OBJECT RECOGNITION USING MULTIPLE VIEWS

Hwang-Soo Kim, Ramesh C. Jain and Richard **A.** Volz

Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor. MI 48109

Dear Professor Volz;

It is my great honor to be one of your students. You have been always good to me, encouraging even when I brought up a silly idea! I learned a lot from you, especially the generous helping mind for others. My best wishes to you and Mary's health!

H. S. Kim
School of CSE
KNU, Korea

IEEE International Conference of Robotics and Automation, March 25-28 1985

OBJECT RECOGNITION USING MULTIPLE VIEWS*

Hwang-Soo Kim, Ramesh C. Jain and Richard A. Volz

Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109

ABSTRACT

A new approach to model based object recognition employing multiple views is described. The emphasis is given on the determination of camera viewpoints for succesive views looking for distinguishing features of objects. The distance and direction of the camera are determined separately. The distance is determined by the size of the object and the feature, while the direction is determined by the shape of the feature and the presence of the occluding objects.

1. INTRODUCTION

Two major issues in computer vision have been the problem of object recognition and determination of its position and orientation (or pose). There have been extensive research works on object recognition and some works on determination of position and orientation of objects using various techniques. However, most of them implicitly assume that they can recognize and/or determine the orientation of an object with a single view. This is not true for many objects. Most objects which are almost symmetric – that is, symmetric object combined with some asymmetric parts – fall into a class which can not always be handled by vision systems employing single view.

Jain[Jain85] has classified the domain of the vision systems according to the relation between the camera and the scene as follows.

1. Stationary camera, stationary objects (SCSO).
2. Stationary camera, moving objects (SCMO).
3. moving camera, stationary objects (MCSO).
4. moving camera, moving objects (MCMO).

Our problem belongs to MCSO which has received little attention in the past. We assume that we can place the camera at any desired position and direction to look at *distinguishing features* [1] to obtain information on object's identity and orientation, where we define distinguishing features to be those object features which enable us to recognize or determine the orientation unambiguously. The features are usually the asymmetric elements of the object in the class described above.

The notion of using more than one image to recognize an object and its pose raises the following questions :

1. *"How can we recognize and determine the orientation of the object with minimum number of image analyses ?"*
2. *"What should be the next viewpoint of the camera to take the most informative picture ? "*.

This paper addresses above problems.

2. PROBLEM AND APPROACH

2.1. The Problem

We classify objects classes according to the capability of recognition and orientation determination.

1. *Decidable objects* are those which are unique from every viewpoint so that the orientation as well as the identification can be determined from any view.
2. *Undecidable objects* are those for which the *precise* orientation can not be determined (due to its symmetry), no matter how we select the viewpoint and no matter how many views we consider. Note that although the "precise" orientation can not be determined, the orientation can be determined to within an *equivalent class*. For almost all purpose, this is sufficient and in a sense, this class is similar to the class of decidables.
3. *Semi-decidable objects* are those for which the identity and/or orientation can be determined when *seen from some viewpoints, but not determinable* when seen from other viewpoints.

   Examples are a cup with a handle and a fan blade with fixing screw hole. Yet another example is cubes with holes on one face. Symmetric objects to which other parts have been added to break their symmetry fall into this class. In most cases, these asymmetric features play some essential roles in the function of the objects, so the determination

*This work has been supported by AFOSR.

[1] We will use "feature" and "distinguishing feature" interchangeably in this paper.

CH2152-7/85/0000/0028$01.00 © 1985 IEEE

of the orientation and location of the features are crucial part of object recognition. For some objects, even though the orientation can not be determined precisely, an orientation range might be determinable. For example, range of the orientation of a cup with a handle can be determined when the handle is not visible.

Now the problem can be stated as follows:
Assuming that

- We have the models of objects in a database,
- An object recognition process (ORP) which is capable of identifying, locating and determining orientation (when possible) of objects is available,
- We can place the camera (more generally, imaging device) in any position and direction,
- The previous analyses of the image failed to give enough information,

determine the orientation of the object by employing multiple views. We address this problem by succesively determining the camera position and direction to see the features more effectively using the model of the objects .

2.2. General Approach

We address the class of semi-decidable objects. We assume that the ORP returns the identity as a set of objects which could match the views obtained thus far and the range of location and orientation for each object in the set. We also consider the presence of the obstacles.[2]

In one case we assume that we know the object identity and its location but not its orientation. We determine the possibly invisible portion of the surfaces of the object where the features are located (called the feature surface hereafter). Then we select a distinguishing feature and determine the desirable position and direction of the camera. When the identity is given as a set, we assume that the object is one of the set. We determine the visibility of the feature, and the camera position and direction can be determined as before. The size of the set can be reduced after each view. The number of trials may depend on the order of the object we assume as well as the viewpoint we select. It is desirable for the ORP to return the certainty factors (confidence values) of the identity for each object. In that case, we can try in the order of certainty to reduce the number of trials.

When the identity (or a set identity) is known, but orientation is totally unknown, we should use blind search until we get some orientation information. One good first viewpoint is the one at the opposite side from the current position. If we can interact with the ORP and the high level and low level process of the ORP can interact with each other, we may ask the ORP whether some predicted image shape is there or not. Then the assumption on the orientation of the object can be made more reasonably. One way of doing this is to match the image with the visual aspect graph (VAG) to get reasonable orientation information.

When we consider occlusion, we need to consider two possible reasons for invisibility of the feature

(a) the feature is on a side of the object hidden from the viewpoint or

(b) the feature is on visible side but occluded by obstacles.

Case (a) can also involve occluding objects which are behind the object from the observer. These are discussed in sec.4. In case (b), we can compare the size of image of occluding object and the feature size predicted by the model in perspective view using the known distance. If the latter is bigger, there is no possibility of the feature being at visible side.

Our general approach is to determine the distance and direction of the camera separately, although they are not completely independent of each other. This approach is valid since the distance is related to the image size of the object and the feature while the direction is mostly related to the selection of the viewpoint that allows informative view of the feature avoiding occlusion. By this separation, the problem becomes easier in most cases. We determine the distance first, and then the direction. The determination of camera distance is discussed in more detail in section 3. Two methods for determining the camera direction are presented, the VAG method and the projection method. The desirable direction is a direction which makes the feature directly visible and occluded as little as possible. It is discussed in section 4.

As we get more views, more information can be gathered for unidentified objects. We can also reduce the set of possible objects returned by ORP as we get more views. One possible way is to successively verify and eliminate those that are inconsistent and put more constraint on objects as we get more accurate information about the objects. This process should be repeated until all objects are identified, located and oriented. There may be unidentifiable objects even after all these steps. For example, for the objects in a similar-objects class, their distinguishing feature may be face-down on a table. In such a case, we may need to use a robot to turn them upside down.

2.3. Related Works

The problem of object recognition using multiple views has not received much attention so far and there are few papers related to this. Some related works are described

[2] In this paper, *object* is the one to be recognized, other objects are called *obstacles*.We use the term "obstacle" and "occluding object" interchangeably.

below.

Brooks[Broo81] used the term *geometric reasoning* to mean making deductions about spacial relationship of objects given description of the position, orientation and shape of the objects. He used a symbolic algebraic manipulator called constraint manipulation system (CMS) to predict whether an object is visible at all or not. The possible reasons of invisibility considered were improper camera direction and occlusion by other objects. His concern was the matching of the given image to a model and used the prediction to verify some features to support the matching.

The visibility problem is addressed in many papers, in various areas ([Avis81], [Tous80], [Davi79]). However, most of them deal with the internal visibility of 2-D polygonal world. The hidden surface removal problem is one of the classic problems in computer graphics. An excellent survey is given in [Suth74]. The hidden surface finding algorithm is used in graphical prediction in [Scot84].

Koenderink and Van Doorn[Koen79] addressed one aspect of visibility as *visual potential*. It is a connected graph where the nodes represent *visual aspects* (an aspect is a set of views in which the images have the same topology) and the edges represent *visual event*. Castore and Crawford[Cast84] call it *aspect graph*. We will call it visual aspect graph (VAG) in this paper. It is a graphical representation of various views differnt in topology. [Cast84] discussed how to generate it and partition the space into cells (*parcellation* in their term) corresponding to the node of the graph for simple objects, but the problem of generating it for reasonably complex object and associating it to the viewing space need to be developed more. Ikeuchi[Ikeu83] used *extended Gaussian image* (EGI) to represent an object and used it to determine the attitude of it. In the EGI, each object surface is mapped onto a sphere with the direction of its normal vector and magnitude proportional to its area. Fekete and Davis[Feke84] proposed a spherical representation called *property sphere (PS)* which is used to represent properties of object seen from different viewpoints. These spherical representations are convenient to choose the viewing direction. We propose to associate the visual aspect graph to the viewing sphere similar to the property sphere of [Feke84] to get the orientation information and choose the viewpoint.

The idea of using multiple views to get more information is not new. The Stereo and motion are the examples, but they are not related to our work. However, more relevant works have appeared recently. Among others, Herman et. al.[Herm84] discussed how to build the 3-D model of the city buildings from multiple views using domain specific knowledge. Martin and Aggarwal[Mart83] describe building volumetric models of 3-D objects using silhouettes of multiple views.

3. THE CAMERA DISTANCE

In this section, we want to determine optimal camera distance from the object by considering some simple objects and deriving some relations between the size of the object, the distance, the visible portion of object and maximal number of views we need to take. First, we consider the number of views and the distance required to see an entire object. Then we consider the distance to properly see a feature on an object. We consider sphere and arbitrary-shaped solids here. Some other objects are discussed in our forthcoming report.

3.1. Sphere : The simplest object

The sphere is the simplest 3-D object. We can get some insight to the problem by considering this simple object. Moreover, the results can be extended to other objects.

Let the radius of the sphere be r, the distance between the camera and the center of the sphere be R and the distance of the image plane be d. To take the foreshortening[3] effect into account, let us define *safety angle* α as in figure 1. Note that $-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$. The angle β in the figure is termed as *prospect angle* in [Ikeu83] and we will call γ as *viewing angle* . It can be shown that

$$\frac{R}{r} = \frac{\sin\alpha}{\sin(\alpha-\beta)} \tag{1}$$

or

$$\beta = \alpha - \arcsin(\frac{r}{R}\sin\alpha). \tag{2}$$

Note that when R $= \infty$, $\beta = \alpha$. The visible area A_v is given by

$$A_v = 2\pi r^2(1-\cos\beta) \tag{3}$$

and the solid angle ω of visible area determined by β is given by

$$\omega = 2\pi(1-\cos\beta) = 2\pi(1-\sin\gamma). \tag{4}$$

Unlike 2 dimensional circle, we can not cover whole surface area of a sphere by n views even though we can see $1/n$ of the total area by a view, since we see a round dome shaped surface from a viewpoint and there is no way to partition the surface of a sphere with the dome other than the hemisphere. They should overlap. If we call the set of visible points on the surface from viewpoint i to be V_i and $\bigcup_{i=1}^{n} V_i$ = whole surface, $V_i \cap V_j \neq \emptyset$, for adjacent viewpoint i, j.

[3] When a planar surface of area A is seen at an angle α from the surface normal, then the apparent area of it is given by $A\cos\alpha$. This effect is called *foreshortening*.

The required distance for some given number of views can be calculated using above equations and the results are summarized in table 1. We can see from the ratio $\frac{a}{b}$ in the table that as we use more views, the required distance is less dependent on α.

No. of views	β required	a=$\frac{R}{r}$ ($\alpha = 75°$)	b=$\frac{R}{r}$ ($\alpha = 90°$)	$\frac{a}{b}$
2,3	90.0	impossible	∞	-
4	70.53	12.4	3.0	4.13
5	63.44	4.82	2.24	2.16
6	54.74	2.79	1.73	1.61

Table 1. Number of views and the required distance

3.2. More General Objects

For more general solids, we can determine the approximate distance by using the enclosing sphere. This would not work well for much elongated object, in which case we should consider the feature surface and feature size, as discussed later. For polyhedral objects, the visibility of the faces can be determined as follows. Let $\vec{n_i}$ be the normal vector of i-th surface, the *center* of that surface be $C_i(x_i, y_i, z_i)$ and $\vec{n_P} = \frac{\vec{C_iP}}{|\vec{C_iP}|}$ where P is the viewpoint. Then the i-th surface is visible if $\vec{n_i} \cdot \vec{n_P} > 0$.[4] These arguments assume that the diameter of each surface is much smaller than R so that foreshortening effect is not significantly different from the center of the surface to the corner of the surface. For a tetrahedron, prism and rhomboid, proper choice of viewpoint allows 50% or more visible area while worst choice allows only one face. Alternatively, if we have the visual aspect graph, then we can identify the visible surfaces immediately and we can determine the distance using the surface sizes associated with the node. We can also determine the number of views necessary to cover whole surfaces.

For general curved convex solids, we can compute the visibility as follows. Let $\vec{n}$ and unit vector $\vec{v}$ be defined as in figure 2. Then the surface patch ds is visible if $\vec{n} \cdot \vec{v} > \cos\alpha$. So the visible area A_v is given by

$$A_v = \int_{surf} D(\vec{v}, \vec{n}) ds$$

where D is deciding function such that $D(\vec{v}, \vec{n}) = 1$ if $\vec{n} \cdot \vec{v} > \cos\alpha$ and $D(\vec{v}, \vec{n}) = 0$ otherwise. For concave objects, we must take into account the occlusion by the object itself, since ray of sight $\vec{qP}$ can be blocked by other part of the object when looking at concave point even though the $\vec{n} \cdot \vec{v} > 0$ so the D function above should be modified as $D(\vec{v}, \vec{n}) = 1$ if $\vec{n} \cdot \vec{v} > \cos\alpha$ and $\vec{qP}$ isn't blocked by other part of the object, otherwise 0.

3.3. Feature Size Requirements

There is typically some restriction on minimum relative feature size and it constrains the allowable distance. Let S be the resolution of the image (the number of pixels along one dimension assuming that vertical and horizontal resolution is same), L be the larger side length of the object surface on which the feature is located, f be the feature size in length, p be the minimum number of pixels required to interpret correctly. Note that p is dependent on the shape and it should take into account the foreshortening effect. There could be another $p' < p$, by which we can not interpret correctly but we can detect that there is something which might be the feature. The minimum feature size required to be able to recognize correctly is given by

$$\frac{f}{L} \geq \frac{p}{S}$$

which means that when we make the largest image fit onto the screen, the feature should be recognized. In the case of the circle (or sphere), $L = 2r$, $S = c \cdot \frac{2r}{R} d$, where c is a constant (number of pixels per unit length). So the feature should be

$$f \geq \frac{pR}{cd}.$$

4. THE VIEWING DIRECTION

After we determined optimal distance of the camera, we select the direction of the camera using the model and available orientation information to see some distinguishing features. In order to do this, we need some method to represent the probable region of the space in object-centered coordinate system. When there are no obstacles, the problem is simpler and we only need to pick a direction which will show the feature. This can be done once we have description of the feature surface portion. The VAG method and projection method are proposed to select the direction.

The general procedure to determine the direction is as follows

1. Identify the unexplored portion of the feature surface.
2. Represent it in proper method.
3. Take obstacles between the object and the camera into account if any.
4. Select the most unoccluded region about the point of interest or desired angle with the feature.
5. Verify goodness of the selected viewpoint by hidden surface method. If it is not good enough, that is, if it is predicted that we can not see the feature

[4] Considering the foreshortening effect, $\vec{n_i} \cdot \vec{n_P} > \cos\alpha$.

well enough, select another viewpoint and repeat above procedure.

4.1. VAG method

When the VAG is available or computable, we can select the direction as follows. We identify the nodes of the VAG and their corresponding 'cells' of spherical space which shows the feature as desired. Then the possible areas to see the feature are those areas swept by the cells when we transform them according to the orientation range of the object. We can select the viewing direction in those areas which will show the feature best. Note that we can assign 'goodness' of the view to the nodes of the VAG, so we can select a point which is in the area swept by the 'best' node cell. The aspects vary with the radial distance, requiring computation of parcellation of spherical space each time we select the distance. This problem can be alleviated by pre-computing the optimum distance for the feature, use it as radius of the viewing sphere and parcellate it. We can take the obstacles into account by a method similar to the projection method described in next subsection.

4.2. Projection Methods

In order to select a viewing location that is most probable to see the feature effectively in an environment with several occluding objects, the classification of the space into 2 classes of regions, the regions where we can see the feature and the regions where we can not, is necessary. The *projection* method is proposed here to sort the space into occluded and unoccluded regions. The procedure procedes as follows.

As we want to see the feature surface that was not shown so far, we are going to project the invisible portion of feature surface on spherical or cylindrical *screen.*
The radius of the screen is determined by the desirable camera distance, its center is at the object center and the height of the cylindrical screen is determined by the height of the model. The choice between spherical or cylindrical screen is made according to the shape of the object. The next step is to project the occluding objects which are between the object and the screen, if there are any. Only those which affect the projected portion of the feature surface need to be considered.

There are two projection methods, called eclipse method and point method as shown in figure 3. The eclipse method is more accurate description in the sense that it distinguishes whether the object is completely or partly occluded. The disadvantage is the complexity of the computation. The point method is very simple, yet it gives reasonably good estimate of how much the feature will be occluded.

To represent arbitrary 2-D shape of occluded area, the quadtree may be used since (a) quadtree can represent any 2-D shape and (b) it is easy to pick large unoccluded area in quadtree, since we only need to choose a white node close to point of interest at higher level. We consider *cylindrical* and *spherical quadtree* here to represent the *cylindrical* and *spherical projection* .

1. Cylinder : cylindrical quadtree may be defined by dividing the height and angle along the axis by $N = 2^n$ where n is the height of the tree and determined by required resolution.
2. Sphere : spherical quadtree can be defined by succesively dividing regions as in figure 4. Note that the center quadrant is bigger than outer ones in terms of solid angle. The ratio of center one to outer one varies non-linearly with the levels of the tree, making it computationally inefficient and the algorithmically complex to handle this.

Among the free area candidates, heuristically the one closer to the projection of the feature may be chosen since it has better chance of seeing the feature more directly, possibly with less foreshortening. The position of the camera is selected as the center of the quadrant and direction is toward the object center from that point. Note that this is not the viewpoint that looks the feature at right angle. The view angle depends on the angle between the normal of the feature surface and the line from center of the object to the feature(radius vector).

The merit of VAG is that we can assign the 'goodness' of the view to the nodes and the disadvantage is the complexity of generating VAG. The main advantage of projection method is its capability of handling the occlusion. The disadvantage is that we have less control on camera direction, we just determine the camera location and the view angle is dependent on the feature surface direction.

5. CONCLUSION

We proposed an extension of single-view-ORP that is capable of handling hard-to-recognize/orient objects in an environment of many obstacles by employing multiple views. In the future, we are going to address the measure of *goodness* of view, and rules to select the good direction, rules to decide the acceptability of chosen point by prediction and the goodness measure, methods to succesively refine the constraints on orientation and set of possible objects as we get more views, use of *feedback* concept to the help of the orientation assumption when the ORP does not return the orientation information, extension of domain to handle realistic objects, obstacle avoidance of camera, automatic determination of features and classification of decidability. MCSO is wide-open research area and there are many interesting problems to be investigated.

REFERENCES

1. [Avis81] D.Avis and G.T.Toussaint, "An optimal algorithm for determining the visibility of a polygon from an edge". IEEE TC v.c30, no.12, Dec.1981
2. [Broo81] R. A. Brooks, "Symbolic Reasoning among 3D models and 2D images". AI, v.17 (1981) p.285-348
3. [Cast84] G. Castore and C. Crawford, "From Solid Model to Robot Vision". Proc. of Int. conf. on Robotics. IEEE 1984.
4. [Davi79] L. S. Davis and M. Benedikt , "Computational Models of space : Isovist and Isovist Fields". CGIP v.11, p.49-72, 1979
5. [Feke84] G. Fekete and L. S. Davis, "Property Spheres: A New Representation for 3-D Object Recognition". Proc. of the Workshop on Computer Vision, Representation and Control". IEEE 1984.
6. [Herm84] M. Herman, T. Kanade and S. Kuroe, "Incremental Acquisition of s Three-Dimensional Scene Model from Images". PAMI v.6, no.3, May 1984 p.331-340
7. [Ikeu83] K. Ikeuchi , "Determining attitude of objects from needle map using extended Gaussian image". MIT AImemo no. 714, Apr. 1983
8. [Jain85] R. Jain, "Dynamic Scene Analysis". Progress in Pattern ,Recognition, v.2, North-Holland 1985 A. Rosenfeld and L. Kanel Eds.
9. [Koen79] J. J. Koenderink and A. J. van Doorn, "The internal representation of solid with respect to vision". Biological Cybernetics, v.32 (1979) p.211-216
10. [Mart83] W.N.Martin and J. K. Aggarwal, "Volumetric descriptions of objects from multiple views". IEEE PAMI v.5, no.2, Mar.1983
11. [Scot84] R. Scott, "Graphics and Prediction from Models". Proc. Image Understanding Workshop, 1984.
12. [Suth74] I.E. Sutherland, R.F. Sproull and R. A. Schumacker, "A Characterization of Ten Hidden-Surface Algorithms". Computing Surveys, v.6, no.1, Mar. 1974. pp. 1-55
13. [Tous80] G. T. Toussaint, "Pattern recognition and geometrical complexity". IEEE Proc. of 5th int. conf. of Pattern recognition, Miami Beach, FL, 1980

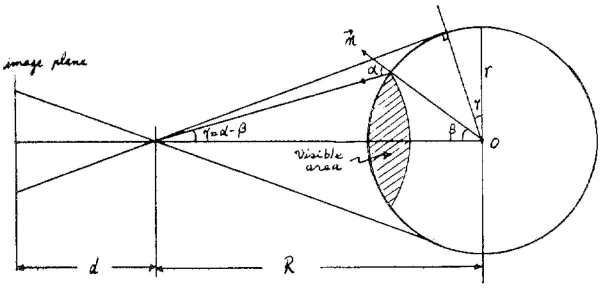


Figure 1: Viewing a sphere

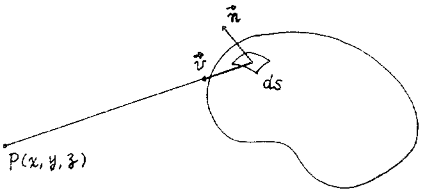
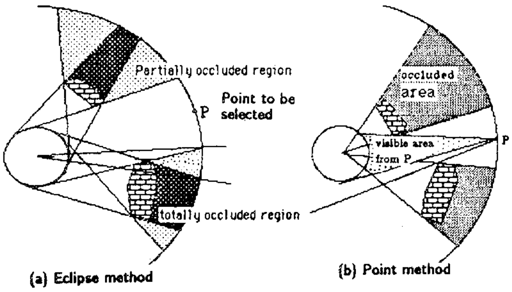


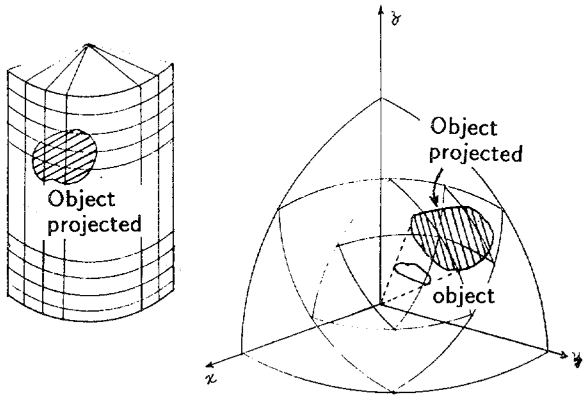
Figure 2: Viewing general solid.



(a) Eclipse method

(b) Point method

Figure 3: Projection



(a) Cylindrical

(b) Spherical

Figure 4: Cylindrical and Spherical Quadtree