g \) is an element of \( G \) and a geometric entity \( F(x, p) \) of \( S \) in a space, \( x, p \) stand for the coordinate parameter vector and the vector, respectively, and \( I(p) \) is the number of algebras with respect to \( F(x, p) \) and only related to the parameter vector \( p \). When \( g \) acts on \( F(x, p) \), there are

\[
g: F(x, p) \mapsto F(x', p').
\]

\( x', p' \) are the transformed coordinate vector and parameter vector, respectively, and \( I(p') \) is the number of generations with respect to \( F(x', p') \) and only with respect to the parameter vector \( p \); if present,

\[
I(p) = h(g)I(p').
\]

Then, we show that \( I(p) \) is an invariant of the geometric body \( F(x, p) \) under the transformation group, where \( h(g) \) is a quantity which depends on \( g \) only and is independent of the coordinates and parameters, and \( I(p) \) is an absolute invariant if \( h(g) \) equals (3).

Euclidean transformation, projective transformation, similar transformation, and affine transformation are all typical transformations in mathematics, which can be reduced to the transformation group in mathematics.

Let a point \( \overline{M} = (X, Y, Z)^T \) in 3D space project to a corresponding point \( \overline{m} = (x, y)^T \) through a projective centre \( O \), and let \( \overline{M} = (X, Y, Z, 1)^T \) and \( \overline{m} = (x, y, 1)^T \) be the coordinates of \( \overline{M} \) and \( \overline{m} \), respectively, in which case the relationship between the 3D coordinate \( M \) and the 2D coordinate \( m \) can be described by a projective matrix \( \lambda I \), which is

\[
\lambda \begin{bmatrix} x' \\ y' \\ 1 \\ 1 \end{bmatrix} = I \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},
\]

where \( \lambda \) is a scale factor. \( I \) is a \( 3 \times 4 \) projective matrix, related to parameters such as cameras.

Projective transformation is the key content of graphic transformation, and it is a bridge connecting 3D objects with 2D graphics. The projective transformation generally involves the projective matrix, but the projective matrix has high computational complexity and poor real-time performance and the results are easily affected by image noise. This problem has not been solved well yet. In order to avoid the specific calculation process of the projective matrix and to more conveniently represent the relationship between space points and their projection points, the following transformation is performed.

First, change \( Z \) in the spatial coordinate point \( M = (X, Y, Z, 1)^T \) to 0. We set \( E_T \) to the rotation and translational operator in the Euclidean transformation, by projecting the \( i \)-th point \( M_i = (X_i, Y_i, Z_i, 1)^T \) on the space plane \( \Gamma \) onto the image plane \( XY \) to derive \( m_i \) from (3).

\[
IM_i^Γ = I \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = I \begin{bmatrix} 1 & 0 & c_1 & 0 & X_i \\ 0 & 1 & c_2 & 0 & Y_i \\ 0 & 0 & c_3 & 1 & Z_i \\ 0 & 0 & c_4 & 1 & 1 \end{bmatrix}.
\]

We set

\[
E_T^{-1} = \begin{bmatrix} 1 & 0 & c_1 & 0 \\ 0 & 1 & c_2 & 0 \\ 0 & 0 & c_3 & Z \\ 0 & 0 & c_4 & 1 \end{bmatrix}.
\]

Then, (3) changes into

\[
\lambda m_i = IE_T^{-1}M_i^Γ,
\]

where \( E_T \) is a \( 4 \times 4 \) matrix and \( c_1, c_2, c_3, c_4 \) are any constants, which are converted from \( E_T \). Since the third row \( Z \) of \( M_i^Γ \) is 0, the third column of \( IE_T^{-1} \) that can be ignored makes equation (6) simpler:

\[
\lambda m_i = T_i k_i,
\]
where the $3 \times 3$ matrix $T_f$ is obtained by removing the third column of $IE_{13}^3$ and $k_f^t = (X^t_f, Y^t_f, 1)^T$ is obtained by removing the third row of $M_f^t$.

For any point in space, the projected image point can be described by (7). On this basis, we can get the invariants of many spatial point structures.

2.1. Invariants of Eight-Point Geometry on Two Adjacent Planes in Space. In this paper, the relationship between the 3D point and its corresponding 2D plane projection point is obtained. Using this relationship, this paper proposes a new invariant of the geometric structure composed of points, that is, the geometric structure of eight points on two adjacent planes in space, as shown in Figure 1. Here, points $A$, $B$, $C$, $D$, and $E$ are located on a plane $\Phi$, any three points of the five points are not collinear, points $D$, $E$, $F$, $G$, and $H$ are located on the plane $\Psi$, likewise, any three of the five points are not collinear, and their corresponding 2D plane projection points are $A' \sim H'$, respectively; then, by using equation (7), we get the following series of relationships.

For $A$, $B$, $C$, $D$, and $E$ on the plane $\Phi$, we get

$$
\lambda_A m_{A'} = T_\Phi k_A^\Phi,
\lambda_B m_{B'} = T_\Phi k_B^\Phi,
\lambda_C m_{C'} = T_\Phi k_C^\Phi,
\lambda_D m_{D'} = T_\Phi k_D^\Phi,
\lambda_E m_{E'} = T_\Phi k_E^\Phi.
$$

We write points $A$, $B$, $D$; $A$, $B$, $E$; $A$, $C$, $D$; and $A$, $C$, $E$, respectively, in a matrix form to get

$$
\begin{bmatrix}
\lambda_A m_{A'} & \lambda_B m_{B'} & \lambda_D m_{D'} \\
\lambda_A m_{A'} & \lambda_B m_{B'} & \lambda_E m_{E'} \\
\lambda_A m_{A'} & \lambda_C m_{C'} & \lambda_D m_{D'} \\
\lambda_A m_{A'} & \lambda_C m_{C'} & \lambda_E m_{E'}
\end{bmatrix}
= T_\Phi
\begin{bmatrix}
k_A^\Phi & k_B^\Phi & k_D^\Phi \\
k_A^\Phi & k_B^\Phi & k_E^\Phi \\
k_A^\Phi & k_C^\Phi & k_D^\Phi \\
k_A^\Phi & k_C^\Phi & k_E^\Phi
\end{bmatrix}.
$$

(8)

The determinant is obtained from both sides of the equation:

$$
\begin{align*}
\lambda_A \lambda_B \lambda_D | m_{A'} m_{B'} m_{D'} | &= |T_\Phi||k_A^\Phi k_B^\Phi k_D^\Phi|,  \\
\lambda_A \lambda_B \lambda_E | m_{A'} m_{B'} m_{E'} | &= |T_\Phi||k_A^\Phi k_B^\Phi k_E^\Phi|,  \\
\lambda_A \lambda_C \lambda_D | m_{A'} m_{C'} m_{D'} | &= |T_\Phi||k_A^\Phi k_C^\Phi k_D^\Phi|,  \\
\lambda_A \lambda_C \lambda_E | m_{A'} m_{C'} m_{E'} | &= |T_\Phi||k_A^\Phi k_C^\Phi k_E^\Phi|.
\end{align*}
$$

(10)

Similarly, for $D$, $E$, $F$, $G$, $H$ on the plane $\Psi$, we have the relationship:

$$
\begin{align*}
\omega_D m_{D'} &= T_\Psi k_D^\Psi,  \\
\omega_E m_{E'} &= T_\Psi k_E^\Psi,  \\
\omega_F m_{F'} &= T_\Psi k_F^\Psi,  \\
\omega_G m_{G'} &= T_\Psi k_G^\Psi,  \\
\omega_H m_{H'} &= T_\Psi k_H^\Psi.
\end{align*}
$$

(13)

We write points $G, F, D; G, F, E; G, H, D$; and $G, H, E$, respectively, in a matrix form to get

$$
\begin{align*}
\omega_G m_{G'} &\omega_F m_{F'} \omega_D m_{D'} | = T_\Psi[k_G^\Psi k_F^\Psi k_D^\Psi],  \\
\omega_G m_{G'} &\omega_F m_{F'} \omega_E m_{E'} | = T_\Psi[k_G^\Psi k_F^\Psi k_E^\Psi],  \\
\omega_G m_{G'} &\omega_H m_{H'} \omega_D m_{D'} | = T_\Psi[k_G^\Psi k_H^\Psi k_D^\Psi],  \\
\omega_G m_{G'} &\omega_H m_{H'} \omega_E m_{E'} | = T_\Psi[k_G^\Psi k_H^\Psi k_E^\Psi].
\end{align*}
$$

(14)

The determinants on both sides are

$$
\begin{align*}
\omega_G \omega_F \omega_D | m_{G'} m_{F'} m_{D'} | &= |T_\Psi||k_G^\Psi k_F^\Psi k_D^\Psi|,  \\
\omega_G \omega_F \omega_E | m_{G'} m_{F'} m_{E'} | &= |T_\Psi||k_G^\Psi k_F^\Psi k_E^\Psi|,  \\
\omega_G \omega_H \omega_D | m_{G'} m_{H'} m_{D'} | &= |T_\Psi||k_G^\Psi k_H^\Psi k_D^\Psi|,  \\
\omega_G \omega_H \omega_E | m_{G'} m_{H'} m_{E'} | &= |T_\Psi||k_G^\Psi k_H^\Psi k_E^\Psi|.
\end{align*}
$$

(15)–(18)

If we set

$$
Z_{ijk} = |m_i^t m_j^t m_k^t|,  \\
k_{ijk} = |k_i^x k_j^x k_k^x|,
$$

(19)

where

$$
x = \Phi, \Psi,
$$

(20)

then (15)–(18) can be written in simpler forms as follows:
to the formula, the relationship of the projection points mentioned method, it is convenient to obtain the invariant the actual matching.

4 Mathematical Problems in Engineering

\[
\begin{align*}
\lambda_{A'B'D'} &= |T_a| K_{ABD}', \\
\lambda_{A'B'E} &= |T_a| K_{ABE}', \\
\lambda_{A'C'D'} &= |T_a| K_{ACD}', \\
\lambda_{A'C'E} &= |T_a| K_{ACE}', \\
\omega_{G',D'} &= |T_\varphi| K_{GFD}', \\
\omega_{G',E'} &= |T_\varphi| K_{GFE}', \\
\omega_{G',D'} &= |T_\varphi| K_{GHD}', \\
\omega_{G',E'} &= |T_\varphi| K_{GHE}'.
\end{align*}
\] (21)

Then, the variant of the eight-point structure should be

\[
I_{\text{point}} = \frac{Z_{A'B'D'} Z_{A'C'D'} Z_{G'D'} Z_{A'E'}}{Z_{A'B'E'} Z_{A'C'E'} Z_{G'E'} Z_{A'D'}}
\]

\[
= \frac{K_{GFD}' K_{GFE}' K_{GHD}'}{K_{ABD}' K_{ABE}' K_{ACD}' K_{ACE}'}
\]

(22)

The calculation of the invariant requires eight points. For the structure of the eight points, the flexibility is relatively large, and it is not easy to obtain the same projection structure, which leads to errors. So, it is highly applicable for the actual matching.

2.2. Comparison with Other Methods. Using the aforementioned method, it is convenient to obtain the invariant of the general position of the space at 5 o'clock and the invariant of the 7-point structure of the space proposed by Zhu et al. [18].

We use the algorithm of this paper to calculate the projection invariant of 5 points coplanar.

For a general position on the space plane \( \pi \), \( X_i \), \( i = 1, 2, \ldots, 5 \), and any 3 points are not collinear. According to the formula, the relationship of the projection points \( X_i \) can be obtained for each point as follows:

\[
\varepsilon_i m_i' = T_\varphi k_i^\varphi, \quad i = 1, 2, \ldots, 5.
\] (23)

We write points 1, 2, 4; 1, 2, 5; 1, 3, 4; 1, 3, 5 in a matrix form, respectively.

\[
\begin{align*}
[\varepsilon_1 m_1', \varepsilon_2 m_2', \varepsilon_4 m_4'] &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi], \\
[\varepsilon_1 m_1', \varepsilon_2 m_2', \varepsilon_5 m_5'] &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi], \\
[\varepsilon_1 m_1', \varepsilon_3 m_3', \varepsilon_4 m_4'] &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi], \\
[\varepsilon_1 m_1', \varepsilon_3 m_3', \varepsilon_5 m_5'] &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi].
\end{align*}
\] (24)

Taking the determinant on both sides of the aforementioned four equations, we get

\[
\begin{align*}
\varepsilon_1 \varepsilon_2 \varepsilon_4 m_1', m_2', m_4' &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi]. \\
\varepsilon_1 \varepsilon_2 \varepsilon_5 m_1', m_2', m_5' &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi], \\
\varepsilon_1 \varepsilon_3 \varepsilon_4 m_1', m_3', m_4' &= |T_\varphi| [k_1^\varphi k_2^\varphi k_3^\varphi].
\end{align*}
\] (25)

Similarly, we set

\[
Z_{i',j,k} = |m_i' m_j' m_k'|, \\
K_{i,j,k}' = |k_i^\varphi k_j^\varphi k_k^\varphi|.
\] (29)

where \( i, j, k = 1, 2, 3, 4, 5 \), and then, (25)~(28), can be written in simpler forms as follows:

\[
\begin{align*}
\varepsilon_1 \varepsilon_2 \varepsilon_4 Z_{1',2',4'} &= |T_\varphi| K_{124}', \\
\varepsilon_1 \varepsilon_2 \varepsilon_5 Z_{1',2',5'} &= |T_\varphi| K_{125}', \\
\varepsilon_1 \varepsilon_3 \varepsilon_4 Z_{1',3',4'} &= |T_\varphi| K_{134}', \\
\varepsilon_1 \varepsilon_3 \varepsilon_5 Z_{1',3',5'} &= |T_\varphi| K_{135}'.
\end{align*}
\] (30)

At this point, we can get the invariant of the 5 points as

\[
I_1 = \frac{Z_{i',j,k} Z_{i',j',k'}}{Z_{i',j',k} Z_{i',j,k'}} = \frac{K_{i,j,k}' K_{i,j',k'}}{K_{i,j',k'} K_{i,j,k'}}
\]

(31)

The traditional method is to use the cross ratio to calculate its invariant, also known as the cross ratio invariant. It can be defined by a dot column or by a wire harness. If any four points \( A_1, A_2, A_3, A_4 \) are on line three, the cross ratio \( \lambda \) can be defined as

\[
\lambda = \frac{(A_1 A_2) (A_3 A_4)}{(A_1 A_3) (A_2 A_4)} = \left[ A_1 \wedge A_2 \right] \left[ A_3 \wedge A_4 \right].
\] (32)

The ratio of the two secondary cross ratios is the geometric projection invariant of the four points of the collinear line:

\[
J(\lambda) = \frac{2\lambda^6 - 6\lambda^5 + 9\lambda^4 - 8\lambda^3 + 9\lambda^2 - 6\lambda + 2}{\lambda^6 - 3\lambda^5 + 3\lambda^4 - \lambda^3 - 3\lambda^2 - 3\lambda + 1}.
\] (33)

However, when the calculation ratio reaches 5, the calculation complexity is high. At the same time, when using the cross ratio to calculate the invariant, there will be cases where the mismatched point ratios are equal, resulting in inaccurate matching, and the invariants can be easily obtained by the method in this section, and the matching efficiency is relatively high.

We use the algorithm of this paper to calculate the invariants of the spatial six-point structure proposed by Zhu et al.

Zhu et al. designed a geometric structure that intersects two planes \( \alpha, \beta \) in 3D space, taking two points \( C, D, E, \) and \( F \) at the intersection line \( AB \) and beyond the intersection line on the two planes. The geometric structure of 6 points is shown in Figure 2.

For two groups of coplanar points \( A, B, C, D \) and \( A, B, E, F \), any three points in each group are not collinear, and their corresponding two-dimensional plane projection points are \( A', B', C', D', E', F' \), respectively. According to the theory of the previous section, the relationship between the points in the 3D space and their projection points can be written into the following form.
Figure 2: Geometry of six points in two adjacent planes.

For points on the plane $\alpha$ and their projection points $A', B, C, D'$ are

\[
\delta_A m_A' = T_A k_A', \\
\delta_B m_B' = T_A k_B', \\
\delta_C m_C' = T_A k_C', \\
\delta_D m_D' = T_A k_D'.
\]  

Writing the three points $A, C, D$ and the three points $B, C, D$ in a matrix, we get

\[
\begin{bmatrix}
\delta_A m_A \\
\delta_B m_B \\
\delta_C m_C \\
\delta_D m_D
\end{bmatrix}
= T_A \begin{bmatrix}
k_A^a & k_C^a & k_D^a \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
\begin{bmatrix}
\delta_A m_A' \\
\delta_B m_B' \\
\delta_C m_C' \\
\delta_D m_D'
\end{bmatrix}
= T_A \begin{bmatrix}
k_A' & k_C' & k_D' \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]  

Taking the determinant on both sides of the above-mentioned two equations, we get

\[
\delta_A \delta_B \delta_C \delta_D m_A m_B m_C m_D = |T_A| \begin{bmatrix}
k_A^a & k_C^a & k_D^a \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
\delta_A \delta_B \delta_C \delta_D m_A' m_B' m_C' m_D' = |T_A'| \begin{bmatrix}
k_A' & k_C' & k_D' \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]  

Similarly, for points $A, B, E, F$ on a plane $\beta$ and their projection points $A', B', E', F'$,

\[
y_A m_A' = T_B k_A', \\
y_B m_B' = T_B k_B', \\
y_E m_E' = T_B k_E', \\
y_F m_F' = T_B k_F'.
\]  

Writing the three points $A, E, F$ and the three points $B, E, F$ in a matrix, we get

\[
\begin{bmatrix}
y_A m_A \\
y_E m_E \\
y_F m_F
\end{bmatrix}
= T_B \begin{bmatrix}
k_A^\beta & k_E^\beta & k_F^\beta \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
\begin{bmatrix}
y_A m_A' \\
y_E m_E' \\
y_F m_F'
\end{bmatrix}
= T_B \begin{bmatrix}
k_A'^\beta & k_E'^\beta & k_F'^\beta \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]  

Taking the determinant on both sides of the above-mentioned two equations, we get

\[
y_A y_E y_F m_A m_E m_F = |T_B| \begin{bmatrix}
k_A^\beta & k_E^\beta & k_F^\beta \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
y_A y_E y_F m_A' m_E' m_F' = |T_B'| \begin{bmatrix}
k_A'^\beta & k_E'^\beta & k_F'^\beta \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}.
\]  

Similarly, we set $Z_{\chi j k} = [ m_{\chi} m_{j} m_{k} ]$, $K_{ijk} = [ k_{i}^\chi k_{j}^\chi k_{k}^\chi ]$, where $\chi = \alpha, \beta, i, j, k = A, B, C, D, E, F$, and then, (39)–(40) can be written in simpler forms as follows:

\[
\delta_A \delta_C \delta_D Z_{A' C' D'} = |T_A| K_{ACD}^a,
\]

\[
\delta_B \delta_C \delta_D Z_{B' C' D'} = |T_A| K_{BCD}^a,
\]

\[
\gamma_A \gamma_E \gamma_F Z_{A' E' F'} = |T_B| K_{AEF}^a,
\]

\[
\gamma_B \gamma_E \gamma_F Z_{B' E' F'} = |T_B| K_{BEF}^a.
\]  

So, we can get its invariants as

\[
I_1 = \frac{Z_{A'C'D'} Z_{B'C'D'} Z_{A'E'F'} Z_{B'E'F'}}{Z_{B'C'D'} Z_{A'E'F'}} = \frac{K_{ACD}^a K_{AEF}^a}{K_{BCD}^a K_{BEF}^a}.
\]  

The invariant is the same as the invariant proposed by Zhu et al., but it can be seen that Zhu et al. obtained by introducing an intersection of seven lines to obtain its invariant. The calculation method is more complicated, and the calculation process is relatively simple by using the algorithm in this section, and it is easy to extend to the invariants of other point structures in space.

2.3. Algorithm Design. Based on the abovementioned derivation, for the proposed eight-point invariant geometric structure, this section first gives the specific framework of the algorithm design. As shown in Figure 3, by using the 3D images obtained by using the 3D object and the camera, we can calculate the 3D invariant and 2D invariants and, then, compare and analyse the structures of the two invariants.

Since the invariant algorithm proposed in this paper is about the calculation of the point, the experiment must extract the feature of the point in the picture. The corner point is used as the feature point on the image and contains important information, which plays an important role in the understanding and analysis of the image. Here, we use the Harris and Piersol [19] corner detection algorithm to extract the points that need to be used in the experiment.

In addition, since the coplanar five points in the eight-point structure have any three points that are not collinear, the noncollinearity test needs to be performed first on the selected points. The so-called collinearity test means that, in a conformal geometric algebra, the straight line determined by any two points $x_1, x_2$ can be expressed as $l = x_1 \wedge x_2 \wedge e_{oo}$ and, then, for any point $x$, if $x$ is on a straight line, then the point satisfies the equation $x \wedge l = 0$, but in the actual image application, the processing is easy to produce errors, the three points that seem to be noncollinear may be collinear, so we choose to use the area formed by the three points to measure whether they are collinear. To set a very small threshold $\eta$, if the area meets the equation $|x \wedge l| = (x \wedge l)^T l = x \cdot l > \eta$, then the three points are not collinear [20].

3. Experimental Results and Analysis

3.1. Experimental Platform. Hardware: personal PC, Intel (R) T2400 processor, 1.83 GHz, 1.00 GB memory, and one camera, 8 MP.

Software platform: Windows XP operating system, 3D graphics software AutoCAD, and image processing platform software MATLAB 7.5.0 (R2007b).
3.2. Simulation Experiment. The simulated image is drawn in AutoCAD 3D space. In order to obtain a geometric structure of two adjacent planes and eight points, we construct a model as shown in Figure 4. In order to see the point structure model at a glance, we extracted the model. The wireframe structure is shown in Figure 5, and select 8 points A–H as our experimental points.

First, we calculate the 3D invariant value of the point model, that is, the true value. If you use the point shown in the Figure 5 to establish the coordinate system as the origin, you can use AutoCAD to get the coordinates of the eight points, as shown in Table 1 and, then, use the formula to calculate its 3D invariant. The 3D projective invariant $I_{3D}$ of the model is calculated to be 0.4135.

Second, in order to calculate the 2D projective invariants, we use the same camera to select the 12 frames of images acquired by the model under different viewpoints, rotation angles, and scaling, as shown in Figure 6, and, then, extract the 20 corner points in the subimage, as

| No. | Coordinate (X, Y, Z)       |
|-----|---------------------------|
| A   | (238.72, 0, 629.97)       |
| B   | (238.72, 860.24, 629.97)  |
| C   | (238.72, 459, 457.72)     |
| D   | (238.72, 0, 219.46)       |
| E   | (238.72, 1028.41, 219.46) |
| F   | (661.04, 0, 219.46)       |
| G   | (831.48, 1028.41, 219.46) |
| H   | (831.48, 189.86, 219.46)  |
| $I_{3D}$ | 0.4135               |
shown in Figure 7 by using the Harris corner detection algorithm. Also, their 2D invariants are, respectively, calculated.

After detecting the corner points, we extract the coordinates of the 8 points $A \sim H$ we need and, then, use the collinearity test described above to perform the noncollinearity test on the five points on the coplanar surface. Here, we select 12 frames in which the constraint is satisfied and calculate a 2D invariant of the 3D model. As shown in Table 2, $I_{2D}$ represents the projective invariant values calculated from (3)–(20). The last column represents the absolute value of the relative error of each invariant to the true value of the 3D invariant calculated above 0.4135. The penultimate row represents all 2D mean values of the invariant and the error of the obtained average relative to the true 3D value, and the last line is the standard deviation of the obtained invariant value. We also used the methods of Zhang and Mui [3] and Deng et al. [21] to calculate the invariants of the above-mentioned 12 graphs. The absolute values of the relative errors are listed in the third and fourth columns of the table below, as can be seen from the table. The method in this article has certain advantages.

The abovementioned table is more clearly expressed as shown in Figure 8.

It can be seen from Tables 1 and 2 and Figure 8 that as it adopts the simulation image, the 3D invariant calculated by the algorithm is 0.4135, and the average of the 2D invariant is 0.4176, that is, despite the different viewpoint positions of the camera and the changes of 3D model, the calculated invariant results are ideal, and the projective invariants remain basically unchanged in 3D objects and 2D images. At the same time, the same object remains basically unchanged under different viewpoints, and the error is small. This invariant property can facilitate target recognition and object matching in computer vision.

In order to verify the stability and accuracy of the eight-point geometric structure, the invariants of the eight-point geometric structure obtained in this paper are
Figure 7: Harris corner extraction map of the 12 images.

Table 2: 2D invariants of the eight-point geometric structure calculated in the simulation image experiment and comparison with other methods.

| Image serial number | 2D invariant value ($I_{2D}$) | Error rate (%) (Zhang and Mui [3]) | Error rate (%) (Deng et al. [21]) | Error rate (%) (OURS) |
|---------------------|-------------------------------|-------------------------------------|-----------------------------------|----------------------|
| 1                   | 0.4108                        | 0.70                                | 0.68                              | 0.65                 |
| 2                   | 0.4144                        | 0.31                                | 0.29                              | 0.22                 |
| 3                   | 0.4139                        | 0.12                                | 0.15                              | 0.09                 |
| 4                   | 0.4143                        | 0.23                                | 0.31                              | 0.19                 |
| 5                   | 0.4208                        | 0.25                                | 0.22                              | 0.18                 |
| 6                   | 0.4175                        | 1.31                                | 1.12                              | 0.97                 |
| 7                   | 0.4171                        | 0.91                                | 0.87                              | 0.87                 |
| 8                   | 0.4206                        | 1.83                                | 1.91                              | 1.72                 |
| 9                   | 0.4189                        | 1.52                                | 1.62                              | 1.31                 |
| 10                  | 0.4226                        | 2.49                                | 2.53                              | 2.20                 |
| 11                  | 0.4256                        | 2.98                                | 2.84                              | 2.93                 |
| 12                  | 0.4148                        | 0.58                                | 0.49                              | 0.31                 |
| Mean value          | 0.4176                        | 1.01                                | 1.00                              | 0.97                 |
| Standard deviation  | 0.0043                        |                                     |                                   |                      |
compared with those of six-point and seven-point geometric structures proposed by Zhu et al. The error percentage of the invariant of the six-point-seven-point geometric structure relative to the true value and the error percentage of the invariant of the eight-point geometric structure relative to the true value of the three-dimensional invariant in this experiment are calculated, respectively, to get Figure 9, in which the blue line represents the structure invariant error rate of the space six-point structure and the red line represents the structure invariant error rate of the space seven-point structure. The black line represents the spatial eight-point structure, the structure invariant, and the error rate. As can be seen from the figure, the accuracy of the eight-point invariant geometric structure proposed in this paper is slightly higher than that of the six-point and seven-point invariant structures, and the stability is also improved.

3.3. Real Image Experiment. In order to further verify the reliability of the geometric invariant structure proposed in this paper in real images, we use books to form two adjacent planes, which constitute a 3D scene and, then, select eight points on the plane to form the proposed the eight-point invariant structure, as shown in Figure 10. The figure is marked with 8 points, among which A, B, C, D, E are located in one plane and D, E, F, G, H on another adjacent plane. Also, D, E are located on the intersection of the two planes. In order to verify that the invariant structure is protectively invariant, 20 views are acquired from different positions and angles using the same camera, as shown in Figure 11. Then, the corner points of each image are extracted by using the Harris corner extraction, as shown in Figure 12, and finally, one 2D geometric invariant of each image is calculated using the extracted corner points.
Figure 12 shows a Harris corner extraction map of the 20 images.

It can be seen from the corner extraction diagram that the eight points required for the experiment are basically detected. Now, we use the extracted corner points to calculate the 2D invariant and compare them with the methods of Zhang and Mui [3] and Deng et al. [21]. The calculation results are shown in Table 3. The first column is the image number, the second column is the invariant value of the corresponding image, and the third, fourth, and fifth columns use, as in the work of Zhang and Mui [3] and Deng et al. [21], the error percentage of the invariant value of each image.
Figure 12: Harris corner extraction image.

Table 3: The invariants of the eight-point geometric structure calculated in the real image experiment and the comparison of different methods.

| Image serial number | 2D invariant value ($I_{2D}$) | Error rate (%) (Zhang and Mui [3]) | Error rate (%) (Deng et al. [21]) | Error rate (%) |
|---------------------|--------------------------------|------------------------------------|------------------------------------|---------------|
| 1                   | 6.4569                         | 1.1240                             | 0.9945                             | 0.9569        |
| 2                   | 6.3249                         | 1.2101                             | 1.1465                             | 1.1070        |
| 3                   | 6.4281                         | 0.6172                             | 0.5248                             | 0.5066        |
| 4                   | 6.3887                         | 0.1294                             | 0.1395                             | 0.1094        |
| 5                   | 6.5018                         | 1.7514                             | 1.6912                             | 1.6589        |
| 6                   | 6.3593                         | 0.6129                             | 0.6358                             | 0.5691        |
| 7                   | 6.4640                         | 1.0923                             | 1.0521                             | 1.0679        |
| 8                   | 6.3848                         | 0.2412                             | 0.1836                             | 0.1704        |
| 9                   | 6.3809                         | 0.3482                             | 0.2436                             | 0.2314        |
| 10                  | 6.1893                         | 3.8142                             | 3.4954                             | 3.2272        |
| 11                  | 6.3802                         | 0.2524                             | 0.2012                             | 0.2424        |
| 12                  | 6.3773                         | 0.3756                             | 0.3125                             | 0.2877        |
| 13                  | 6.4713                         | 1.2921                             | 1.2125                             | 1.1820        |
| 14                  | 6.4907                         | 1.5978                             | 1.4935                             | 1.4854        |
| 15                  | 6.4870                         | 2.0145                             | 2.0014                             | 1.4275        |
| 16                  | 6.3414                         | 0.9412                             | 0.9017                             | 0.8490        |
| 17                  | 6.3775                         | 0.3514                             | 0.3124                             | 0.2846        |
| 18                  | 6.3337                         | 0.9956                             | 0.8475                             | 0.9694        |
| 19                  | 6.3940                         | 0.1245                             | 0.0912                             | 0.0266        |
| 20                  | 6.3830                         | 0.5241                             | 0.3417                             | 0.1986        |
calculated by the method in this paper relative to the average value. The last line is the average value of the invariant results of the 20 images and the average value of the error.

Using the method in this paper to compare with the work of Zhang and Mui [3] and Deng et al. [21], it was seen that, except the error on the image 11, it was 18 points lower than the method proposed by Deng et al. [21], the error of other image invariant points is better than that in the work of Deng et al. [21], the error of the image was 11, and 18 points of this method is completely better than that of that in the work of Zhang and Mui [3], which proves that the method has a certain cutting edge.

As can be seen from Table 3, the invariant value is basically stable around 6.4, and the calculated results are generally ideal, which shows that the geometric structure of the eight-point projective invariant proposed in this paper remains basically unchanged under different viewpoints, and the error is small. However, there is still a problem in the experiment; when the view is scaled to a very small size, the corner cannot be detected accurately, resulting in a large error in the calculation of invariants, so it is very important to find a better structure of invariants or other corner extraction methods. In order to more clearly see the distribution of invariants, MATLAB is used to draw its invariant distribution, as shown in Figure 13; from the figure, it can be seen that invariant value distribution is flat, basically unchanged.

4. Conclusions

In this paper, we propose a new geometric structure of projective invariants. Compared with the traditional invariant calculation method based on 3D reconstruction, the reliability of the invariant calculation is comparable. Moreover, the method proposed in this paper only needs to find out the geometric relationship between 3D empty points and two-dimensional space points, and its invariants can be obtained by using a single frame image. In the method based on 3D reconstruction, the basic matrix of two images is estimated first, and then, the 3D projective invariants are calculated according to the basic matrix. Therefore, in terms of algorithm complexity, the method proposed in this paper is superior to the traditional method.

In addition, this paper studies the projection transformation relationship from a 3D point to a 2D point in space. According to this relationship, the geometric invariant relationships of other point structures can be easily derived, which have important applications in model-based object recognition. At the same time, the experimental results show that the eight-point structure invariant proposed in this paper can effectively describe the essential characteristics of the 3D structure of the target, is not affected by view, scaling, lighting, and other link factors, and has good stability and reliability.

Data Availability

The (PNG images) data used to support the findings of this study have not been made available because of privacy issues.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61871186 and 61771322 and the Zhejiang Natural Science Foundation under Grant no. LY14F030007.

References

[1] W. Cao, J. Yuan, Z. He, Z. Zhang, and Z. He, “Fast deep neural networks with knowledge guided training and predicted regions of interests for real-time video object detection,” IEEE Access, vol. 6, pp. 8990–8999, 2018.
[2] W. Cao, W. Feng, Q. Lin, G. Cao, and Z. He, “A review of hashing methods for multimodal retrieval,” IEEE Access, vol. 8, pp. 15377–15391, 2020.
[3] Y. Zhang and Y. Mui, “Invariants of space line element structure based on projective geometric algebra,” IEEE Access, vol. 8, pp. 62610–62618, 2020.
[4] Z. Chen and R. Zhao, “Geometric invariance and its applications to 3D object recognition,” in Proceedings of the Signal and Information Processing Technology-The First Joint Conference on Signal and Information Processing, pp. 145–153, Xi’an, China, 2002.
[5] C. Li, “A method for camera calibration based on invariant feature of projection transformation,” Mathematics in Practice and Theory, vol. 40, no. 10, pp. 124–130, 2010.
[6] H. Wei and Z. Qiu, “Line feature based man-made object recognition with invariance,” Chinese Journal of Computers, vol. 33, no. 6, pp. 1088–1099, 2010.
[7] W. Cao, Z. Yan, and Z. He, “A comprehensive survey on geometric deep learning,” IEEE Access, vol. 8, pp. 35929–35949, 2020.
[8] L. Ding, H. Feng, and Y. Huang, “3D face pose correction based on geometrical invariance by geometrical algebra theory,” Journal of Computer-Aided Design & Computer Graphics, vol. 36, no. 1, pp. 177–181, 2015.
[9] W. Cao, Q. Lin, Z. He, and Z. He, “Hybrid representation learning for cross-modal retrieval,” Neurocomputing, vol. 345, pp. 45–57, 2019.

[10] H. Li, “From geometric algebras to advanced invariant computing,” Journal of Systems Science and Mathematical Sciences, vol. 28, pp. 915–929, 2008.

[11] H. Li, “Conformal geometric algebra and algebraic manipulations of geometric invariants,” Journal of Computer-Aided Design & Computer Graph, vol. 18, no. 7, pp. 902–911, 2006.

[12] C. Xing, Research and Application on Feature Extraction Algorithms of Geometric Invariant, pp. 17–92, Chongqing University, Chongqing, China, 2015.

[13] J. YanC. Li et al., “Adaptive discrete hyper graph matching,” IEEE Transactions on Cybernetics, vol. 48, no. 2, pp. 765–779, 2017.

[14] R. Wang, M. Shen, T. Wang, and W. Cao, “L1-Norm minimization for multi-dimensional signals based on geometric algebra,” Advances in Applied Clifford Algebras, vol. 29, no. 2, pp. 1–18, 2019.

[15] M. Shen, R. Wang, and W. Cao, “Joint sparse representation model for multi-channel image based on reduced geometric algebra,” IEEE Access, vol. 6, pp. 24213–24223, 2018.

[16] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513, Washington, DC, USA, June 2004.

[17] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: speeded up robust features,” European Conference on computer Vision, vol. 3951, pp. 404–417, Springer, Berlin, Germany, 2006.

[18] H. Zhu, H. Shu, T. Xia, L. Luo, and J. L. Coatrieux, “Translation and scale invariants of Tchebichef moments,” Pattern Recognition, vol. 40, no. 9, pp. 2530–2542, 2007.

[19] C. M. Harris and A. G. Piersol, Harris’ Shock and Vibration Handbook, McGraw-Hill, New York, NY, USA, 1976.

[20] W. Cao and R. Wang, Geometric Algebra in Information Science, Science Press, Beijing, China, 2018.

[21] H. Deng, T. Birdal, and S. Ilic, “Ppf-Foldnet: unsupervised learning of rotation invariant 3D local descriptors,” in Proceedings of the European Conference on Computer Vision (ECCV), pp. 602–618, Munich, Germany, September 2018.