International Journal on Smart Material and Mechatronics
IJSMM Vol. 3 No. 1 2016

p-ISSN: 2356-5314
e-ISSN: 2460-075X

Object Recognition Using a Human-like Vision Analysis System

Maierdan Maimaitimin, Keigo Watanabe and Shoichi Maeyama
Department of Intelligent Mechanical Systems
Okayama University
Okayama, Japan
merdan-m@usmm.sys.okayama-u.ac.jp,{watanabe,maeyama}@sys.okayama-u.ac.jp

Received: 30 September 2015 / Revised: 10 November 2015 / Accepted: 18 December 2015 / Published online: 20 January 2016 © IJSMM2016

***Abstract*—This paper addresses the problem of recognizing three-dimensional objects, in which it considers not only the shape of objects but also the surface conditions. These two information sources are combined by using a human-like vision analysis, which can detect the shape features in 2D and the surface condition in 3D. Here we create two kinds of database from CAD models and real world objects for Hidden Markov Model. One is the 3D to 2D omni-directional projected shape data and the other is the surface condition data extracted from each projected direction. The shape feature and the surface condition are encoded experimentally by an easy way and saved as HMMs' observation data. By performing the same process, the hidden state dataset can be obtained from the real world objects. The result shows the surface condition is a very valid argument for an object recognition system.**

***Index Terms*—3D point cloud, feature extraction, HMM, object recognition.**

I. Introduction

Object recognition has been intensively studied over the last decade. In the very begging of it, feature-based geometric approaches were the most wildly used and genetic algorithms were used for recognizing 3D objects from two-dimensional intensity images by assuming orthographic projection [1]. A way that uses a new class of local image features was developed, and it was invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection [2]. These results show the importance of RGB image in object recognition, but with the development of science and technology, a lot of new sensors are available, so that a depth sensor such as the Microsoft Kinect can be used to improve the object recognition. After that there appeared a lot of research which based on 3D features. There are some very good research results, i.e., it approached some methods on object recognition based on 3D point cloud data [3-5].

It is very effective in some research fields of intelligent system, but not enough for the human machine interaction in the civil application. When interacting with an intelligent system, we hope it can recognize object in the way how a human recognizes it. So it not only recognizes the object in a high resolution, but also should see things like a human. That is, it should find the features of an object which also can be captured by human vision; otherwise some misunderstanding may happen during human computer interaction. In this paper, we discuss a proper way to recognize objects by using object's common features which are utilized to recognize objects by human. Human can recognize objects almost instantaneously and involuntarily because of the invariance of objects. Whenever a human looks at any objects, a human brain extracts the features in some way, irrespective of the size, orientation, illumination, perspective, etc. [6]. Human brains can remember an object by its shape and inherent features. In our system, we use the object's shape and the invariance features, which are represented by surface conditions [7]. Also since a human needs more than a decade to experimental things to recognize most of objects, in our system we process a huge RGB-D data as database. The surface condition dataset is created by analyzing the most part of an object's surface, which compose a curved surface or an inclined plane. It is proved that it strongly affects the result of a recognition system performed by the HMMs [8-9], where the concept chart is shown in Fig. 1.

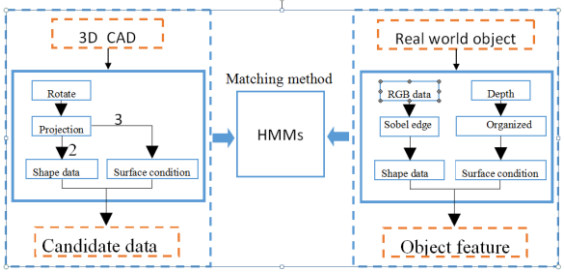


Fig. 1. Concept of recognition system

II. World Data Processing

It needs more than a decade for a human to see and remember the more and more objects, and summarize the common features of those objects. According to this, the size of available data must be huge and complete, in which we call such data the world data. CAD models are used to create the world database. As can be seen in Fig. 1, the data created from a CAD model have two parts, one is the 3D to 2D omni-directional projected shape data and the

other is the extracted surface condition data when CAD models are projected onto the 2D plane.

A. Omni-directional Projection

To generate any omni-directional projected data, it needs to project the CAD model onto a 2D plane when rotating it. Fig. 2 shows the concept of virtual omni-directional projection space.

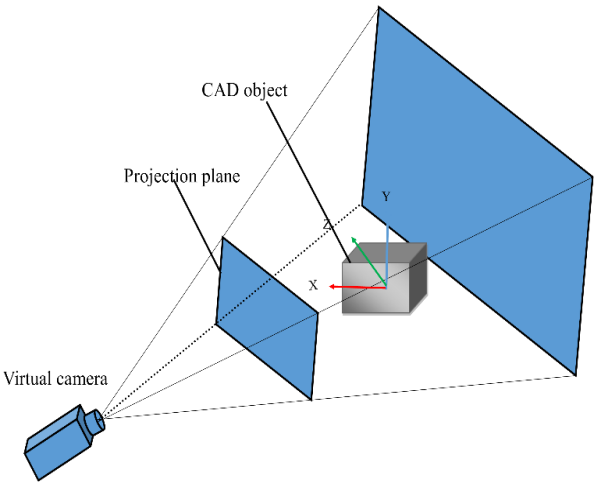


Fig. 2. Omni-directional projection

1) Rotation

For rotating an object in 3D space, we use a basic rotation matrix which is presented in the following equations:

$$R_x(\theta_x)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x & \cos\theta_x\end{bmatrix} \quad (1)$$

$$R_y(\theta_y)=\begin{bmatrix}\cos\theta_y & 0 & \sin\theta_y\\ 0 & 1 & 0\\ -\sin\theta_y & 0 & \cos\theta_y\end{bmatrix} \quad (2)$$

$$R_z(\theta_z)=\begin{bmatrix}\cos\theta_z & -\sin\theta_z & 0\\ \sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{bmatrix} \quad (3)$$

We can summarize the whole rotation as $R=R_z(\alpha)R_y(\beta)R_x(\gamma)$ which depends on the roll angle α pitch angle β, and yaw angle γ .

2) 3D to 2D Projection

After rotating the object in each angle, it is projected on a 2D plane to generate the shape dataset. The following equations show a very general way to project objects onto 2D plane.

$$X'=\{(X-X_c)\times(F/Z)\}+X_c \quad (4)$$

$$Y'=\{(Y-Y_c)\times(F/Z)\}+Y_c \quad (5)$$

where the virtual camera is located in (X_c,Y_c,Z_c) and the 3D point to be projected is denoted by $P=(X,Y,Z)$. The distance from the camera to the 2D plane to be projected is F, so that the equation of the plane is $F=Z-Z_c$. The 2D coordinates of P projected onto the plane are given by (X',Y'). After generating the binary image of a model shape to be projected, the obtained image is divided on the average into 16 small windows.

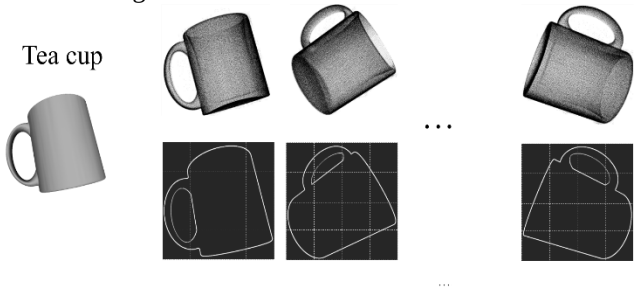


Fig. 3. Omni-directional projection

3) Corner Detection Methods

Since the binary shape image of each angle is obtained, as shown in Fig. 3, the Sobel operator can be used to generate the edge lines of each projection. After that, the Harris corner detection algorithm is used with a high threshold value to detect a large angle corner. The Harris corner detection algorithms is presented as the following steps [10]:

a) Compute x and y derivatives of image

$$I_x=G_\sigma^x\cdot I \qquad \textbf{Error! Not a valid link.}$$

where G_σ^x and G_σ^y denote the gradient in the X and Y direction and I is the intensity.

b) Compute the products of derivatives at every pixel

$$I_{x2}=I_x\cdot I_x \qquad I_{y2}=I_y\cdot I_y \qquad I_{xy}=I_x\cdot I_y$$

c) Compute the sums of the products of derivatives at each pixel

$$S_{x2}=G_{\sigma'}\cdot I_{x2} \qquad S_{y2}=G_{\sigma'}\cdot I_{y2} \qquad S_{xy}=G_{\sigma'}\cdot I_{xy}$$

d) Define at each pixel (x,y) the matrix

$$H(x,y)=\begin{bmatrix}S_{x2}(x,y) & S_{xy}(x,y)\\ S_{xy}(x,y) & S_{y2}(x,y)\end{bmatrix}$$

e) Compute the response of the detector at each pixel $R=\mathrm{Det}(H)-k(\mathrm{Trace}(H))^2$ *, where k is an empirical constant value.*

f) Take the point as a corner, if R exceeds a certain threshold

p-ISSN: 2356-5314
e-ISSN: 2460-075X

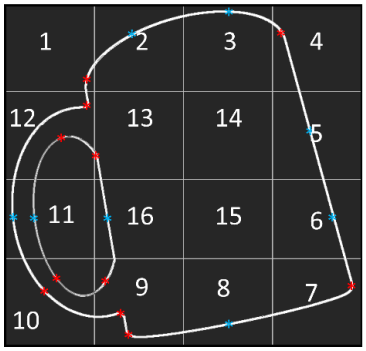


Fig. 4. Result of corner detection: red points represent the corners, whereas blue points represent the edges.

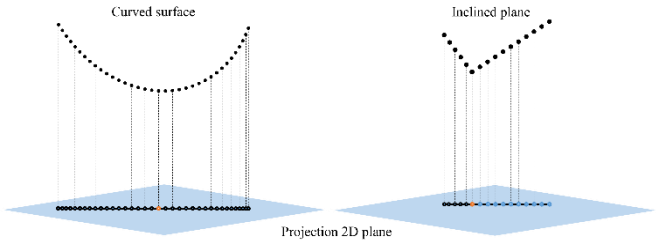
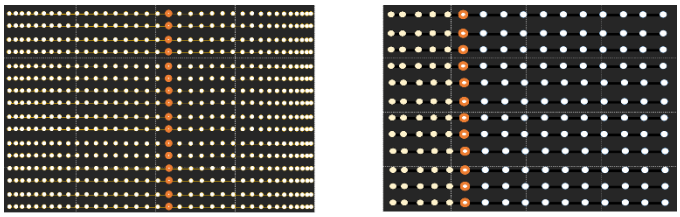


Fig. 5. Difference manifestations of surface projections (1)



(c) (d)

Fig. 6. Difference manifestations of surface projections (2)

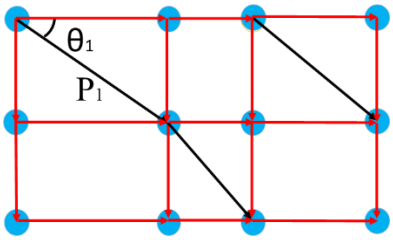


Fig. 7. Analysis method of a surface

Also the small angle corners can be found by detecting the intersection point of the line with the small window's edge.

If the line is intersected with adjacent-side edges, then it is defined as a corner; otherwise it is regarded as an edge. The result is shown in Fig.4.

4) Data Summarizing in 2D

Once the corner detection is completed, a bunch of corner points is obtained, as shown in Fig.4, by separating the image into 16 small windows, so that it can calculate how many corners and edges are shown in each window. All the corner and edge with different values generate a new value for each window, which is used to represent the shape features.

B. Surface Analysis

When generating the data of the shape feature from CAD models, the ray tracking algorithm, which is presented by the Point Cloud Library, is an easy way to generate a point cloud data from a mesh. When projecting a point in 3D space onto 2D plane, the curved surface and the inclined plane have different expression in the 2D plane, as shown in Fig. 5 and Fig. 6. Also as shown in Fig. 7, the red arrows represent the distance between the two points, then intensity of a curvature can be presented as the vector $\vec{P}$, whereas the bending direction of the surface is shown as the angle θ. So by performing this, the differences between the curved surface and the inclined plane can be quantified, and the intersecting line of two inclined planes or the curvature of a curved surface can be easily found. There are plenty of points in each small window, which was defined above. The difference of size between two adjacent vectors $\vec{P}$s, which is called Δp, is the most important condition to determine the logical value for every plane which is represented by different vectors $\vec{P}$s. So according to the Δp and $\Delta\theta$, the feature values can be defined as:

- If the Δp is changing by linearly, and $\Delta\theta < 90°$, then it is a curved surface without a peak, assigning a value as 1.
- If the Δp is always 0, and $\Delta\theta < 30°$ then it is a normal plane, assigning a value as 2.
- If the Δp is changing, and $\Delta\theta < 90°$ in different direction, then it is a curved surface with a peak, assigning a value as 3.
- If there isn't an available $\vec{P}$ or the number of $\vec{P}$ is not enough to reflect the changing of Δp, then assigning as 0.

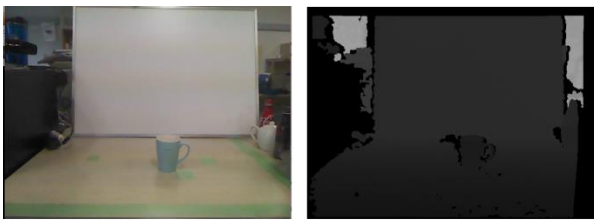


Fig. 8. Real object data capturing environment

III. Real Object Data Processing

The Microsoft Kinect sensor is used to capture the data, which include the RGB data and the depth data. Fig. 8 shows the real object data captured by the Kinect sensor.

For capturing the object data, a large amount of environment data are segmented in two groups, such as background area and object area, by using the distance between the object and camera. The next step is to use the following Euclidean cluster extraction to separate the horizontal plans from the object, which is given by [7]:

- create a Kd-tree representation for the input point cloud dataset P;

p-ISSN: 2356-5314
e-ISSN: 2460-075X

- set up an empty list of clusters C, and a queue of the points that need to be checked Q;
- then for every point p_i in P, perform the following steps:
  - add p_i to the current queue Q;
  - for every point $p_i \in Q$ do:
    i. search for the set P_k^i of point neighbors of p_i in a sphere with radius $r < d_{th}$;
    ii. for every neighbor $p_k^i \in P_k^i$, check if the point has already been processed, and if not add it to Q;
  - when the list of all points in Q has been processed, add Q to the list of clusters C, and reset Q to an empty list
- the algorithm terminates when all points $p_i \in P$ have been processed and are now part of the list of point clusters Q

A sample of object point cloud data after these processing is provided as shown in Fig. 9.

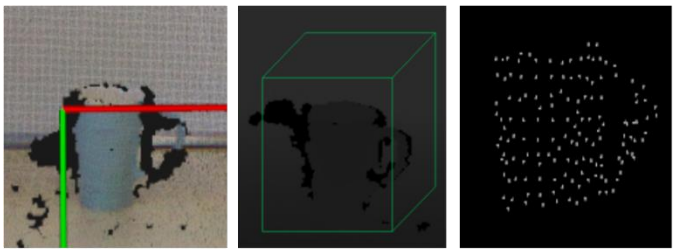


Fig. 9. Capture process

Since a dataset has been created from CAD models, the methods as described above are still used for capturing the shape of real object, the image is changed to grey level and the Gaussian filter is used to create a smooth image. Then, the Sobel operator is used to generate the edge line, where at the last, the Harris corner detection algorithm is used to build a sequence dataset which represent the corner in the image. For the surface condition dataset, the same analysis method as used in the model dataset is applied, where the 16 elements are used to present the evaluation values about the intensity of curvature and the bending direction. Our real objects are shown in Fig. 10.

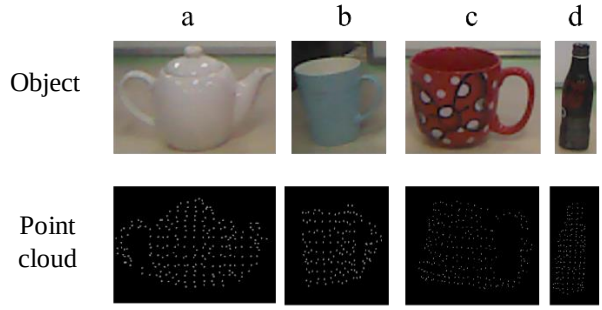


Fig. 10. Captured objects: (a) a teapot; (b) a blue tea cup; (c) a red tea cup; (d) a bottle

IV. Data Structure

The whole dataset for observation includes 3 objects which are generated by using CAD models, consisting of Cup, Teapot and Bottle. By considering the reality, the data in 3 different situations (upright, lateral and inverted) were only generated in 36 angles (every 10 degree in Y-Axis), as shown in Fig. 11. As an example, the first 16 elements present the shape features, and the remaining 16 elements denote the evaluation values about the intensity of curvature and the bending direction in each small window, as shown in Fig. 12.

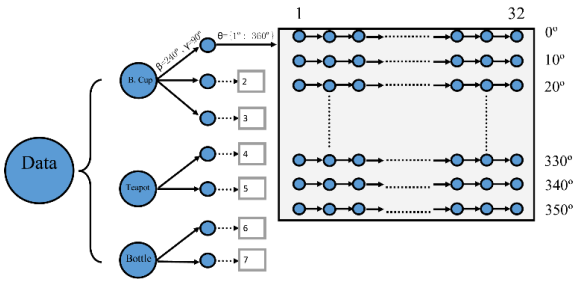


Fig. 11. Observe datasets

Also with the same concept, the real objects with both shape features and surface condition in 36 angles are handled, and these datasets are sent as hidden status in Hidden Markov Model (HMM).

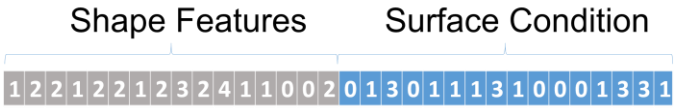
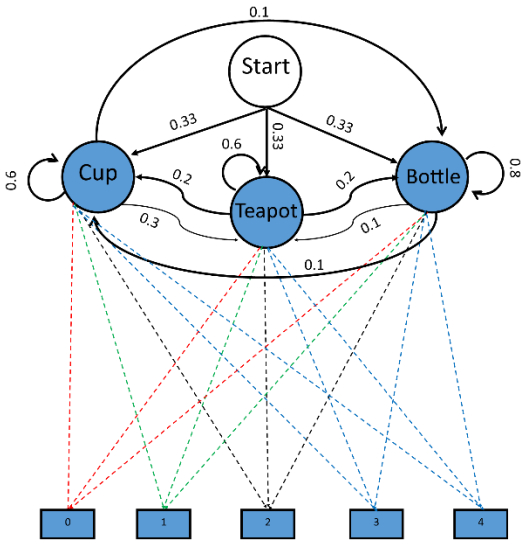


Fig. 12. Data structure

V. Hidden Markov Model

HMM is a well-known probabilistic model of time series data. In our case, the forward HMMs are used as a simple test for the data, as shown in Fig. 13.



p-ISSN: 2356-5314
e-ISSN: 2460-075X

Fig. 13. The HMM structure in our case

The following items comprise a discrete HMM:

a) a set of N states $\{1,...,N\}$

b) an alphabet of M output symbols, $s = \{s_1,...,s_M\}$

c) a set of output probabilities, $B = \{b_{ij} \mid 1 < i < N, 1 < j < s_1,\ldots,s_M\}$

d) a set of state transition probabilities, $A = \{a_{ij} \mid 1 < i < N, 1 < j < s_1,\ldots,s_M\}$

e) an initial start probability distribution, $\pi = \{p_1,...,p_n\}$

As the parameters of HMMs, there are the initial probability $P(X_o)$, the state transition probability $P(X_t \mid X_{t-1})$, and the output probability $P(X_t \mid O_t)$. All these parameters are generated experimentally.

The following two tables show the results of experiments in 100 time for each objects, in which Table I shows the result without surface conditions, whereas Table II shows the result with surface conditions.

VI. Conclusion and Future Work

It has been found out that the surface condition affected not a little the recognition rate of a 3D object. In particular, if the most basic HMMs were used, the result still was able to prove that the surface condition was a very important in object recognition. In the future we will apply the data processing methods in other construction of HMM or other recognition methods.

TABLE I. Result Without Surface Condition

Times(*n*/100)	Cup	Teapot	Bottle
Cup	76	17	7
Teapot	17	69	14
Bottle	6	10	84

TABLE II. Result With Surface Condition

Times(*n*/100)	Cup	Teapot	Bottle
Cup	89	6	4
Teapot	12	76	12
Bottle	2	4	94

References

[1] Bebis, S. Louis, and S. Fadali, "Using genetic algorithms for 3d object recognition," in 11th International Conference on Computer Applications in Industry and Engineering, Las Vegas, USA, Nov. 1998, pp. 13–16.

[2] D. Lowe, "Object recognition from local scale-invariant features," in The Proceedings of the Seventh IEEE International Conference, Kerkyra, Sep 20-27 1999.

[3] K. Lai, L. Bo, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," EEE International Conference on Robotics and Automation (ICRA), pp. 1817–1824, May 9-13 2011.

[4] K. Lai, L. Bo, X. Ren, and D. Fox, "Consumer depth cameras for computer vision," Consumer Depth Cameras for Computer Vision, p. 167, 2013.

[5] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," Robotics and Automation (ICRA), 2014 IEEE International Conference on, May 31- June 7 2014.

[6] D. Salomon, Data Compression: The Complete Reference, 2nd ed., 2000.

[7] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 2011.

[8] J. Hornegger, H. Niemann, D. Paulus, and G. Schlottke, "Object recognition using hidden markov models," Proc. Pattern Recognition in Practice, no. Informatik 5, pp. 1–9, 1994.

[9] GHAHRAMANI and ZOUBIN, "An introduction to hidden markov models and bayesian networks," pp. 9–42, 2001.

[10] C. Harris and M. Stephens, "A combined corner and edge detector," in In Proc. of Fourth Alvey Vision Conference, 1988, pp. 147–151.