An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 42 (2001) 1465–1483
www.elsevier.nl/locate/camwa

Three-View Matching Algorithm for Multipolyhedron Reconstruction Using Genetic Algorithm

JIANN-DER LEE
Department of Electrical Engineering
Chang Gung University
Tao-Yuan, Taiwan 333, R.O.C.
jdlee@mail.cgu.edu.tw

(Received May 1999; revised and accepted May 2000)

Abstract—In this paper, a novel stereo matching algorithm based on genetic algorithm (GA) is proposed to find the correspondences among multipolyhedron objects in a three-view system. Optimal camera configuration in the approach is used to reduce the feature searching area on the image plane, and GA is then used to refine the matching results. Due to optimal camera configuration, we can achieve a smaller searching area, less ambiguity, and a faster matching process. The features on the second image near the intersection of the two epipolar lines derived from the first and the third images considered as possible triplets are refined using GA. The fitness function for every corresponding triplet is assigned according to the corresponding distance between the feature location and the intersection of two epipolar lines. The consistent connection relationship is then enforced in the evolution process to resolve the ambiguous correspondence triplets. Experimental results show that the 3-D multiple polyhedra in a complex scene can be successfully reconstructed in a three-view system.

Keywords—Multipolyhedron reconstruction, Three-view analysis, Genetic algorithm.

NOMENCLATURE

O-x-y-z	the world coordinate system	f	the focal length
O′-u-v-w	the camera coordinate system	R	the rotation matrix
O′-u′-v′	the image coordinate system	T	the translation matrix
(x_c,y_c,z_c)	the coordinates of the camera lens center in world coordinate system	λ	a scale factor defined as $\overline{O'P_n}/\overline{O'P'_n}$ where P_n is a point in 3-D space

1. INTRODUCTION

In the application of computer vision, it is an important subject to reconstruct three-dimensional objects from stereo images. Although active ranging methods are useful in controllable

The author wishes to thank the anonymous reviewers for their careful review and valuable suggestions to improve the paper.

0898-1221/01/$ - see front matter Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(01)00255-3

environments, passive stereo methods are more convenient and suitable for general machine vision applications, such as aircraft navigation, industrial automation, passive navigation, cartography, surveillance, etc., [1–4]. In general, 3-D reconstruction consists of four major steps, namely, camera calibration, feature extraction, stereo matching, and depth reconstruction. Once the camera configuration is established, the mapping parameters of the cameras referring to the global coordinate system can be obtained. After the images of the objects are taken, features on the image are extracted by employing a series of image processing techniques. Among the images, corresponding feature pairs or triplets are identified and then examined to find the stereo matching. Associated with the camera geometry, the matching results are then applied to compute the 3-D information of the scene for the depth reconstruction.

In the above-mentioned steps, the matching of stereo images is essential in identifying the corresponding points in two images that are taken from the same physical point in the 3-D space. Various hierarchical and global matching techniques [5–15] have been proposed to avoid ambiguous matching. Most of the matching schemes include two stages [16–18]. The purpose of the first stage is to reduce the matching computation. In this stage, the geometric constraints in physical dimensions and relative locations are used to decide the image region within which the corresponding features will be searched. Then the similarities of the local information of the features within the searching regions are exploited to pick out the potential correspondences. In the second stage, the topological constraints in connections between the features using relaxation or coarse-fine matching strategies are applied to eliminate the ambiguity.

In the past, Hung *et al.* [6,7] proposed an approach for 3-D single-polyhedron reconstruction by using three-view stereo analysis. In their approach, the "point corresponding constraint function" of a corner pair on two image planes is derived from the relationship between the global coordinates and the image coordinates of the corner pair. Hung used this function as a necessary condition for matching corner points for single-polyhedron reconstruction. Their method, however, cannot be applied to the problem of multipolyhedron reconstruction. Basically, reconstruction problems associated with the multipolyhedron scenes include the following three conditions:

(1) the feature points spread all over the image plane;
(2) many lines are missing in the images due to noise;
(3) false connections between corners will be detected due to corners or edge lines overlap on the image plane under perspective projections.

In order to avoid ambiguous matching in a three-view system, an efficient matching algorithm for multipolyhedron reconstruction with the aid of genetic algorithm (GA) is proposed. This scheme includes two stages: search region finding and corresponding point verification. In the stage of search region finding, based on optimal camera configuration developed in our previous study [13], geometric constraints in physical dimensions and relative locations are used to decide the image region within which the corresponding features will be searched. Then, the similarities of the local feature information within the searching regions are exploited to sort out the potential correspondences. More specifically speaking, for a corner pair from two input images, we define a corresponding corner candidate function (CCCF) as a necessary condition for finding its corresponding corner point in the third input image. In the stage of corresponding point verification, the genetic algorithm using the topological constraints in connections between the features is applied to eliminate the ambiguity of the matching results obtained in the stage of search region finding. The features used in the method are corner points and the edge connection information among the corners of the polyhedron. Figure 1 shows the flowchart of the proposed algorithms for multipolyhedron reconstruction.

The remainder of this paper is organized as follows. The proposed algorithm, including the derivation of the camera extrinsic parameters, the derivation of the CCCF, and stereo matching using genetic algorithm, are presented in Section 2. The preprocessing steps consisting of geomet-

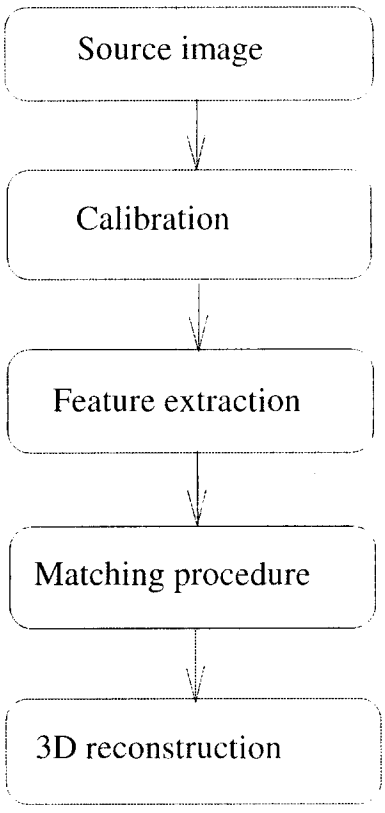


Figure 1. The flowchart of the proposed method.

rical correction, feature extraction, connection finding, and experimental results are illustrated in Section 3. Finally, discussions and conclusions are presented in Section 4.

2. THE PROPOSED ALGORITHM

In this approach, the corresponding corner candidates of a feature point in the first image are found by searching along the epipolar line in the second image. The epipolar line is derived by computing the intersection of the epipolar plane and the second image plane. For a three-view system, two epipolar lines can be derived on an image plane [6,7]. The feature points around the intersection of the two epipolar lines are regarded as the possible correspondences. The mathematical derivation of the proposed method is given below.

2.1. Derivation of the Camera Extrinsic Parameters

A general model of single-view camera geometry is shown in Figure 2. The camera parameters include intrinsic parameters (i.e., image center, focal length, and aspect ratio) and extrinsic parameters (i.e., rotation matrix and translation matrix). It is assumed that the intrinsic parameters of the camera have been obtained from previous experience, so the calibration task is to solve the extrinsic parameters for each camera. According to references [19,20], a 3-D point P with world coordinates and camera coordinates is related to its corresponding image point P′ with image coordinates by

$$(x, y, z)' = R \cdot (u', w', v')' + (x_c, y_c, z_c)', \tag{1}$$

$$(u', w', v') = \lambda(u, f, v), \tag{2}$$

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix},$$

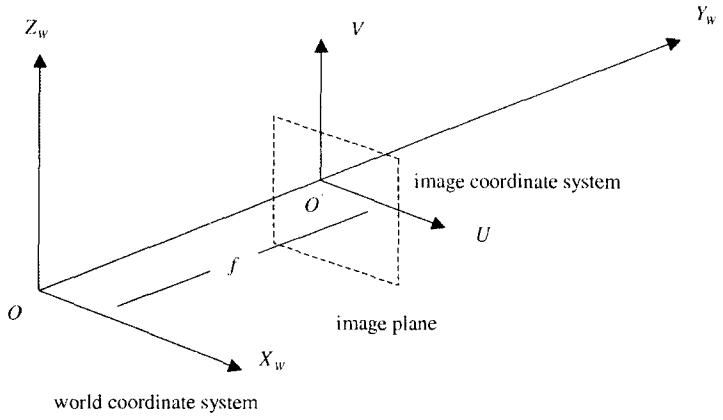


Figure 2. The camera viewing model used in this approach.

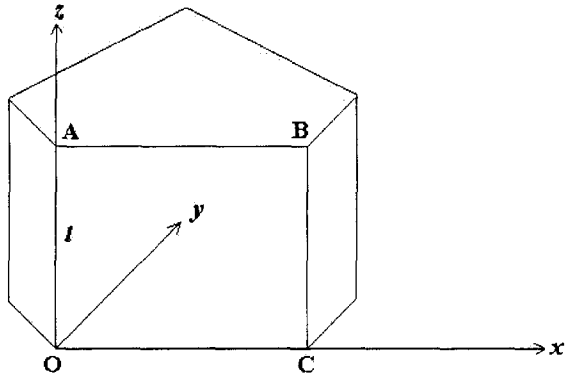


Figure 3. A regular prism used for calculating the camera extrinsic parameters.

where $\lambda = \overline{O'P}/\overline{O'P'}$ is a scale factor and R is the rotation matrix. From the geometric properties of the mark shown in Figure 3, we obtain

(1) $\overline{AB} = \overline{OC}$,
(2) $\overline{AO} = l$ (a known constant),
(3) $\overline{AB}$ is perpendicular to $\overline{AO}$.

Substituting the image coordinates of O, A, B, C into Properties 1–3, the focal length f, and the scale factors λ_O, λ_A, λ_B, λ_C can be derived sequentially by the following equations:

$$f = \left[\frac{(K_B u_B - K_A u_A)(K_O u_O - K_A u_A) + (K_B v_B - K_A v_A)(K_O v_O - K_A v_A)}{(K_A - K_B)(K_O - K_A)}\right]^{1/2}, \tag{3}$$

$$\lambda_O = \frac{l}{[(K_A u_A - u_O)^2 + (K_A f - f)^2 + (K_A v_A - v_O)^2]^{1/2}}, \tag{4}$$

$$\begin{aligned} \lambda_A &= K_A \lambda_O, \\ \lambda_B &= K_B \lambda_O, \\ \lambda_C &= K_C \lambda_O, \end{aligned} \tag{5}$$

where K_A, K_B, K_C are constants. From equation (2), the camera coordinates of points A, B, C, O, and the length $\overline{OC}$ (denoted as $\|\overline{OC}\|$) can be obtained. Let L_A, L_C, L_O represent the distance between points A, C, O, and the camera lens center O′, respectively. From the landmark

and the camera coordinate system, we have

$$L_{\mathrm{O}}^2 = \lambda_{\mathrm{O}}^2 \left(u_{\mathrm{O}}^2 + f^2 + v_{\mathrm{O}}^2\right) = x_{\mathrm{C}}^2 + y_{\mathrm{C}}^2 + z_{\mathrm{C}}^2, \tag{6}$$

$$L_{\mathrm{A}}^2 = \lambda_{\mathrm{A}}^2 \left(u_{\mathrm{A}}^2 + f^2 + v_{\mathrm{A}}^2\right) = x_{\mathrm{C}}^2 + y_{\mathrm{C}}^2 + (z_{\mathrm{C}} - l)^2, \tag{7}$$

$$L_{\mathrm{C}}^2 = \lambda_{\mathrm{C}}^2 \left(u_{\mathrm{C}}^2 + f^2 + v_{\mathrm{C}}^2\right) = \left(x_{\mathrm{C}} - \left\|\overline{\mathrm{OC}}\right\|\right)^2 + y_{\mathrm{C}}^2 + z_{\mathrm{C}}^2. \tag{8}$$

Rearranging the above equations, $(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}})$ can be derived as

$$z_{\mathrm{C}} = \frac{1}{2l}\left(L_{\mathrm{O}}^2 - L_{\mathrm{A}}^2 + l^2\right), \tag{9}$$

$$x_{\mathrm{C}} = \frac{1}{2\left\|\overline{\mathrm{OC}}\right\|}\left(L_{\mathrm{O}}^2 - L_{\mathrm{C}}^2 + \left\|\overline{\mathrm{OC}}\right\|^2\right), \tag{10}$$

$$y_{\mathrm{C}} = \left(L_{\mathrm{O}}^2 - x_{\mathrm{C}}^2 - z_{\mathrm{C}}^2\right)^{1/2}. \tag{11}$$

Once $(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}})$ is derived, substituting the coordinates of points A, B, C, O into equation (1), can solve the rotation matrix R.

2.2. Derivation of Corresponding Corner Candidate Function (CCCF)

Rearranging equations (1) and (2), we have

$$(x - x_{\mathrm{C}}, y - y_{\mathrm{C}}, z - z_{\mathrm{C}})^t = \lambda(\alpha, \beta, \gamma)^t, \tag{12}$$

with

$$(\alpha, \beta, \gamma)^t = R(u, f, v), \tag{13}$$

where $(x_{\mathrm{C}}, y_{\mathrm{C}}, z_{\mathrm{C}})$ is the location of the camera lens center in the global coordinate system. From equation (12), two rays of a 3-D point in a two-view system, which consists of Cameras 1 and 2, can be expressed as

$$\begin{aligned} x - x_{\mathrm{C1}} &= \lambda_1 \alpha_1, \\ y - y_{\mathrm{C1}} &= \lambda_1 \beta_1, \\ z - z_{\mathrm{C1}} &= \lambda_1 \gamma_1, \end{aligned} \tag{14}$$

and

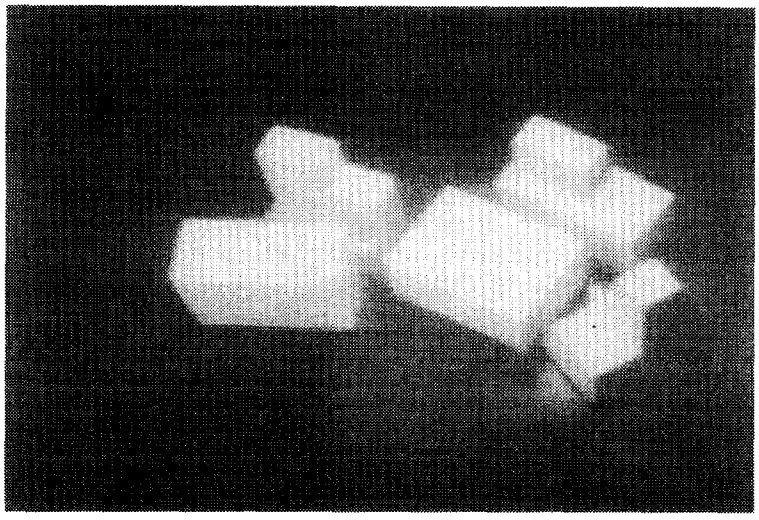
$$\begin{aligned} x - x_{\mathrm{C2}} &= \lambda_2 \alpha_2, \\ y - y_{\mathrm{C2}} &= \lambda_2 \beta_2, \\ z - z_{\mathrm{C2}} &= \lambda_2 \gamma_2, \end{aligned} \tag{15}$$

respectively. Under the assumption that the two cameras are located above the object, i.e., $z - z_{\mathrm{C1}} \neq 0$ and $z - z_{\mathrm{C2}} \neq 0$, we have $\lambda_1 = (z - z_{\mathrm{C1}})/\gamma_1$ and $\lambda_2 = (z - z_{\mathrm{C2}})/\gamma_2$. Thus, equation (15) can be rewritten as

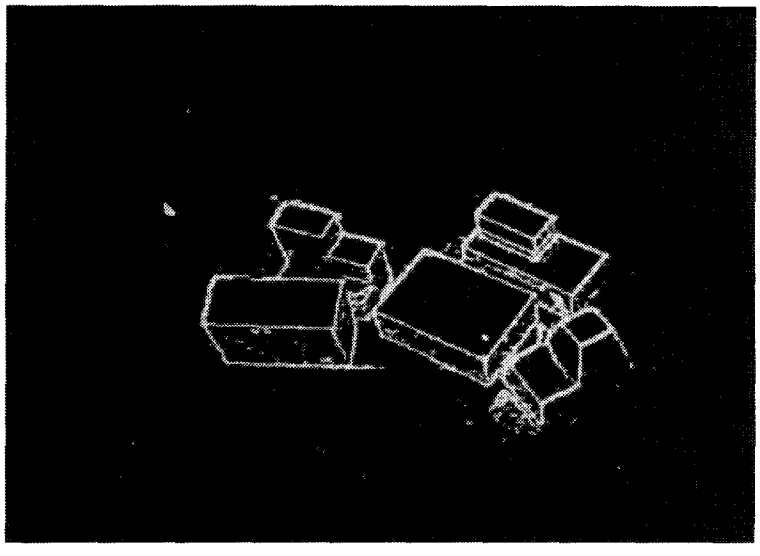
$$\begin{aligned} x - x_{\mathrm{C1}} &= \left(\frac{\alpha_1}{\gamma_1}\right)(z - z_{\mathrm{C1}}), \\ y - y_{\mathrm{C1}} &= \left(\frac{\beta_1}{\gamma_1}\right)(z - z_{\mathrm{C1}}), \end{aligned} \tag{16}$$

and

$$\begin{aligned} x - x_{\mathrm{C2}} &= \left(\frac{\alpha_2}{\gamma_2}\right)(z - z_{\mathrm{C2}}), \\ y - y_{\mathrm{C2}} &= \left(\frac{\beta_2}{\gamma_2}\right)(z - z_{\mathrm{C2}}). \end{aligned} \tag{17}$$



(a) An example source image.



(b) The resultant image of Figure 4a after thinning.

Figure 4.

Subtracting equation (16) from equation (17) and eliminating variable z, we obtain the CCCF as follows:

$$\begin{aligned}\text{CCCF}\left(u_i^a, v_i^a, u_j^b, v_j^b\right) &= (\beta_1\gamma_2 - \beta_2\gamma_1)\Delta x_{\text{C}} - (\alpha_1\gamma_2 - \alpha_2\gamma_1)\Delta y_{\text{C}} \\ &\quad +(\alpha_1\beta_2 - \alpha_2\beta_1)\Delta z_{\text{C}} = 0,\end{aligned} \tag{18}$$

where $\Delta x_{\text{C}} = x_{\text{C2}} - x_{\text{C1}}$, $\Delta y_{\text{C}} = y_{\text{C2}} - y_{\text{C1}}$, and $\Delta z_{\text{C}} = z_{\text{C2}} - z_{\text{C1}}$. In (u_i^a, v_i^a), the superscript 'a' denotes View 1 and the subscript 'i' denotes the sequence number. Assume (u_i^a, v_i^a) is a fixed feature point (i.e., u_i^a and v_i^a are constant) in View 1, the epipolar line (u_i^a, v_i^a) in View 2 can be expressed as

$$\text{CCCF}\left(u_j^b, v_j^b\right) = (\beta_1\gamma_2 - \beta_2\gamma_1)\Delta x_{\text{C}} - (\alpha_1\gamma_2 - \alpha_2\gamma_1)\Delta y_{\text{C}} + (\alpha_1\beta_2 - \alpha_2\beta_1)\Delta z_{\text{C}} = 0. \tag{19}$$

Associated with feature points (u_i^a, v_i^a) in Image 1 and (u_k^c, v_k^c) in Image 3, the two epipolar lines on Image 2 for a three-view system are given by

$$\begin{aligned}\text{CCCF}\left(u_i^a, v_i^a, u_j^b, v_j^b\right) &= (\beta_1\gamma_2 - \beta_2\gamma_1)\Delta x_{\text{C1}} - (\alpha_1\gamma_2 - \alpha_2\gamma_1)\Delta y_{\text{C1}} \\ &\quad +(\alpha_1\beta_2 - \alpha_2\beta_1)\Delta z_{\text{C1}} = 0\end{aligned} \tag{20}$$

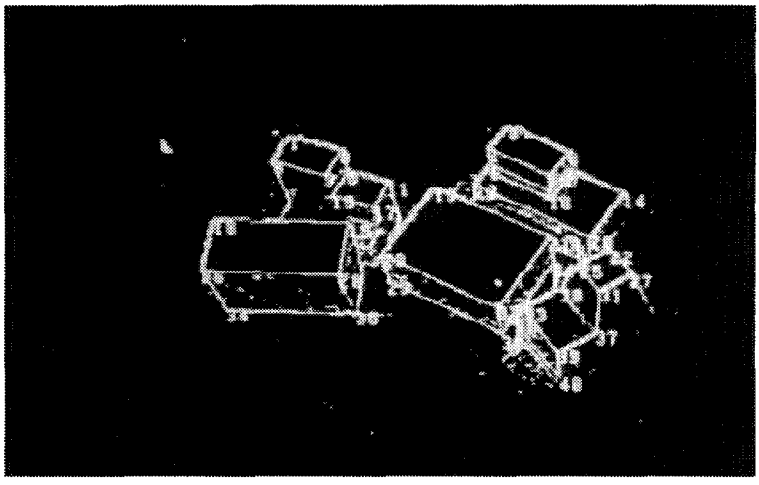
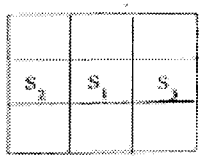
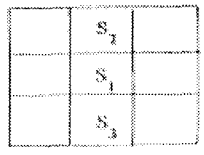


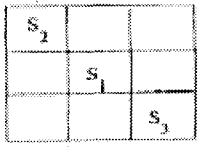
Figure 5. The edge image of Figure 4 after corner extraction.



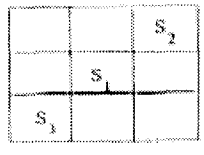
(a) Scanning Model 1.



(b) Scanning Model 2.



(c) Scanning Model 3.



(d) Scanning Model 4.

Figure 6. The scanning models.

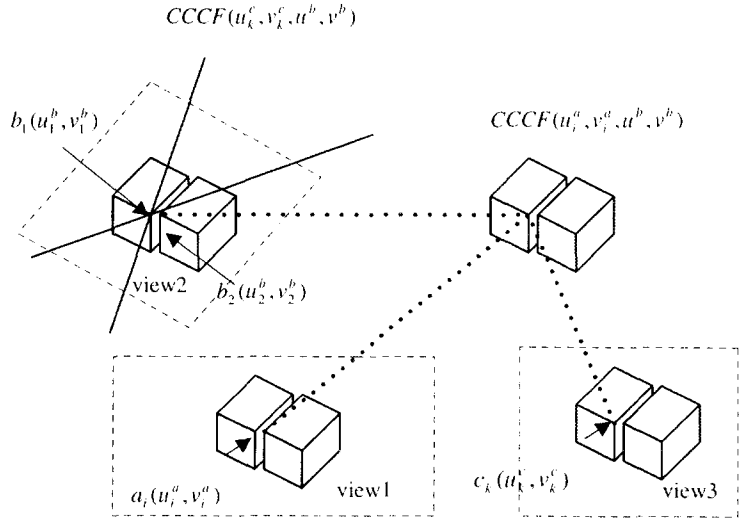


Figure 7. Illustration of constructing possible triplets.

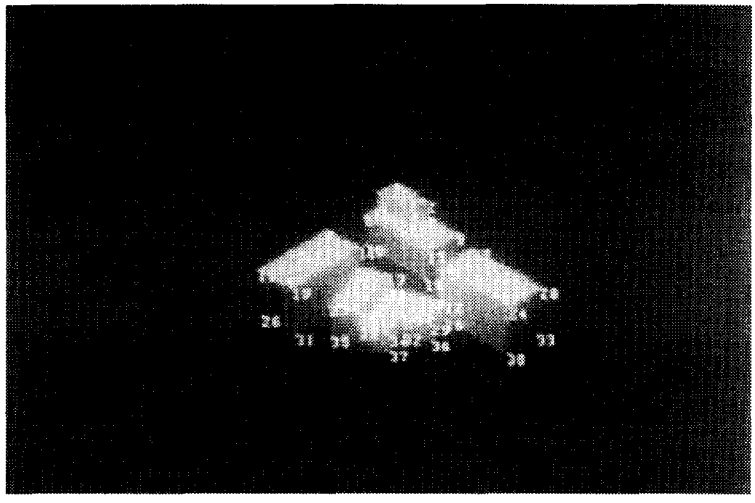
Table 1. Node connection information of the sample image.

Node	Connected Node
0	1,2
1	0,3,4
2	0,4,6
3	1,5,7
4	1,2,7
5	3,13,10
6	2,7,9
7	3,4,6
8	11,16
9	6,13
10	5,18
11	19,8
12	15,20
13	5,9,18
14	17,23
15	12
16	8,19,26
17	14,21
18	13,10
19	11,16,31
20	12,24,33
21	17,30,22
22	28,21
23	14,25,35
24	20,38,15
25	17,23,30
26	16,31
27	29,34,32
28	22,36
29	22,27,36
30	25,34
31	19,26
32	27,36,37
33	20,38
34	37,27,30
35	23,37
36	28,32
37	32,35
38	33

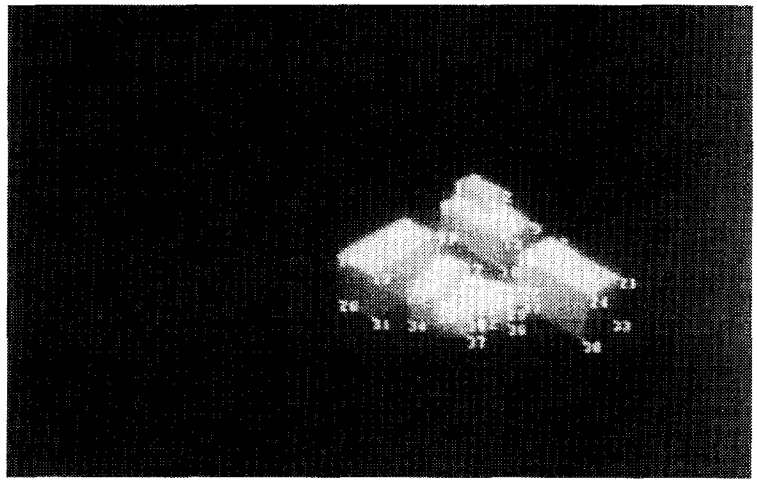
and

$$\begin{aligned} \mathrm{CCCF}\left(u_k^c, v_k^c, u_j^b, v_j^b\right) = (\beta_3\gamma_2 - \beta_2\gamma_3)\Delta x_{C3} - (\alpha_3\gamma_2 - \alpha_2\gamma_3)\Delta y_{C3} \\ +(\alpha_3\beta_2 - \alpha_2\beta_3)\Delta z_{C3} = 0, \end{aligned} \tag{21}$$

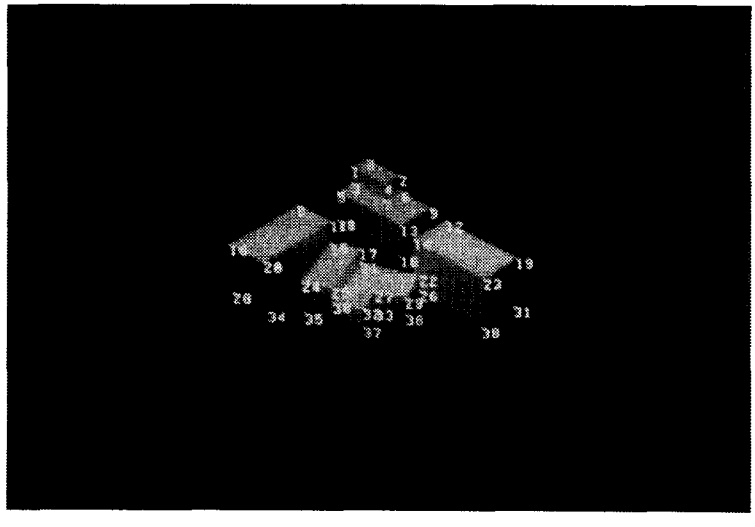
respectively, where $\Delta x_{C1} = x_{C2} - x_{C1}$, $\Delta y_{C1} = y_{C2} - y_{C1}$, $\Delta z_{C1} = z_{C2} - z_{C1}$, $\Delta x_{C3} = x_{C2} - x_{C3}$, $\Delta y_{C3} = y_{C2} - y_{C3}$, and $\Delta z_{C3} = z_{C2} - z_{C3}$. It is noted that the parameters required for calculating CCCF can be obtained from camera calibration described in previous section. In other words,



(a) The source image taken by Camera 1.



(b) The source image taken by Camera 2.



(c) The source image taken by Camera 3.

Figure 8.

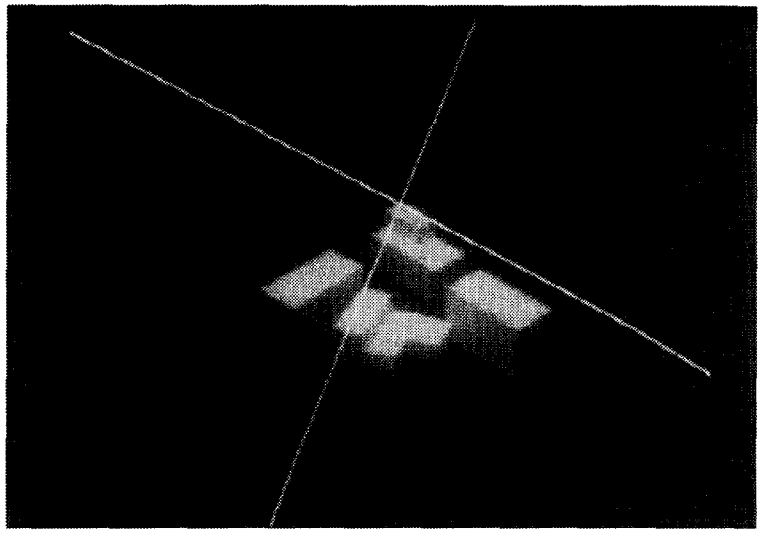


Figure 9. The two epipolar lines on View 2.

from equations (12)–(14), we have

$$\begin{aligned}
\alpha_1 &= R^a_{11}\times u^a_i + R^a_{12}\times f + R^a_{13}\times v^a_i, & \beta_1 &= R^a_{21}\times u^a_i + R^a_{22}\times f + R^a_{23}\times v^a_i,\\
\gamma_1 &= R^a_{31}\times u^a_i + R^a_{32}\times f + R^a_{33}\times v^a_i, & \alpha_2 &= R^b_{11}\times u^b_j + R^b_{12}\times f + R^b_{13}\times v^b_j,\\
\beta_2 &= R^b_{21}\times u^b_j + R^b_{22}\times f + R^b_{23}\times v^b_j, & \gamma_2 &= R^b_{31}\times u^b_j + R^b_{32}\times f + R^b_{33}\times v^b_j,\\
\alpha_3 &= R^c_{11}\times u^c_k + R^c_{12}\times f + R^c_{13}\times v^c_k, & \beta_3 &= R^c_{21}\times u^c_k + R^c_{22}\times f + R^c_{23}\times v^c_k,\\
&& \gamma_3 &= R^c_{31}\times u^c_k + R^c_{32}\times f + R^c_{33}\times v^c_k.
\end{aligned}$$

Simplify equations (20) and (21), the intersection of two epipolar lines in View 2 can be expressed by

$$u^b_j = \frac{\mathrm{A_1B_2 - A_2B_1}}{\mathrm{C_1B_2 - C_2B_1}}, \qquad v^b_j = \frac{\mathrm{A_1C_2 - A_2C_1}}{\mathrm{C_1B_2 - C_2B_1}}, \tag{22}$$

where

$$\begin{aligned}
\mathrm{A_1} &= \Delta x_{\mathrm{C1}}\left(R^b_{22}\gamma_1 - R^b_{32}\beta_1\right) - \Delta y_{\mathrm{C1}}\left(R^b_{12}\gamma_1 - R^b_{32}\alpha_1\right) + \Delta z_{\mathrm{C1}} f\left(R^b_{12}\beta_1 - R^b_{22}\alpha_1\right),\\
\mathrm{A_2} &= \Delta x_{\mathrm{C3}}\left(R^b_{22}\gamma_3 - R^b_{32}\beta_3\right) - \Delta y_{\mathrm{C3}}\left(R^b_{12}\gamma_3 - R^b_{32}\alpha_3\right) + \Delta z_{\mathrm{C3}} f\left(R^b_{12}\beta_3 - R^b_{22}\alpha_3\right),\\
\mathrm{B_1} &= \Delta x_{\mathrm{C1}}\left(R^b_{23}\gamma_1 - R^b_{33}\beta_1\right) - \Delta y_{\mathrm{C1}}\left(R^b_{13}\gamma_1 - R^b_{33}\alpha_1\right) + \Delta z_{\mathrm{C1}}\left(R^b_{13}\beta_1 - R^b_{23}\alpha_1\right),\\
\mathrm{B_2} &= \Delta x_{\mathrm{C3}}\left(R^b_{23}\gamma_3 - R^b_{33}\beta_3\right) - \Delta y_{\mathrm{C3}}\left(R^b_{13}\gamma_3 - R^b_{33}\alpha_3\right) + \Delta z_{\mathrm{C3}}\left(R^b_{13}\beta_3 - R^b_{23}\alpha_3\right),\\
\mathrm{C_1} &= \Delta x_{\mathrm{C1}}\left(R^b_{21}\gamma_1 - R^b_{31}\beta_1\right) - \Delta y_{\mathrm{C1}}\left(R^b_{11}\gamma_1 - R^b_{31}\alpha_1\right) + \Delta z_{\mathrm{C1}}\left(R^b_{11}\beta_1 - R^b_{21}\alpha_1\right),\\
\mathrm{C_2} &= \Delta x_{\mathrm{C3}}\left(R^b_{21}\gamma_3 - R^b_{31}\beta_3\right) - \Delta y_{\mathrm{C3}}\left(R^b_{11}\gamma_3 - R^b_{31}\alpha_3\right) + \Delta z_{\mathrm{C3}}\left(R^b_{11}\beta_3 - R^b_{21}\alpha_3\right).
\end{aligned}$$

From equation (16), it is clear that if (u^a_i, v^a_i) and (u^c_k, v^c_k) form a correspondence pair, then (u^b_j, v^b_j), (u^a_i, v^a_i), and (u^c_k, v^c_k) form a corresponding triplet. Due to image quantization error, measurement error of the camera parameters and derivation of the corner points exist in practical application, the CCCF values of two corresponding corner points are usually not equal to zero. However, using the criterion of $|\mathrm{CCCF}| \leq Th$, where Th is a preset value, we can find all candidates of corresponding corner triplets among the three images. The real triplets are finally determined by GAs using the topological constraints in connections, i.e., triplet connection force (TCF) between each detected corner point.

2.3. Genetic Algorithm for Stereo Matching

Genetic algorithms [21–23] are stochastic searching algorithms based on the natural selection and natural genetics. Genetic algorithms, differing from traditional search techniques, start with an initial set of random solutions called population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. In other words, each chromosome is coded as a bit-string. The chromosome evolves through successive iterations, called generations. During each generation, the chromosomes are evaluated using some measure of fitness. To create the next generation, new chromosomes, called offspring, are formed by either

(1) merging two chromosomes from current generation using a crossover operator, or
(2) modifying a chromosome using a mutation operator.

A new generation is formed by

(1) selecting some of the parents and offspring, according to the fitness values, and
(2) rejecting others so as to keep the population size constant.

The fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosomes, which represents the optimum or suboptimal solution to the problem.

In short, the first step in applying the genetic algorithm is the creation of the initial population. In each iteration, a new population of the same size is generated from the current population using genetic operations, (i.e., crossover and mutation) and evolution operation (i.e., selection). In this work, for each image i, we let m be the number of the detected corner point, and $C^i = (C_1^i, C_2^i, \ldots, C_m^i)^t$, $i = 1, \ldots, 3$, be the index vector of these corner points. Thus, each gene is represented by a binary strings of length 3*MAX(N_1, N_2, N_3) bits. If the bit has a value 1, then the corresponding corner point has a triplet candidate and its index in other two images are recorded. Otherwise, it is excluded from the gene to indicate no possible triplet exist. For example, let $N_1 = 3$, $N_2 = 4$, $N_3 = 3$ and the gene is 1011100011. It means that the first and third corner points in Image 1, the first and second corner points in Image 2, and the second and third corner points in Image 3 have triplet candidates. In order to evaluate the importance of various terms in TCF, three types of fitness functions are proposed for a given chromosomes chr. These fitness functions are expressed as below, respectively.

TYPE-1.

$$\text{Fit(chr)} = \text{MIN} \sum_{a,b=0}^{m} \left| \text{TCF}^1\left(\text{C}_\text{A}^2\right) - \text{TCF}^1\left(\text{C}_\text{B}^i\right) \right|, \qquad i = 1, 3, \tag{23}$$

where $\text{TCF}^1(\text{C}_\text{A}^i) = \|\overline{\text{C}_{a-1}^i \text{C}_\text{A}^i}\| + \|\overline{\text{C}_\text{A}^i \text{C}_{a+1}^i}\|$ and (C_{a-1}^i and C_{a+1}^i) is the neighboring two corner points of C_A^i. MIN is a minimum operator.

TYPE-2.

$$\text{Fit(chr)} = \text{MIN} \sum_{a,b=0}^{m} \left| \text{TCF}^2\left(\text{C}_\text{A}^2\right) - \text{TCF}^2\left(\text{C}_\text{B}^i\right) \right|, \qquad i = 1, 3, \tag{24}$$

where $\text{TCF}^2(\text{C}_\text{A}^i) = \angle(\text{C}_{a-1}^i \text{C}_\text{A}^i \text{C}_{a+1}^i)$, and $\angle()$ denotes the interior angles between these corner points (i.e., C_A^i, C_{a-1}^i, and C_{a+1}^i).

TYPE-3.

$$\text{Fit(chr)} = \text{MIN} \sum_{a,b=1}^{m} \left| \text{TCF}^3\left(\text{C}_\text{A}^2\right) - \text{TCF}^3\left(\text{C}_\text{B}^i\right) \right|, \qquad i = 1, 3, \tag{25}$$

where $\text{TCF}^3(\text{C}_\text{A}^i) = K_1(\|\overline{\text{C}_{a-1}^i \text{C}_\text{A}^i}\| + \|\overline{\text{C}_\text{A}^i \text{C}_{a+1}^i}\|) + K_2 \angle(\text{C}_{a-1}^i \text{C}_\text{A}^i \text{C}_{a+1}^i)$, K_1, and K_2 are weighting factors.

Table 2. The initial matching results before using GA.

Image 1	Image 2	Image 3
0	0	0
1	2	1
2	2	3
3	4	1
4	4	4
5	5	5
6	6	6
7	8	8
8	9	9
9	10	9
10	10	10
11	11	11
12	13	14
13	11	12
14	16	15
15	17	14
16	15	14
17	17	17
18	18	18
19	19	22
20	21	22
21	20	19
22	22	23
23	23	22
24	24	23
25	26	25
26	26	28
27	28	29
28	27	27
29	29	29
30	31	30
31	30	34
32	32	34
33	33	31
34	35	36
35	33	32
36	36	36
37	37	37
38	37	35

Table 3. The final matching results using Type-1 fitness function.

Image 1	Image 2	Image 3
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	16	16
15	14	15 ·
16	15	14
17	17	17
18	18	18
19	19	20
20	21	21
21	20*	19*
22	22	22
23	23	24*
24	24	23*
25	25	25
26	26	28
27	28	26*
28	27	27*
29	29	29
30	30	30
31	31	34
32	32	33
33	33	31
34	35	35
35	34	32
36	36	36
37	37	37
38	38	38

*: Error Matching

Table 4. The final matching results using Type-2 fitness function.

Image 1	Image 2	Image 3
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14*	16	16
15	14*	15
16	15	14
17	17	17
18	18	18
19	19	20*
20	21	21*
21	20	19
22	22	22
23	23	24
24	24	23
25	25	25
26	26	28
27	28	26
28	27	27
29	29	29
30	30	30
31	31	34
32	32	33
33	33	31
34	35	35
35	34	32
36	36	36
37	37	37
38	38	38

*: Error Matching

Table 5. The final matching results using Type-3 fitness function.

Image 1	Image 2	Image 3
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	15	15
15	16	14
16	14	16
17	17	17
18	18	18
19	19	20
20	21	21
21	20	19
22	22	22
23	23	24
24	24	23
25	25	25
26	26	28
27	28	26
28	27	26
29	29	29
30	30	30
31	31	34
32	32	33
33	33	31
34	35	32
35	34	35
36	36	36
37	37	37
38	38	38

*: Error Matching

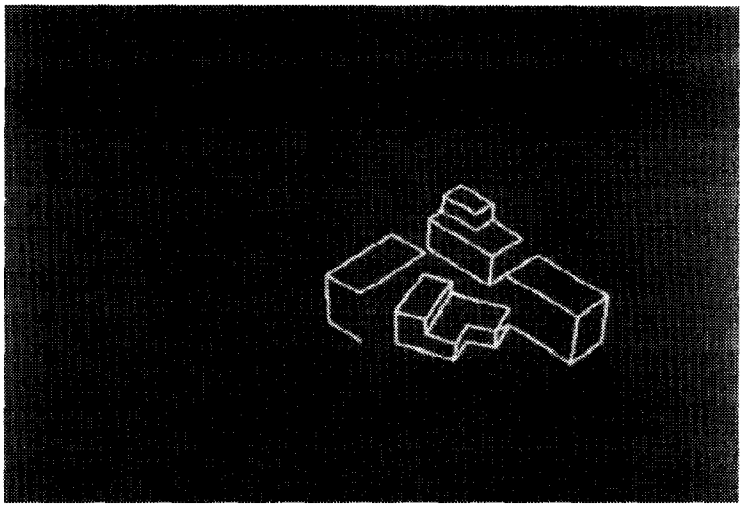


Figure 10. The reconstructed image of Figure 8.

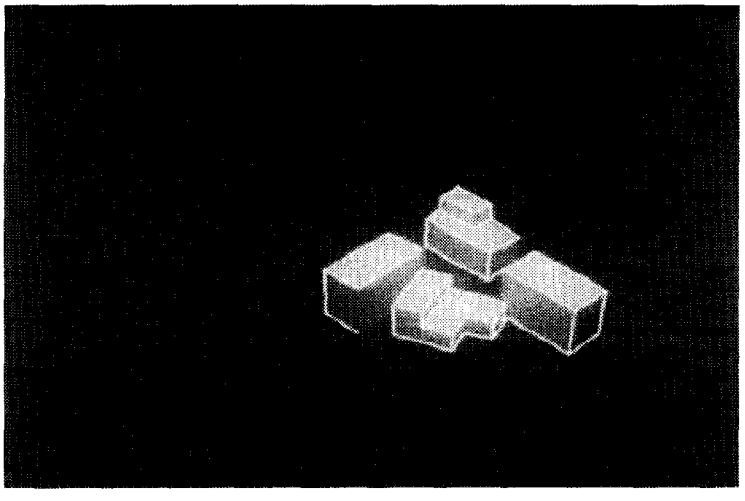
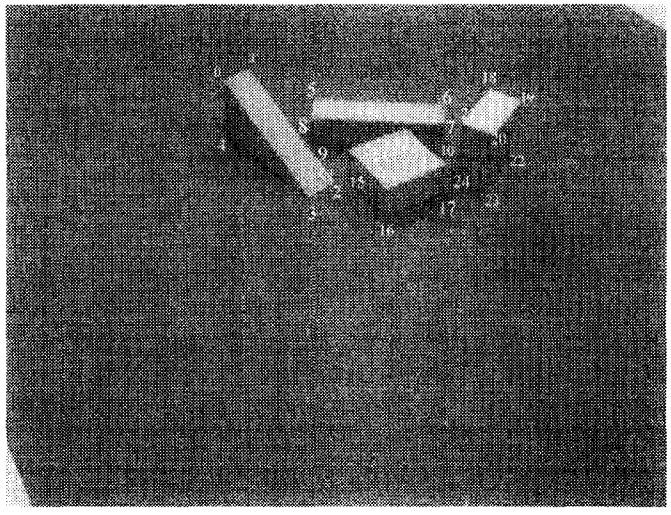


Figure 11. The overlay image of Figures 8 and 10.

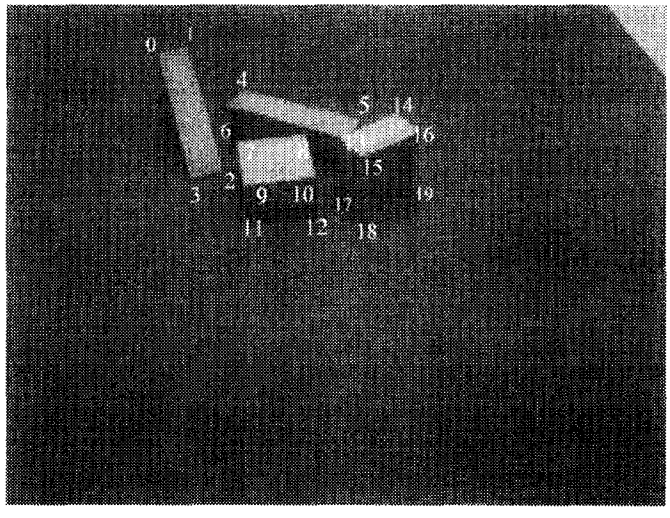
Since the selection probability (SP) of a particular chromosome is proportional to the fitness value, it is expressed as $\mathrm{SP} = \mathrm{Fit(chr)}/\sum_{\mathrm{chr}=1}^{N} \mathrm{Fit(chr)}$, where N is the population size. We have chosen one point crossover operator, and a mutation operator with elitist strategy as in the simple genetic algorithm given in [23]. By iterative running of the evolution cycle, chromosome strings are updated and the performance is gradually improved. The cycle terminates when all chromosome strings no longer change, i.e., the chromosome strings converge.

3. IMAGE PREPROCESSING AND EXPERIMENTAL RESULTS

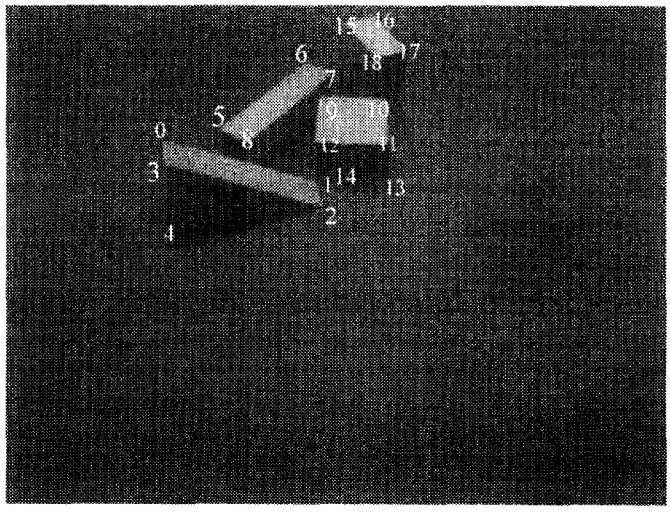
A three-views system has been built to reconstruct multipolyhedron by using the proposed approach. Under an optimal camera configuration [18], three CCD cameras first captured three images of a test sample (a scene of multipolyhedron). All the cameras are set at the same half space, and they are well calibrated using a mark. The optical axis of every camera is toward the center of the polyhedra which are put together on the table and the working environment is controlled to form black background. In this approach, the desired features for matching and reconstruction are corner points and edge lines. Preprocessing consists of geometrical correction and feature extraction is employed to extract those features. The details of these procedures are described as follows.



(a) The source image taken by Camera 1 in Experiment 2.



(b) The source image taken by Camera 2 in Experiment 2.



(c) The source image taken by Camera 3 in Experiment 2.

Figure 12.

Table 6. Node connection information of the first image used in Experiment 2.

Node	Connected Node
0	1,3,4
1	0,2
2	1,3
3	0,2,4
4	0,3
5	6,8
6	5,7
7	6,8,10
8	5,7,9
9	8,10
10	7,9
11	12,14
12	11,13,15
13	14,15,16
14	11,13,17
15	12,16
16	15,17
17	14,16
18	19,21
19	18,20
20	19,21,23
21	18,20,24
22	19,23
23	22,24
24	21,23

Table 7. The initial matching results before using GA.

Image 1	Image 2	Image 3
0	4	0
1	1	3
2	6	1
3	3	2
4	0	3
5	4	8
6	13	6
7	5	7
8	6	6
9	8	7
10	13	18
11	5	9
12	8	9
13	9	13
14	10	10
15	7	12
16	11	13
17	10	10
18	14	15
19	16	16
20	15	12
21	13	18
22	19	10
23	18	13
24	18	1

GEOMETRICAL CORRECTION. Due to the limitations of optical imaging or electric scanning systems in the picture-taking process, the scale of the picture increases or decreases with the distance from the image center. This leads to some types of geometrical distortion. To remove the geometrical distortion [24], a linear transformation function is used to describe the mapping between the grid points or landmarks (i.e., the ideal picture) and the grabbed picture (i.e., the distorted image). The gray level at each pixel of the grabbed picture is corrected by using a bilinear interpolation technique [24].

FEATURE EXTRACTION. To extract local features, many researchers used the *a priori* knowledge about the objects to extract the line drawings of polyhedron [25,26]. These algorithms first use some clustering process to fit sets of straight lines in possible positions of edge lines, then heuristically search for the junction in the areas nearby these lines, and last build up the line drawings. Here, we proposed a fast method to extract the desired features from noisy data. The method includes edge detection, thinning, corner extraction, and edge line extraction. Figures 4a and 4b show an example source picture and the resultant pictures, respectively, after thinning. After obtaining a thinned picture, we use a corner finder CF $= |\Delta\phi|/mg$ to detect the desired corner points [7], where ϕ is the derivative function of an image and mg is the mean of the gradient magnitude of the thinned points within a window. If CF $\geq \delta$ at a position, where δ is a

Table 8. The final matching results using Type-1 fitness function in Experiment 2.

Image 1	Image 2	Image 3
0	0	0*
1	1	-*
2	2	14*
3	3	2
4	-	-*
5	4	5
6	5	6
7	13	7*
8	6	8
9	-	-
10	-	-
11	8	9
12	7	12
13	9	11
14	10	10
15	-	1*
16	11	13
17	12	-
18	14	16
19	16	17
20	15	18
21	13	15
22	19	-
23	17	-*
24	18	-*

*: Error Matching

-: No Correspondence

Table 9. The final matching results using Type-2 fitness function in Experiment 2.

Image 1	Image 2	Image 3
0	0	3
1	1	0
2	2	14*
3	3	2
4	-	-*
5	4	5
6	5	6
7	-	7
8	6	8
9	-	-
10	-	-
11	8	9
12	7	12
13	9	11
14	10	10
15	-	14
16	11	13
17	12	-
18	14	16
19	16	17
20	15	18
21	13	15
22	18	-*
23	19	-*
24	17	-

*: Error Matching

-: No Correspondence

preset value, then the position is detected as a corner. The window size used is 5*5. According to the slope of the edge in Figure 5, four scanning modes (see Figure 6) are employed to find the local connection information of these edges. For each mode, we check whether node 1 (i.e., corner point 1) is connected with node 2 or not. Figure 7 illustrates how to construct the possible triplets. In other words, in this trinocular system, we form two crossing epipolar lines on View 2. One of the epipolar lines is derived from a node in View 1 by equation (20), and the other is derived from a node in View 3 by equation (21). From equation (22), we obtain the intersection of the two epipolar lines for each node in View 2. The node connection information of the sample image is listed in Table 1.

The system proposed is implemented on an IBM PC/586 microcomputer, and the software is coded in C language. To evaluate the feasibility of our system, several experiments with multipolyhedron scenes with different arrangements have been carried out.

To reduce the construction error, we employ a voting method to examine the connection information after the matching procedure. In the voting method, each connection between two

Table 10. The final matching results using Type-3 fitness function in Experiment 2.

Image 1	Image 2	Image 3
0	0	3
1	1	0
2	2	1
3	3	2
4	-	-*
5	4	5
6	5	6
7	-	7
8	6	8
9	-	-
10	-	-
11	8	9
12	7	12
13	9	11
14	10	10
15	-	14
16	11	13
17	12	-
18	14	16
19	16	17
20	15	18
21	13	15
22	19	-
23	18	-
24	17	-

*: Error Matching

-: No Correspondence

corners must be confirmed at least by two images. Therefore, missing (or false) lines in one of the three images can be tolerated. After feature extraction and matching, the objects are partially reconstructed using the knowledge of the corner correspondences and the connection information among the three images.

In the experiment, the three source images of multipolyhedron grabbed by Camera i, $i = 1, \ldots, 3$, are shown in Figures 8a–8c. The sequence number of nodes is superimposed on the images. Figure 9 illustrates the two epipolar lines on View 2. The initial matching results before using GA are shown in Table 2. Next, Tables 3–5 show the final matched results using three mentioned above fitness functions. It is noted that all the nodes among three views are correctly matched using Type-3 fitness function, but one node is missed using Type-1 and Type-2 fitness functions. Obviously, both length of triplet and its interior angle are important factors for evaluating the real triplet among three views. Figure 10 shows the partial reconstruction from Figure 8. Figure 11 shows the overlay image from Figures 8 and 10. The processing time for stereo matching procedure using genetic algorithm took 1.9 sec. Besides, as shown in Figure 12, three images grabbed by the camera with large orientation difference are used in Experiment 2. The node connection information of the first image is listed in Table 6. Table 7 illustrates the initial matching results before using GA. The final matching results for this experiment are listed in Tables 8–10. As we expected, the approach using Type-3 fitness function is superior to the

others. In summary, it is clear that the performance of the proposed method can be significantly improved by GA with the integration of information of length and angle of a triplet.

4. CONCLUSIONS

In this paper, a new stereo matching method for polyhedra reconstruction in a three-view system is proposed. To resolve the ambiguity of the false targets (multiple matches), genetic algorithm is employed to evaluate the local connection information between the nodes. The possible triplets with the highest fitness are chosen as the exact correspondence. The approach is fast, flexible, and stable. It does not require precise information about feature location and connection information. These properties render it superior performance as compared to other previous methods.

REFERENCES

1. C. Goad, Special purpose automatic programming for 3-D model-based vision, In *Proc. Image Understanding Workshop*, pp. 94–104, (1983).
2. D. Gennery, Object detection and measurement using stereo vision, In *Proc. Image Understanding Workshop*, pp. 41–47, (1979).
3. H. Baker, *Depth from Edge and Intensity Based Stereo*, AIM 347, Stanford, (1982).
4. H. Henderson, R. Miller and C. Grosch, Automatic stereo reconstruction of man made targets: Digital processing of aerial images, *Proc. Soc. Photo-Opt. Instrum. Engr.* **186** (6), 240–248, (1979).
5. M. Ito and A. Ishii, Three-view stereo analysis, *IEEE Trans. PAMI* **31**, 2–18, (1986).
6. R.N. Chiou, K.C. Hung, C.N. Shyi, J.Y. Lee and C.H. Chen, A new three-view matching strategy for polyhedron reconstruction, In *Proc. ICS'88*, pp. 1516–1521, (1988).
7. K.C. Hung, R.N. Chiou, C.N. Shyi, J.Y. Lee and C.H. Chen, Polyhedron reconstruction using three-view analysis, *Pattern Recognition* **22** (3), 231–246, (1989).
8. G. Medioni and R. Nevatia, Segment-based stereo matching, *CVGIP* **31**, 2–18, (1985).
9. Y. Ohta and T. Kanade, Stereo by intra- and inter-scanline search using dynamic programming, *IEEE Trans. on PAMI* **7** (2), 139–154, (1985).
10. X.W. Tu and B. Dubuisson, 3-D information derivation from a pair of binocular images, *Pattern Recognition* **23** (4), 223–235, (1990).
11. M. Pietrkainan and D. Harwood, Depth from three-camera stereo, In *Proc. Intl. Conf. CVPR*, pp. 2–8, (1986).
12. S.I. Olsen, Stereo correspondence by surface reconstruction, *IEEE Trans. PAMI* **12** (3), 309–315, (1990).
13. R.N. Chiou, K.C. Hung, J.K. Kuo, C.H. Chen and J.Y. Lee, Polyhedron recognition using three-view analysis, *Pattern Recognition* **25** (1), 1–16, (1992).
14. S.T. Barnand and W.B. Thompson, Disparity analysis of images, *IEEE Trans. PAMI* **2** (4), 333–340, (1980).
15. M. Ito and A. Ishii, Range and shape measurement using three-view stereo analysis, In *Proc. Intl. Conf. CVPR*, pp. 9–14, (1986).
16. S.T. Barnand and M.A. Fischler, Computational stereo, *Computing Surveys* **14** (4), 552–572, (1982).
17. U.R. Dhond and J.K. Aggarwal, Structure from stereo—A review, *IEEE Trans. SMC* **19** (6), 1489–1510, (1989).
18. R.C. Chiou, C.H. Chen, K.C. Hung and J.Y. Lee, The optimal camera geometry and performance analysis of a trinocular vision system, *IEEE. Trans. SMC* **25** (8), 1207–1220, (1995).
19. J.-D. Lee, A fast geometrical approach to camera extrinsic parameters, *Computers Math. Applic.* **32** (12), 93–100, (1996).
20. J.-D. Lee, Indoor robot guidance by landmark tracking, *Mathl. Comput. Modelling* **26** (4), 79–89, (1997).
21. D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, (1989).
22. Q.C. Meng, Genetic algorithms encoding study and a sufficient convergence condition of GAs, In *IEEE SMC'99 Conference Proceedings, Volume 1*, pp. 649–652, (1999).
23. T.C. Fogarty, Classifier system for control, In *Genetic Algorithms for Control System Engineering, IEEE Colloquium*, pp. 811–813, (1993).
24. A. Rosenfeld and A.C. Kak, *Digital Picture Processing, Volume 2*, Second Edition, Academic Press, New York, (1982).
25. F. O'Gorman and M.B. Glowes, Finding picture edges through collinearity of feature points, *IEEE Trans. Comput.* **25**, 449–452, (1976).
26. A.K. Griffith, Edge detection in simple scenes using *a priori* information, *IEEE Trans. Comput.* **22**, 371–380, (1973).