A Novel Live 3D Objects Reconstruction Scheme

Hui Zheng, Jie Yuan, Sidan Du
School of Electronic Science and Engineering, Nanjing University
Nanjing, China
Email: hzheng.nju@gmail.com, yuanjie@nju.edu.cn, coff128@nju.edu.cn

Abstract— **In this paper, we describe a novel method to create the complete 3D model of the object on uncalibrated images. First, we match the points both detected by multi-scale Harris corner detection algorithm and line detection technique. Second, we perform a projected reconstruction based on factorization using Singular Value Decomposition (SVD). After that, we are able to upgrade from projective to Euclidean structure and then eliminate the ambiguity in Euclidean reconstruction. Finally, we use 3D registration algorithm based on common points to build the whole 3D model of the object. Sufficient experiments proved the validity and efficiency of the method.**

Keywords-3D reconstruction; ambiguity in Euclidean Reconstruction; 3D Registration; line detection

I. INTRODUCTION

Reconstruction of the object in 3D from images taken by uncalibrated cameras has long been a topic of research in computer vision. Factorization has been a common and reliable method for 3D reconstruction and motion recovery. Sturm and Triggs [1] proposed a projective reconstruction algorithm based on factorization, they recovered projective depths by estimating a set of fundamental matrixes. Ponce [2] upgraded the projective reconstruction to Euclidean reconstruction on the assumption that the cameras are zero-skew. Mei Han and Takeo Kanade recovered the shape and motion from image sequence which taken by uncalibrated camera using factorization method in [3]. R Szeliski mentioned the ambiguity problem in the process of simultaneously recovering structure and motion with uncalibrated cameras in [4], this ambiguity would cause errors in 3D registration.

Displaying after meshing the 3D models is one of the important applications for 3D reconstruction. Direct feature points detection algorithm [9] [10] would have edge and corner information lost during 3D reconstruction, which would cause the unsatisfactory display. To solve this problem, the method describe in [5] using the feature points which are detected both by Harris and Sift [9] for reconstruction, but it would not reduce the edge information loss.

In this paper, the line detection algorithm and the feature points detection and matching algorithm are proposed in Section II. In Section III, we introduce the projective reconstruction and Euclidean reconstruction in Subsection A and Subsection B, and then propose an algorithm to eliminate ambiguity in Euclidean reconstruction after identifying its cause in Subsection C. In order to create the complete 3D model, we introduce the detail of 3D registration technique in Section IV. Experimental processes and results are introduced in Section V; we use four cameras for synchronized taking $n(n > 2)$ pictures for the moving object, and then reconstruct each of the four parts of the object from the images taken by each camera, and finally create the complete 3D model. In Section VI, we give a conclusion of this paper. There are three contributions in this paper:

- Propose a 3D reconstruction method using line detection technique that ensures the correctness and completeness of corners and edges on the 3D model.
- Describe the cause of ambiguity in Euclidean reconstruction and propose a method to eliminate ambiguity.
- Increase the quantity of common points in 3D using guided matching algorithm that enhance accuracy of 3D registration.

II. FEATURE POINTS DETECTION AND MATCHING

In this paper, we use the multi-scale Harris corner detection algorithm [8] to detect feature points on images. This algorithm is the combination of Harris operator and scale space theory. We first detect the Harris feature points in different scales, and then using LOG operator to select the appropriate scale and obtain the location of the feature points.

The feature points detected by traditional detection algorithm [5] [9] [10] cannot cover the edges and corners of the object, which will cause the loss of edge and corner information when reconstructing the 3D model. So we detect lines on images using line detection algorithm [6] based on Hough transform, and then match the points on the lines for reconstruction.

In this paper, we use the wide baseline matching algorithm [11] to match the points which are detected on the images taken by adjacent cameras, we call it “match between cameras”; we use the algorithm based on guided matching to match the points which are detected on the images taken by one camera, we call it “match in camera”.

There are two steps in “matching between cameras”. First, we execute initial matching. Calculate the correlation coefficient of feature points of the two images respectively. When the correlation coefficient is larger than a given threshold value, both the feature points are considered as the candidate of matching points. Search support strength from the neighborhood to accumulate matching strength. We merely consider the maximum support of each neighborhood as the initial matching points. Next, we use RANSAC method to calculate the fundamental matrix F and homography matrix H, and remove the mismatches at the same time.

There are also two steps in “matching in camera”. We execute initial matching at first. Next, we using guided matching algorithm [12] to find the points on other images that correspond to the feature points obtained from matching points between cameras and detected lines. Guided matching is a technique which can redirected match the designated feature points by using epipolar geometry and homography constraint.

The outline of feature points detection and matching is as follows:

Suppose $I_j^i(i = 1 \dots n)$ is the image taken by jth$(1 \leq$ j ≤ 4) camera at time i.

a) P_j^i =detect feature points on I_j^i

b) match between cameras:
$\{PO_{12}^n\ PO_{21}^1\} = \text{match}\{P_1^n\ P_2^1\}$;
$\{PO_{23}^n\ PO_{32}^1\} = \text{match}\{P_2^n\ P_3^1\}$;
$\{PO_{34}^n\ PO_{43}^1\} = \text{match}\{P_3^n\ P_4^1\}$;
$\{PO_{41}^n\ PO_{14}^1\} = \text{match}\{P_4^n\ P_1^1\}$;

c) line detection:
$m = \left\lfloor \frac{n}{2} \right\rfloor$;
$PL_1^m = line\ detect\{I_1^m\ \}$;
$PL_2^m = \text{line detect}\{I_2^m\ \}$;
$PL_3^m = \text{line detect}\{I_3^m\ \}$;
$PL_4^m = \text{line detect}\{I_4^m\ \}$;
PL are the points spaced selected on detected lines

d) match in camera:
Camera1:
$\{PI_1^1\ PI_1^2 \dots PI_1^n\} = \text{initial match}\ \{P_1^1\ P_1^2 \dots P_1^n\}$;
$\{PIL_1^1\ PIL_1^2 \dots PIL_1^n\} =$
guided match PL_1^m on I_1^m to $I_1^i(i \neq m)$;
$\{PIO_{12}^1\ PIO_{12}^2 \dots PIO_{12}^n\} =$
guided match PO_{12}^n on I_1^n to $I_1^i(i \neq n)$;
$\{\mathrm{PIO}_{14}^1\ \mathrm{PIO}_{14}^2 \dots \mathrm{PIO}_{14}^n\} =$
guided match PO_{14}^1 on I_1^1 to $I_1^i(i \neq 2)$;
$PM_1^i = PI_1^i \cup PIL_1^i \cup PIO_{12}^i(i = 1 \dots n)$;
$\{PM_1^1\ PM_1^2 \dots PM_1^n\}$ are final matching points of Camera 1. Camera 2, camera 3 and camera 4 do the similar way with Camera 1.

III. 3D RECONSTRUCTION

In order to create the complete 3D model, we have to reconstruct each of the four parts of the object. In Section III, we will describe the reconstruction algorithm.

A. Projective Reconstruction

Suppose there are m object points $X_j(j = 1 \dots m)$ and n perspective matrixes $P_i(i = 1 \dots n)$, x_{ij} is the projection of X_j that projected by P_i.

$$W = \begin{bmatrix} \lambda_{11}x_{11} & \lambda_{12}x_{12} & \cdots & \lambda_{1n}x_{1n} \\ \lambda_{21}x_{21} & \lambda_{22}x_{22} & \cdots & \lambda_{2n}x_{2n} \\ \vdots & & & \vdots \\ \lambda_{m1}x_{m1} & \lambda_{m2}x_{m2} & \cdots & \lambda_{mn}x_{mn} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \left[X_1 X_2 \cdots X_n\right] = PX \tag{1}$$

In (1), W is scaled measurement matrix and λ_{ij} is a non-zero factor called projective depth. The goal of projective reconstruction is to estimate the projective depth of x_{ij}. We use the algorithm based on SVD described in [3] [7] to estimate λ_{ij}:

a) Set $\lambda_{ij} = 1$. Compute the current scaled measurement matrix W, by (1)

b) Perform SVD decomposition on W : $UDV = \mathrm{SVD}(W)$.

c) Set $P' = U_4, X' = D_4 * V_4^T$ where $U_{4,}D_4, V_4^T$ are the first four lines of U, D, V.

d) Set $W' = P' * X'$, update λ_{ij} and W:

e) $\lambda_{ij} = \lambda_{ij} * \frac{W'^T_{ij} * W_{ij}}{W_{ij}^T * W_{ij}}$; $\quad W_i = \lambda_{ij} * x_{ij}$

f) Go to step 3 until D(5,5) is small enough.

B. Euclidean Reconstruction

The factorization of (1) recovers the motion P and shape X up to a 4 × 4 linear projective transformation H:

$$W = P' \times X' = (PH) \times (H^{-1}X) \tag{2}$$

The goal of Euclidean reconstruction is to calculate the matrix H that upgrades the P and X in projective space to P' and X' in Euclidean space. The $P_i{}'$ in Euclidean space can be written as:

$$P_i{}' = \alpha_i K * [R_i | T_i] \tag{3}$$

where α_i is a none-zero real number; R_i is a orthogonal matrix, which shows the rotation of the camera; T_i is a vector, which shows the position of the camera. Because of the orthogonality of R_i, we rewrite H as $H = [A|B]$, where A is 4 × 3 and B is 4 × 1. We have:

$$P_i A A^T P_i^T = \alpha_i{}^2 K R_i R_i^T K^T = \alpha_i{}^2 K K^T \tag{4}$$

In this paper we only discuss the case that camera pixels are square, and we shift principal point to the original, so the intrinsic parameters matrix K can be denoted by:

$$K = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where $a_x = a_y$. Substituting (5) into (4):

$$M_i = P_i A A^T P_i{}^T = P_i Q P_i{}^T = \begin{bmatrix} \alpha_i^2 a_x^2 & 0 & 0 \\ 0 & \alpha_i^2 a_y^2 & 0 \\ 0 & 0 & \alpha_i^2 \end{bmatrix} \tag{6}$$

where

$$Q = AA^T. \tag{7}$$

From (6) we can obtain the equation:

$$M_i(0,0) = M_i(1,1)$$
$$M_i(1,2) = M_i(1,3) = M_i(2,3) = 0 \qquad (8)$$

We can set up 4 equations from each frame, and given $\alpha_1 = 1$, we have $4n + 1$ equations to solve Q which have 10 unknown elements. Then we get the matrix A from Q by SVD decomposition.

To solve B we denote $P'$by $P' = [F|T] = P[A|B]$ and substitute $P = (P_x\ P_y\ P_z)^T, T = (T_x\ T_y\ T_z)^T$ into it :

$$T_x = P_x * B;\ T_y = P_y * B;\ T_z = P_z * B \qquad (9)$$

Put the origin of the world coordinate system at the center of gravity of the scaled object points to enforce:

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} * x_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \qquad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} * y_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \qquad (10)$$

From (9) (10) we set up 2n liner equations to solve B.

After A and B have been computed, we can obtain motion matrix P' and shape matrix X' in Euclidean space.

C. Eliminate Ambiguity in Euclidean Reconstruction

There exists an ambiguity in Euclidean Reconstruction we described in Section III Subsection B. The 4×3 matrix A that we obtained from Q using SVD decomposition is non-unique. Suppose A is a solution of (7), we have:

$$Q_{ij} = A_{i1} * A_{j1} + A_{i2} * A_{j2} + A_{i3} * A_{j3} \qquad (11)$$

From (11) we know A still a solution if we reverse any columns of it:

$$Q_{ij} = (\mp A_{i1}) * (\mp A_{j1}) +$$
$$(\pm A_{i2}) * (\pm A_{j2}) + (\pm A_{i3}) * (\pm A_{j3}) \qquad (12)$$

So, there are at most 8 solutions for equation $Q = AA^T$. Substitute A into equation $P^{'} = [F|T] = P \times [A|B]$:

$$F_{ij} = P_{i1} * A_{1i} + P_{i2} * A_{2i} + P_{i3} * A_{3i} \qquad (13)$$

Moreover, $P^{'}$ and $X^{'}$ satisfy:

$$P^{'} \times X^{'} = W \qquad (14)$$

From (13) and (14) we know that we will obtain different motion matrix $P^{'}$ and shape matrix $X^{'}$ corresponding to different solutions of (7). Suppose A_1 and A_2 are different solutions of (7), and the jth$(j < 4)$ column of A_1 has opposite signs to the jth column of A_2, then each members of jth column of $P_1^{'}$ and $P_2^{'}$ computed from A_1 and A_2 is opposite numbers to each other. Also according to (14) we know that the jth row of $X_1^{'}$ and $X_2^{'}$ has opposite signs corresponding to $\mathrm{P}_1^{'}$ and $\mathrm{P}_2^{'}$. When we perform Euclidean reconstruction using the method mentioned in Section III Subsection B, we will obtain one solution randomly from all solutions which satisfy (11) and (14).

Suppose $P_a^{'}$ and $X_a^{'}$ are the solution sets which satisfy (7) (13) and (14).We can divide $X_a^{'}$ into two groups $X_l^{'}$ and $X_r^{'}$. Every solution in one group can transform to every other one in this group through rotation and translation, but the solutions in different groups cannot transform to each other like that. We can image this situation as every solution $X_l^{i'}\ (X_l^{i'} \in X_l^{'})$ is in the left-handed coordinate system, and $X_r^{i'}\ (X_r^{i'} \in X_r^{'})$ is in the right-handed coordinate system.

We must make sure all shapes that are reconstructed from the images which taken by different cameras are in the same type of coordinate system when we generate the complete 3D model of the object using 3D registration algorithm. Figure 1 shows the wrong result that doing 3D registration with two shapes that in different groups.

Suppose I_j^i is the image taken by jth camera at time i, $P_j^{i}{'}$ is the motion matrix of jth camera at time i, $X_j^{'}$ is the shape matrix which reconstruct from I_j^i. If we have the images $\{I_j^{t_1} \ldots I_j^{t_1}\}$ and $\{I_j^{t_2} \ldots I_j^{t_2}\}(j = 1, 2, 3, 4)$ which are taken by each camera at different moment t_1 and t_2, the rotational direction of the object from t_1 to t_2 computed from each motion matrix $P_j^{i}{'}(i = t_1, t_2; j = 1 \ldots 4)$ must be the same(when static cameras capture videos of moving object, the motions that obtained from motion matrixes of cameras have opposite directions with the motion of object). For example, if the object rotate clockwise from t_1 to t_2, the rotational directions that we computed from $P_j^{t_1}{'}$ and $P_j^{t_2}{'}$ must be counter-clockwise. But if we have the wrong situation like Figure 1, each rotational direction obtained from each motion matrixes must be the opposite, so we can eliminate ambiguity according to this property. From (6) we compute intrinsic parameters matrix K as:

$$a_x = a_y = \sqrt{\frac{M_i(0,0) + M_i(1,1)}{2a_i}} \qquad (15)$$

We eliminate the ambiguity of Euclidean reconstruction after normalize $P_j^{i}{'}$:

a) Normalize $P_j^{i'}(i = t1, t2)$:

$\mathrm{if}\left(P_j^{i'}(1,1) < 0\right)$

$P_j^{i'}(:,1) = -P_j^{i'}(:,1) \quad X_j^{'}(1,:) = -X_j^{'}(1,:)$

$\mathrm{if}\left(P_j^{i'}(2,2) < 0\right)$

$P_j^{i'}(:,2) = -P_j^{i'}(:,2) \quad X_j^{'}(2,:) = -X_j^{'}(2,:)$

b) Compute K_j and $\{R_j^{t1}, R_j^{t2}\}$ from (15) and (3).

c) Compute rotational direction RD_j from $\{R_j^{t1}, R_j^{t2}\}$.

d) Compare $RD_j (1 < j \leq 4)$ with RD_1:

e) $\mathrm{if}\left(RD_j \text{ has the opposite direction to } RD_1\right)$

$P_j^{i'}(:,3) = -P_j^{i'}(:,3) \quad X_j^{'}(3,:) = -X_j^{'}(3,:)$

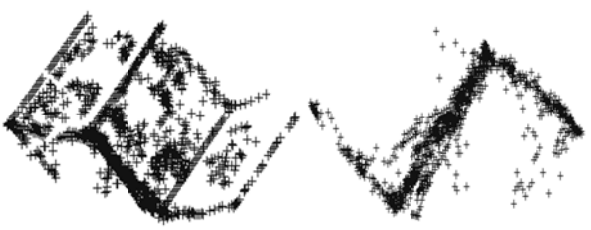

Figure 1. Wrong solution because perform 3D registration with two shapes in different groups.

IV. 3D REGISTRATION

Suppose $\{PO_{12}^{n}\ PO_{21}^{1}\}$ $\{PO_{23}^{n}\ PO_{32}^{1}\}$ $\{PO_{34}^{n}\ PO_{43}^{1}\}$ and $\{PO_{41}^{n}\ PO_{14}^{1}\}$ are matchings between cameras which we obtained in Section II. And we can find $\{X_{12}\ X_{21}\}$ $\{X_{23}\ X_{32}\}$ $\{X_{34}\ X_{43}\}$ $\{X_{41}\ X_{14}\}$, which are 3D points corresponding to them.

We take X_{12} from Camera 1 and X_{21} from Camera 2 as example to describe 3D registration. The transformation between X_{12} and X_{21} is:

$$X_{12} = s \times R \times X_{21} + t \qquad (16)$$

where s is non-zero number called scale factor, R is rotation matrix, t is translation vector. We compute R as:

$$R = U * diag\,(1,\ 1,\ 1,\ det\,(UV')) * V' \qquad (17)$$

We compute U, D, V in (17) from

$$UDV = svd\left(\sum_{i=1}^{n}(X_{12_i} - X_{12A})(X_{21_i} - X_{21A})'\right) \qquad (18)$$

where $X_{12A}\ X_{21A}$ are the center of gravity of $X_{12}\ \ X_{21}$:

$$X_{12A} = \frac{1}{n}\sum_{i=1}^{n} X_{12_i}\,;\ X_{21A} = \frac{1}{n}\sum_{i=1}^{n} X_{21_i} \qquad (19)$$

We obtain s from

$$s = \frac{C'}{\sum_{i=1}^{n} ||X_{21_i} - X_{21A}||^2} \qquad (20)$$

where C' in (20) compute as (21).

$$C' = D(1,1) + D(2,2) + det(UV') * D(3,3). \qquad (21)$$

We obtain t by substituting s, R into (22).

$$t = \frac{1}{n}\sum_{i=1}^{n} X_{12_i} - S * R * \frac{1}{n} * \sum_{i=1}^{n} X_{21_i} \qquad (22)$$

Finally we accomplish 3D registration according to (23).

$$Xb_2' = s \times R \times X_2' + t \qquad (23)$$

V. EXPERIMENTAL ANALYSIS AND RESULTS

In this Section, we will show the processes of our experiments and analyze the results.

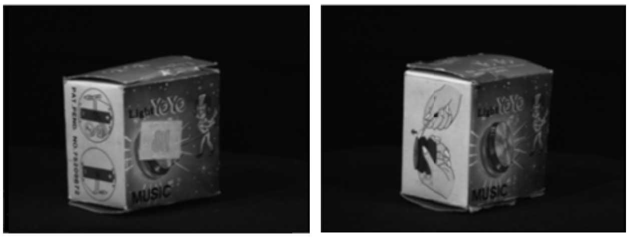

Figure 2. Original images

We use four static uncalibrated cameras to take photos simultaneously of the moving object. Each camera takes three photos on the object with small-scale movement. Figure 2 shows two of the images taken by different cameras. We implement our method in MATLAB, and then display the model with texture in OpenGL.

Reprojective error is an important criterion which represents the reconstruction accuracy. Table 1 shows the main reprojective error of middle image of each part which are reconstructed from images taken by each camera after 3D registration. We can see the reconstruction error of our method is small and acceptable. Because we translate the other parts to part 1's coordinate system in our experiment, the mean reprojective error in part 1 is a little lower than other part.

In order to enhance the accuracy of 3D registration, we using guided matching technique in step 'match in camera' to obtain more common points for 3D registration. Table 2 compares the mean registration error and quantity of common points obtained in 3D registration with guided matching technique with those obtained without guided matching technique. We transform the 3D coordinate into a normalized one before registration, in which each component (x, y or z) of 3D points falls into $[-1\,,1]$. From Table 2, we can see our method obtain more common points than the method without guided matching technique, and the mean registration error computed by our method is much smaller than the other one.

TABLE 1. MEAN REPROJECTIVE ERROR AFTER 3D REGISTRATION

	Part1	Part 2	Part 3	Part 4
Mean error (pixel)	0.154	0.362	0.406	0.572
Max error (pixel)	1.118	1.220	1.966	2.34

TABLE 2. MEAN REGISTRATION ERROR AND QUANTITY OF COMMON POINTS IN REGISTRATION

	Obtain common points with guided matching		**Obtain common points without guided matching**	
	Quantity	**Mean error**	**Quantity**	**Mean error**
Part 1 with Part 2	103	0.0129	11	0.0259
Part 1 with Part 3	122	0.0109	23	0.0207
Part 2 with Part 3 and 4	197	0.0198	24	0.0551

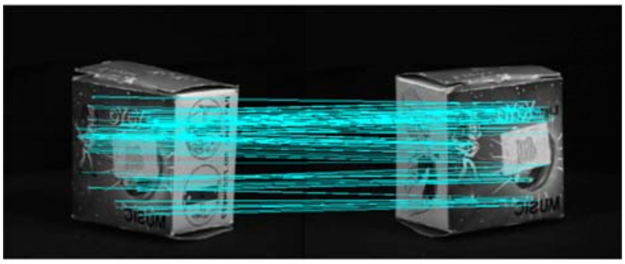

Figure 3. Match between groups

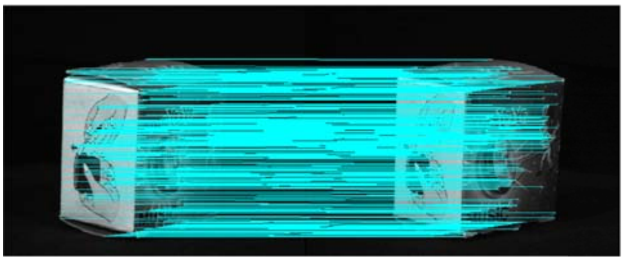

Figure 4. Match in groups

Figure 3 shows the result of 'matching between cameras'. The points in there are matched accurately each other and there are enough matching points for 3D registration. Figure 4 shows one result of "matching in camera". There are three parts matching points in this figure: the points detected by multi-scale Harris corner detection algorithm are the basic matchings; the points on lines and corners prevent the edge and corner information loss of the 3D model; the points corresponding to the matchings in "matching between cameras" step increase the quantity of common points in 3D, which lead to the increase the accuracy of 3D registration.

Figure 5 shows the detected lines on image with white color. Figure 6 shows the mesh of the 3D points generated by performing 3D triangulation algorithm. From that we can see the edges and corners of the box are accurate and complete.

Figure 7 shows the result of 3D reconstruction without line detection technique. There is a lot of edge and corner information loss of the model and the display effect is unsatisfactory.

Figure 8 shows the 3D model reconstructed by our method. Apparently our method prevents the edge and corner information loss effectively and shows a satisfactory display.

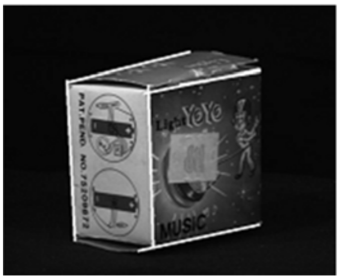

Figure 5. Image with detected lines

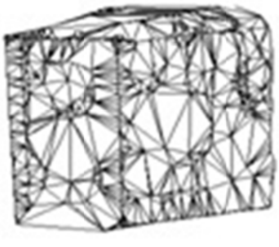

Figure 6. 3D mesh

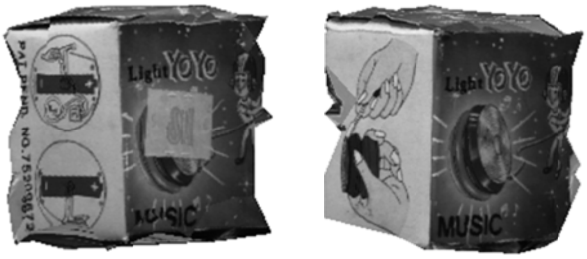

Figure 7. Reconstruct without line detection technique

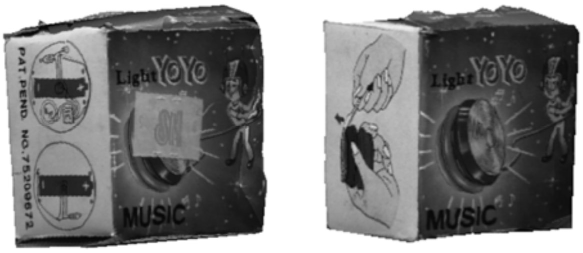

Figure 8. Reconstruction result of our method

VI. CONCLUSION

Several conclusions can be drawn from the experiment results. Firstly, line detection algorithm using in our method insures the accuracy and completeness of the edges and corners on reconstructed model. Secondly, the algorithm we proposed in Section III eliminates the ambiguity in Euclidean reconstruction effectively. Thirdly, since we increase the quantity of common points in 3D using guided matching technique, we obtained high accuracy results of 3D reconstruction and 3D registration. The display effect of the model reconstructed from our method is satisfactorily.

VII. ACKNOWLEDGEMENT

This paper is supported by the Fundamental Research Funds for the Central Universities, Number: 1107021051, and National Natural Science Funds of Jiangsu Province of China, Number: BK2010386.

REFERENCES

[1] Peter Strum and Bill Triggs, "A factorization based algorithm for multi-image projective structure and motion," In European conf. Computer Vision — ECCV '96, Cambridge, U.K., 1996, pp. 709-720.

[2] Jean Ponce, "On Computing Metric Upgrades of Projective Reconstructions under the Rectangular Pixel Assumption," Proc. Workshop. 3D Structure from Multiple Images of Large-Scale Environments (SMILE'00), pp. 52-67, 2002.

[3] Mei Han and Takeo Kanade, "Creating 3D Models with Uncalibrated Cameras," Proc. IEEE Workshop. Application of Computer Vision (WACV2000), pp.178-185, 2000.

[4] Richard Szeliski. "Computer Vision: Algorithms and Applications," Chapter 7, 2010, pp. 343-374

[5] Keju Peng, Xin Chen, Dongxiang Zhou, and Yunhui Liu, "3D reconstruction based on SIFT and Harris feature points," Proc. IEEE Conf. Robotics and Biomimetics (ROBIO'09), pp. 960-964, December 2009

[6] Thuy Tuong Nguyen, Xuan Dai Pham, and Jae Wook Jeon, "An improvement of the Standard Hough Transform to detect line segments," Proc. IEEE Conf. Industrial Technology (ICIT'08), Chengdu, China, pp. 1-6, 2008.

[7] Toshio Ueshiba and Fumiaki Tomita, "A Factorization Method for Projective and Euclidean Reconstruction from Multiple Perspective Views via Iterative Depth Estimation," Proc. European Conf. Computer Vision, pp. 296-310, 1998.

[8] Krystian, Mikolajczyk and Cordelia Schmid, "Scale & Affine Invariant Interest Point Detectors," International Journal of Computer Vision, 2004, pp. 63-68.

[9] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision. 2004, pp. 91-110.

[10] Zhiyong Ye, Yijian Pei, and Jihong Shi, "An Adaptive Algorithm for Harris Corner Detection," Proc. IEEE Conf. Computational Intelligence and Software Engineering (CiSE'09), pp. 1-4, December 2009.

[11] Jiangjian Xiao and Mubarak Shah, "Two-frame wide baseline matching," Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. 2003, pp. 603-609.

[12] Benjamin Ochoa and Serge Belongie, "Covariance propagation for guided matching," Proc. Workshop. Statistical Methods in Multi-Image and Video Processing (SMVP'06). On CD-ROM, 2006.