Primitive Based Stereo for the Can-Picking Robot

Hironobu Takahashi Hideo Hirono Hajime Terasaki
Sanyo Electric Co. Ltd., Tsukuba Research Center
2-1 Koyadai Tsukuba Ibaraki 305 Japan
Fumiaki Tomita
Electrotechnical Laboratory
1-1-4 Umezono Tsukuba Ibaraki 305 Japan

ABSTRACT

This paper presents a stereo vision system that uses image primitives as stereo correspondence units in order to reduce the computational errors. This system, integrated into a "hand-eye" robot system, is able to recognize cylindrical objects, such as cans, so as to have them picked up by the robot.

1. Introduction

The system we are developing is a "hand-eye" robot system using stereo vision in the context of autonomous task planning. Stereo vision will be the most common vision for intelligent robots in the future because of its flexibility and versatility. However, becasue of difficulty to find correspondence between the images derived from the left and right eyes, stereo vision remains behind active rangefinders, as far as computing cost and reliability are concerned. The cause of the difficulty is that usual methods use too simple features for units of correspondence such as edge points. To solve the problem we should use higher-order features for units of correspondence. Higher-order features imply less calculation, due to a smaller amount of data, as well as more reliability because of sufficient constraints on each of the units. From this consideration, we propose a primitive based stereo vision.

The objects whose features are known can be described through layered relationships, such as 'parallelism' and 'symmetry', as shown in figure 1. Units such as lines, ellipses, cans etc., are called image 'primitives', and objects in the image can be recognized and distinguished qualitatively by tracing the tree from bottom to top [2-3]. Also, reliable quantitative data can be determined by matching the primitives between stereo images.

In this papaer, we propose to recognize ellipses as primitives in the image which represent circles in the 3-dimensional world because circles are ~~not only a~~ important cue~~s~~ of many objects such as juice cans. Experimental results for our can-picking robot system are shown as a direct application of the primitive based vision system.

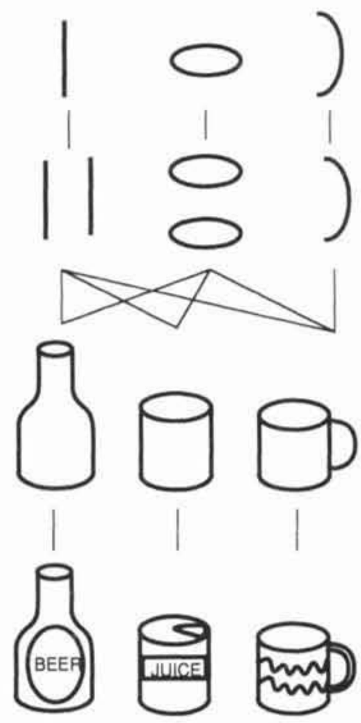


Fig. 1 Image Primitive

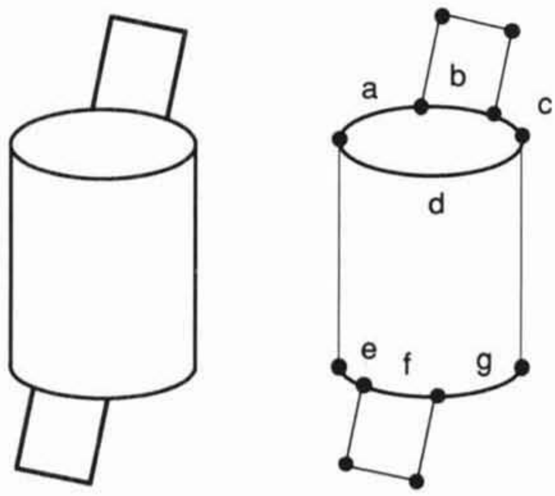
2. Extraction of Primitives

Each image is segmented into regions by an edge detection method, where each region is represented by a list of one external boundary and some boundaries for the holes. Then, each of these boundaries is cut into segments whose vertices are the branching points or the points that maximize the curvature.

Each segment is approximated by either an elliptic curve or a straight line. When the fitting error is larger than the quantization error, the segment is further cut and approximated by new curves or lines. For example, the boundaries shown in figure 2a are cut into the segments shown in figure 2b.

2.1 Set of Segments

The next step is to find out a set of segments representing an ellipse, such as (a,b,c,d) and (e,f,g) in figure



(a) (b)

Fig. 2 Segment

2. However, it is too time consuming to test all possible sets, since an ordinary image consists of 100 to 1,000 segments and the number of such sets would range from 2^{100} to 2^{1000}.

In order to reduce the number of sets, segments are sorted in descending order according to the length, and the "larger" segments are tested in priority. Though a small segment such as segment 'c' in the set (a,b,c,d) may fail to be found by this method, the whole ellipse can be determined successfully as a set (a,b,d).

Even with a limited number of sets, processing an image can take a quite different amount of time, due to the differences in image complexity. This is a big problem for real robot control, and the total searching time needs to be kept within the limit of a few seconds.

In this system, a limit on processing time is assigned to each level of searching, so that computation stops when elapsed time exceeds this value. The top level loop, which searches for the first segment candidate, is allowed to use U second; the next level, which searches for the second candidate, is allowed to use $U \times k$ $(0 < k < 1)$. Then the n'th level is allowed to use $U \times k^{n-1}$. Thus, the worst case of computational time can be expressed by the following equation:

$$TM = U(1+k+k^2+k^3+k^4 \cdots) = U / (1-k)$$

The details of the algorithm are described as follows:

1. Sort segments in descending order according to the number of their points. S(1), S(2), ... , S(N) represent each segment, where N is the number of segments.
2. Initialize parameters as follows:

Time limit	$U(0) := TM \times k$
Segment list	L(0) := [null list]
Range of segment search	R(0) := 1

and call the subroutine Search(0).

Subroutine Search(n):

1. Set i to R(n).
2. Append S(i) to L(n) and copy to L(n+1).
3. Check if all the segments included in L(n+1) belong to only one ellipse.

If they do:

a. Record the set of L(n+1) as a candidate for an ellipse.

b. Set parameters as follows:

$U(n+1) := U(n) \times k$

$R(n+1) := i + 1$

c. call Search(n+1).

4. Increment i.
5. If the time elapsed in the subroutine exceeds the limit then U(n) or i is larger than N, return from the subroutine.
6. go to step 2.

2.2 Valid Ellipses

The system decides whether the segments in a set belong or not to one ellipse through the following procedure.

First, we calculate the equation of the ellipse candidate defined by the segments (C_1, C_2,,, Cn), using the least square method with the following equation:

$$\sum_{k=1}^{n} \sum_{p \in C(k)} (ap_x^2+hp_xp_y+bp_y^2+dp_x+ep_y-1)^2$$

If this equation represents an hyperbolic or parabolic curve or if a straight line is given as a result, the current set of segments is rejected.

Otherwise, if the distance between the ellipse and each point of all the segments is smaller than the quantization error, then the current set is considered to define an ellipse.

This computation involves solving 4th order equations for a large amount of points, and is therefore a time consuming process which becomes the bottleneck of the whole procedure.

Thus, in order to speed up this process, we first calculate the equations of larger and smaller ellipses within the quantization error σ, as shown in figure 3, and described as follows:

$$E_1(x, y)=a_1x^2+h_1xy+b_1y^2+d_1x+e_1y-1=0 \quad (a_1,b_1 > 0)$$

$$E_2(x, y)=a_2x^2+h_2xy+b_2y^2+d_2x+e_2y-1=0 \quad (a_2,b_2 > 0)$$

Then, the condition that the point (xp, yp) exists between these ellipses is expressed by the following equations:

$$E_1(x_p, y_p) > 0, \quad E_2(x_p, y_p) < 0$$

Even though the above condition is not strictly equivalent to the result obtained through distance calculation, the difference is small enough, even if the

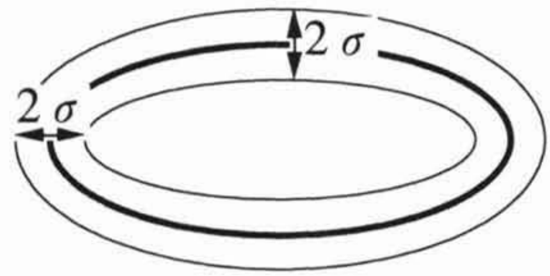


Fig. 3 Larger and Smaller Ellipses

ellipse is extremely elongated.

2.3 Orientation and Center of Valid Ellipses

The orthogonal projection is often used as an approximation of the perspective projection. In this paper, on the other hand, all calculations are done under the perspective projection, since the accuracy is the important factor for robot vision systems and we knew that the error caused by the orthogonal approximation is too large through the real experiments.

As a quadratic curve in the 3-dimensional space is projected as a quadratic curve on an image plane, an ellipse in a space that does not cross the image plane is projected as an ellipse. Then a circle is projected as an ellipse as well, and the orientation of the plane it lies on can be chosen from two possible solutions, as shown in figure 4.

The center position of a circle does not match those of the ellipse and there is one solution for each plane orientation.

3. Stereo Matching

Other parameters of the circle except for the orientation, such as the absolute position and the size, can not be decided from only one view, and misunderstanding of some curve in 3-dimensional space as an circle must be corrected.

In order to choose real circles and decide their parameters, we match circle candidates derived from each image under the following conditions:

a. Calculate the position of the circle center so as to minimize the distances from the epipolar lines to the focal point and the estimated ellipse center. This distance must be smaller than a certain threshold reflecting the quantization error in each image.

b. Both circle candidates must have the same orientation. This condition contributes to not only selection of correct stereo correspondence but also choice of circles from other curves.

c. Calculate the radius of the circle from the distance from each focal point to the circle center and the length of the long axis of the ellipse in each image. Both values must be the same.

The last two conditions above are actually overconstrained by using ellipses as correspondence units, therefore increasing the reliability of the stereo.

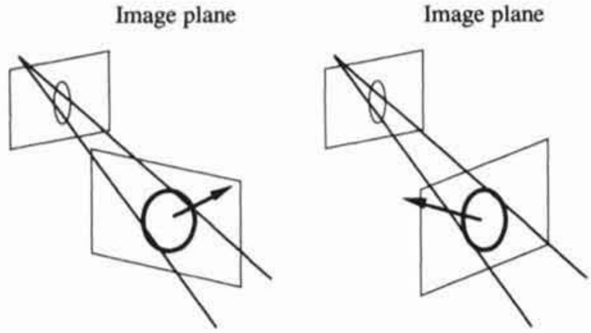


Fig. 4 Direction of Circle

4. Experimental Results

4.1. Image Processing

Color stereo images (480x640 pixels) in fig.5a are input into a SUN3/260 via the frame buffer. Image processing and primitive recognition are performed on a Decsystem 5820 (about 20 MIPS). Each image is segmented into regions by an edge detection method, and small regions are eliminated (fig. 5b). The above procedure takes about 180 seconds.

Figure 5c shows the results of the primitive extraction procedure in the case where the CPU time limit for the method described in section 3.1 is set to 30 seconds. Figure 5d shows one example of a result of stereo correspondence where the position, radius and orientation of the circle primitive can be measured.

4.2. Task Planning

All circle candidates are passed on to the robot planning system (implemented in EUS Lisp [4]). The information given for each circle includes not only the geometrical data but also the probability R defined by the ratio of the portion seen in camera images and the estimated circle. The system tries to find circles that have the same axis and gather them into a group. If such groups are located, they are considered to represent the body of a can. If not, the most probable circle (which have the largest value of R) is chosen instead.

Then the identified can may be grasped and lifted up by the parallel-finger jaw of the robot, and finally transferred to and dropped into a box.

5 Conclusion

In addition to the reduction of the number of units for correspondence, the proposed method has several advantages. First, the ellipse candidates for stereo correspondence can be selected using more constraints, such as the orientation of their axis and the size of their

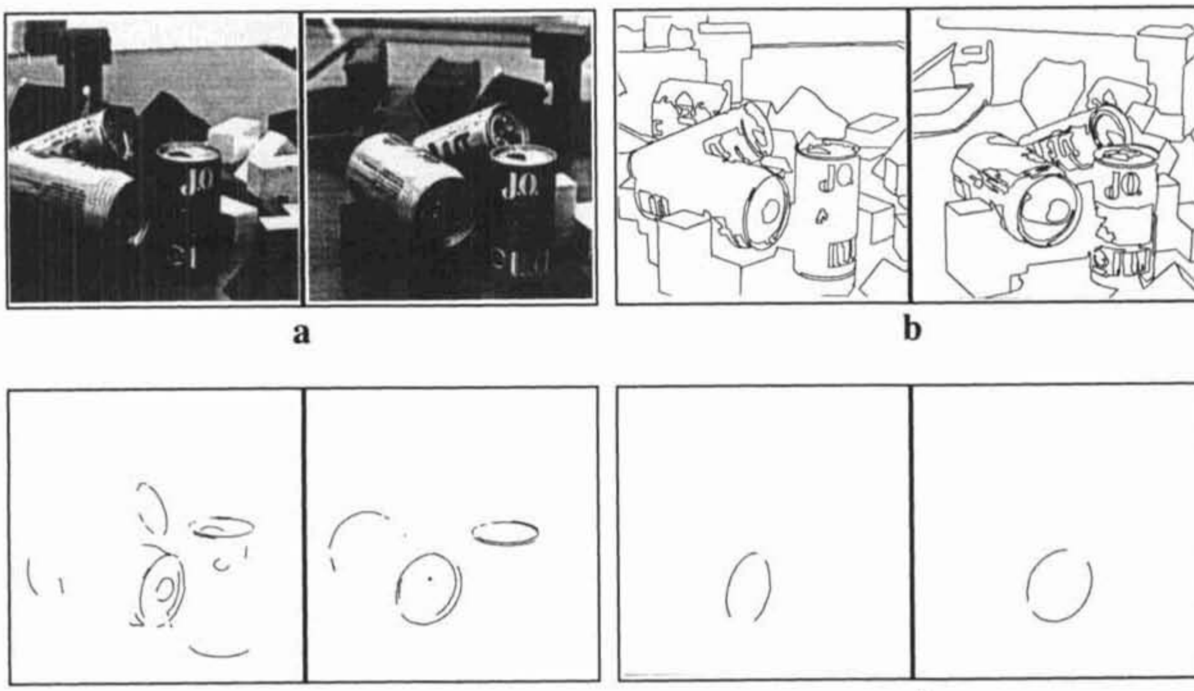


Fig. 5 Experimental Results

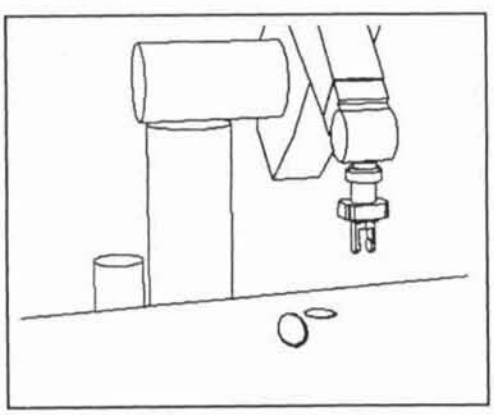


Fig. 6 Can Recognition

Fig. 7 Robot Motion

radius in addition to the epipolar constraint, easily excluding erroneous correspondences.

As shown in the experimental results, this system can recognize and distinguish textured cans present among other objects in an unstructured environment.

Finally, this recognition method for ellipses and cylindrical objects can be applied to many other objects besides cans, such as road signs, cars, cups and so on so forth.

Acknowledgments

The authors are grateful to Dr. M. Pasquier for comments and criticisms which helped to improve the readability of this paper.

References

[1] Terasaki,H.,Asaoka,T.,et al, Automatic Assembly with Vision: Execution by Error Revision, Proc. 30th SICE, 1991, pp. 433-434

[2] Koizumi,M., Tomita,F., Qualitative and quantitative Matching of Solid Models and Images of 3D Objects, Proc. 9th ICPR, 1988, pp. 681-684.

[3] Takahashi,H., Tomita,F., Large Scale Stereo Vision, Proc. 11th ICPR, 1992, pp 628-631

[4] Matsui,T., Inaba,M., EusLisp: An Object-Based Implementation of Lisp, Journal of Information Processing, vol 13, no. 2, 1990