3D Object Localization Based on Occluding Contour Using STL CAD Model

Kenichi Maruyama, Yoshihiro Kawai, Takashi Yoshimi and Fumiaki Tomita
National Institute of Advanced Industrial Science and Technology, Tsukuba, 305–8568 Japan
{*k.maruyama, y.kawai, tak-yoshimi, f.tomita*}*@aist.go.jp*

Abstract

This paper describes a method to localize 3D objects, which is the extension of the segment-based object recognition method to use on a STL CAD model. Models for localization are automatically generated using contour generators, which are estimated by occluding contours of projected images of the CAD model from multiple viewing directions and depth images computed with a graphics accelerator. In addition, the model is dynamically updated in a recognition process according to the viewing direction. Localization process is based on multiple hypotheses verification using the model and 3D boundaries reconstructed from stereo images. Experimental results show the effectiveness of the proposed method.

1. Introduction

Model based 3D object localization from intensity image(s) is one of the most important issues in computer vision. For metallic or plastic industrial objects, it is difficult to obtain stable dense 3D data using area-based stereo method, because these objects often lack texture information. Therefore, a 3D-3D matching method using dense 3D data [5, 1] is unsuitable for these objects. To recognize such objects, several algorithms using boundaries have been proposed [7, 2]. Though most of these methods are a 2D-3D matching algorithm, we are developing a 3D-3D matching algorithm using 3D boundaries in a Versatile Volumetric Vision (VVV) system [10]. Because we consider that the 3D-3D matching method is more robust than the 2D-3D matching method against occlusion and clutter.

This research is the extension of the segment-based object recognition method [9] to use on an STL CAD model. In the new method, models for localization, including polyhedron and free-form objects, are automatically generated using contour generators of the CAD model. The contour generators are estimated by occluding contours (or apparent contours) of projected images of the CAD model and depth images computed with a graphic accelerator. To create the model from various viewing directions, we set multiple viewing directions. In addition, the model is dynamically updated in a recognition process according to the viewing direction. To show the effectiveness of the proposed method, we present the experimental results for objects with various shapes.

2. Basic Object Localization Method

We briefly introduce the segment-based object recognition method [9]. In this method, 3D boundaries are reconstructed from stereo images, and data of the geometric features are generated by fitting a line or circle to edge segments. Multiple hypotheses on the position and orientation are generated by comparing the model geometric features with the data geometric features (we call this procedure initial matching). Each hypothesis is verified and improved by an iterative process using the model and data points (we call this procedure fine adjustment).

Figure 1 (a), (b) show an example of initial matching and fine adjustment, respectively. As shown in Figure 1 (c), geometric features consist of vertexes and arcs.

3. Model

The model is automatically generated from a CAD model. A CAD format used in this research is a Standard Triangulated Language (STL), which is a general CAD format. The model consists of model geometric features and model points.

3.1. Model geometric features

The geometric features of a model consist of vertexes and arcs, which are built from the contour points of the projected image of the STL model. To create

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

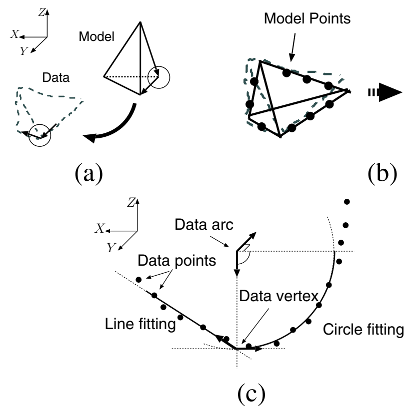
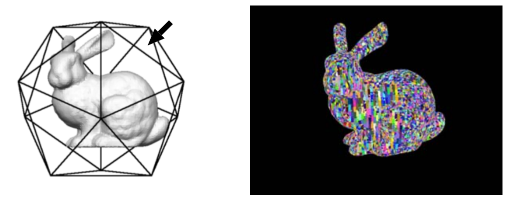


Figure 1. (a) Initial matching using a vertex, (b) fine adjustment, (c) geometric features



(a) a viewing direction (b) projected image

(c) 3D segments (d) geometric features

Figure 2. Generating model

vertexes and arcs from various viewing directions, the directions are discretely set on every face of a geodesic dome. Such discrete modeling can cause an error in initial matching. However, the error can be corrected by fine adjustment.

To project a model onto an image, we use OpenGL, which can execute high-speed rendering and hidden-line removal with existing hardware [6]. In addition, OpenGL can store the depth information at each pixel. Using this information, 3D points corresponding to each contour point are calculated. The 3D contours generated by this process are equivalent to the 3D boundaries obtained by stereo cameras. Therefore, the 3D contours are suitable for generating model vertexes and arcs.

It has been reported that the smoothness of the 3D contours obtained by OpenGL heavily depend on the size of the facets [3]. Because we assume that the STL models used in this research consist of enough small facets, the smoothness does not pose a problem. The procedure of creating the model geometric features is as follows.
1) Select a viewing direction (Figure 2(a)).
2) Project the model onto a 2D image with a different color at each facet (Figure 2(b)).
3) Extract contours of the projected image. The extracted contours are segmented using the same algorithm as the one used for contours from the intensity image [8].
4) Calculate the 3D position for each contour point using the depth values stored by OpenGL. Their normal vectors are set corresponding to facet decided by their color [4].
5) Fit a line or circle to each 3D segment (Figure 2(c)), generate the model vertexes and arcs (Figure 2(d)).
6) Repeat 1 through 5 for each viewing direction.

3.2. Model points

The model points reflect the object's 3D shape, and each of them has a 3D position and normal vector. The model points are generated dynamically from the contour points of the projected image in each step of fine adjustment. If model points, set beforehand at equal intervals in 3D space, are projected onto an image, the density of the projected model points on the image depends on the distance from the camera to object. In the proposed method, the densities of the model points and data points on the image are equal, because both types of points are generated on the image.

The object's position and orientation are expressed as a 4×4 transformation matrix $\boldsymbol{T}$, whose initial value is derived at the initial matching. The procedure of generating model points and fine adjustment is as follows.
1) Project model onto the image according to $\boldsymbol{T}$.
2) Extract contour points of the projected image. Their 3D positions and normal vectors are set in the same way described in 3.1. The points which have 3D information are model points.
3) Search data points corresponding to the model points using their normal vector. Using the sets of correspondences, $\boldsymbol{T}$ is updated by the least squares method [9]. Here, n is the number of the correspondences.
4) Repeat 1 through 3 until the precision is small enough. If the precision does not converge with much iteration, the hypothesis is wrong.

After all hypotheses are verified, the hypothesis which has the largest n is selected. $\boldsymbol{T}$ corresponding to this hypothesis is the recognition result.

Reconstruction of a curved surface using stereo vi-

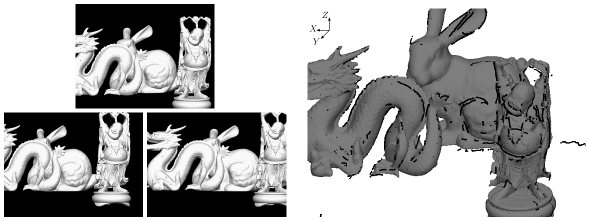


Figure 3. Synthetic stereo images and recognition result

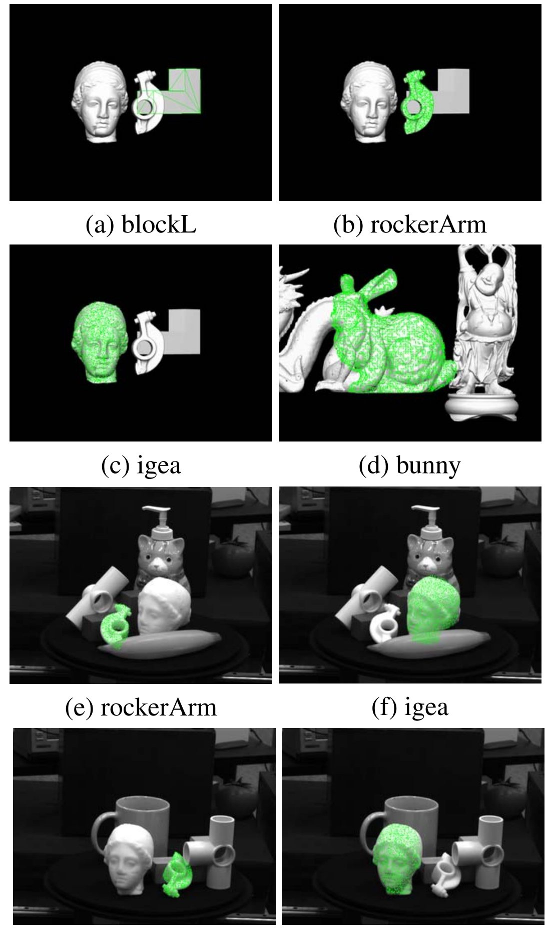
sion involves a measuring error. We do not deal with this error because we assume that this error is small.

4. Experiments

To evaluate the effectiveness of our method, experiments were carried out using synthetic and real images. The resolution of an image is 640×480 pixels and 256 gray levels. Models for localization were generated from STL using the proposed method. The STL models used in our experiments are shown in Table 1. The blockL was created using commercial CAD software. The bunny, dragon and buddha are courtesy of the Stanford 3D Scanning Repository. The rockerArm and igea are courtesy of Cyberware. In our experiments, we used a desktop PC with Intel Core2Duo E6850 and GeForce 8800 Ultra graphics card. Using a 3D printer, we created real 3D objects of the rockerArm and igea.

Figure 3 shows examples of synthetic stereo images and the recognition result: the models are shown with reconstructed 3D boundaries as shaded surfaces. Figure 4 shows the experimental results by projecting the model onto a left stereo image as green dots. In Figure 4, (a) to (d) are the synthetic images, (e) to (h) are the real images. Table 1 shows the computational cost in Figure 4, where M_v, M_c are the number of model vertexes and circles; D_v, D_c are those of data; H is the number of hypotheses in initial matching; $\Delta\phi_x$, $\Delta\phi_y$, $\Delta\phi_z$ are the rotation errors of the coordinate axes (degree); t is the translation error (mm); and T is the execution time (second). V is the number of viewing directions for model generation. These result shows that the proposed method is robust for occlusion and clutter, and has high accuracy of position and orientation. However, there is the problem of much computational cost.

To evaluate the accuracy of the position and orientation in real images, experiments using a turntable were carried out [9]. We used blockL, rockerArm and igea as real objects. The size of each is shown in Table 2. As shown in Figure 5, the rotational stage of the turntable was set up in parallel with the X-Y plane of the world



(a) blockL (b) rockerArm

(c) igea (d) bunny

(e) rockerArm (f) igea

(g) rockerArm (h) igea

Figure 4. Experimental result

coordinate system. An object was set on the rotational table and then rotated for 360 degrees at intervals of 1 degree. Figure 5 shows the motion of a point for each model in X-Y coordinates. Table 2 shows the standard deviation of the Z-position and rotation angles around the X-axis and Y-axis. These values should be constant in this experiment. The results also show that the proposed method has high accuracy.

5. Conclusion

This paper describes a method to localize 3D objects. We extend the object recognition method using 3D boundaries to use on an STL CAD model. Models for localization are automatically generated using contour generators which are estimated by occluding contours of the projected images of the CAD model and depth images computed with a graphics accelerator. To create models from various viewing directions, multiple viewing directions are set. The model points are dy-

Table 1. Computational costs

	V	M_v	M_c	D_v	D_c	H	$\Delta\phi_x$	$\Delta\phi_y$	$\Delta\phi_z$	Δt	T
blockL (Fig. 4(a))	6	28	0	66	64	317	0.05	0.49	0.23	0.61	1.44
rockerArm (Fig. 4(b))	80	979	833	66	64	12485	1.34	0.11	2.26	0.85	67.70
rockerArm (Fig. 4(b))	320	3859	3299	66	64	31023	0.39	1.88	2.81	0.59	269.27
igea (Fig. 4(c))	80	1294	1303	66	64	29854	2.23	0.66	0.84	0.05	99.83
igea (Fig. 4(c))	320	5260	5354	66	64	123527	1.71	2.22	0.37	0.41	416.72
bunny (Fig. 3, Fig. 4(d))	20	510	382	142	246	29054	0.12	0.23	2.33	4.27	80.00
dragon (Fig. 3)	20	922	670	142	246	48860	0.64	0.02	1.35	1.54	172.97
buddha (Fig. 3)	20	515	428	142	246	31450	0.25	0.24	0.50	1.24	105.51
rockerArm (Fig. 4(e))	80	979	833	101	110	19492	–	–	–	–	114.42
igea (Fig. 4(f))	80	1294	1303	101	110	25340	–	–	–	–	117.88
rockerArm (Fig. 4(g))	80	979	833	82	98	19580	–	–	–	–	120.78
igea (Fig. 4(h))	80	1294	1303	82	98	20406	–	–	–	–	102.25

Table 2. Object size and standard deviation of Z-position, R_x, and R_y in Figure 5

	blockL	rockerArm	igea
Size (cm)	9×3×6	4×2×8	7×10×10
Z (mm)	0.50	0.52	1.58
R_x (deg.)	0.57	0.95	5.03
R_y (deg.)	0.61	1.01	5.00

namically updated according to the viewing direction. In this procedure, the densities of the model points and data points on the image are equal. Experimental results using synthetic and real images show the effectiveness of the proposed method. In the future, we will reduce the computational cost by reducing the number of hypotheses in initial matching.

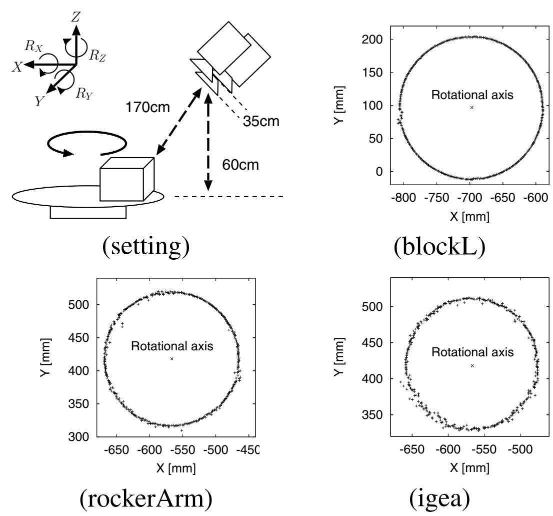


Figure 5. Error evaluation

References

[1] P. J. Besl. A method for registration of 3-D shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992.

[2] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans on PAMI*, 24(7):932–946, 2002.

[3] S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *Int. J. Computer Vision*, 72(2):159–178, 2007.

[4] Y. Iwashita, R. Kurazume, K. Hara, and T. Hasegawa. Fast alignment of 3D geometrical models and 2D color images using 2D distance maps. *3DIM'05*, pages 164–171, 2005.

[5] A. E. Johnson and M. Heber. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. on PAMI*, 21(5):433–449, 1999.

[6] Y. Kita, D. L. Wilson, J. A. Noble, and N. Kita. A quick 3D-2D registration method for a wide-range of applications. *Proc. ICPR 2000*, 1:981–986, 2000.

[7] D. Kotake, K. Satoh, S. Uchiyama, and H. Yamamoto. A fast initialization method for edge-based registration using an inclination constraint. *ISMAR 2007*, pages 239–248, 2007.

[8] K. Sugimoto and F. Tomita. Boundary segmentation by detection of corner, inflection and transition points. *Proc. VCV'94*, pages 13–17, 1994.

[9] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita. 3D object recognition in cluttered environments by segment-based stereo vision. *Int. J. of Computer Vision*, 46(1):5–23, 2002.

[10] F. Tomita, T. Yoshimi, T. Ueshiba, Y. Kawai, Y. Sumi, T. Matsushita, K. Sugimoto, and Y. Ishiyama. R&D of versatile 3D vision system VVV. *Proc. IEEE SMC'98*, pages 4510–4516, 1998.