Applying Sensor Models to Automatic Generation of Object Recognition Programs

Katsushi Ikeuchi **Takeo Kanade**

Computer Science and Robotics
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

One of the most important and systematic methods to build model-based vision systems is that to generate object recognition programs automatically from given geometric models. Automatic generation of object recognition programs requires several key components to be developed: object models to describe the geometric and photometric properties of an object to be recognized, sensor models to predict object appearances from the object model under a given sensor, strategy generation using the predicted appearances to produce an recognition strategy, and program generation converting the recognition strategy into an executable code.

This paper concentrates on sensor modeling and its relationship with strategy generation, because we regard it as the bottle neck to automatic generation of object recognition programs. We consider two aspects of sensor characteristics: sensor detectability and sensor reliability. Sensor detectability specifies what kinds of features can be detected and in what condition the features are detected; sensor reliability is a confidence for the detected features. We define the configuration space to represent sensor characteristics. We propose a representation method for sensor detectability and reliability in the configuration space. Finally, we investigate how to use the proposed sensor model in automatic generation of an object recognition program.

INTRODUCTION

A large class of practical vision problems includes object recognition, that is, recognizing and locating objects in a scene by means of visual input. Examples of this include visual part acquisition on a conveyer belt or from a bin of parts, target recognition in aerial images, and landmark recognition by a mobile robot. In most of these cases, we have some prior knowledge of the objects of interest, such as the shapes, sizes, reflective properties, and so forth. Model-based vision [1, 2] seeks to actively use such prior knowledge of objects for guiding the recognition process in order to achieve efficiency and reliability.

One of the critical issues in building a model-based vision system is how to quickly extract and organize the relevant knowledge of an object and to systematically turn it into a vision program. One method for increasing efficiency is compiling object models into an object program automatically [3, 4, 5, 6]. That is, the relevant knowledge in the object models is extracted and compiled into an object recognition strategy at compile time so that as little computation as possible is spent at run time. A large portion of the computation needed for using the object model, such as analysis of the best recognition strategy, analysis of occlusion, and estimation of expected feature values, can be done at compile time, and the result can be compiled into the special program. In some cases, the object properties might be represented in the flow of the program rather than its data structure. As a result, the compiled special program to run on-line can be more efficient than generic programs. Also, since the program is generated automatically, the development time could be reduced.

Automatic generation of object recognition programs requires several key components:

- *object models* to describe the geometric and photometric properties of an object to be recognized;
- *sensor models* to predict object appearances from the object model under a given sensor;
- *strategy generation* using the predicted appearances to produce an recognition strategy;
- *program generation* converting the recognition strategy inot an executable code.

This paper concentrates on sensor modeling and its relationship with strategy generation, because we regard it as the bottle neck to automatic generation of vision programs. The object appearances are determined by a *product* of an object model with a sensor model. As shown in Figure 1, the same object model in the same attitude can create different appearances and features when seen by different sensors. Edge-based binocular stereo reliably detects depth at edges perpendicular to the epipolar lines. Photometric stereo or a light-stripe range finder detects surface orientation and depth of surfaces which are illuminated and visible both by the light source and by the cameras.

Thus, in object recognition, it is insufficient to consider only an object model; it is essential to exploit a sensor model as well. On the other hand. modeling sensors for object recognition has attracted little attention; quite often, researchers who are familiar with the sensors they use tended to construct object appearances by implicitly incorporating their sensor behavior. This paper, in contrast, explores a general framework for explicitly incorporating sensor models which govern the relationship between object models and object appearances.

A sensor model must be able to specify two important characteristics: sensor detectability and sensor reliability. The sensor detectability specifies what kind of features can be detected in what condition. The sensor reliability is a measure of uncertainty in the detected features. This paper presents a method for modeling sensors with sensor detectability and sensor reliability, and how to use them in object recognition. We define the configuration space to represent sensor characteristics. Next, we consider two aspects of sensor characteristics: sensor detectability and sensor reliability. We propose a representation method for sensor detectability, and examine how to use the representation for generating object appearances. Sensor reliability analysis consists of determining uncertainty in sensory measurement and analyzing uncertainty

This research was sponsored by the Defense Advanced Research Projects Agency, DOD, through ARPA Order No. 4976 under contract F33615-87-C-1499 and monitored by the Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Aeronautical Systems Division (AFSC), Wright-Patterson AFB, OHIO 45433-6543. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the U.S. Government.

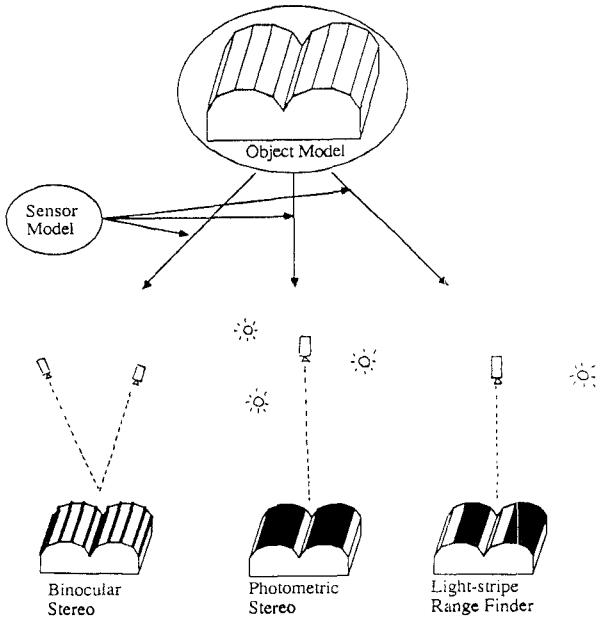


Figure 1: Object-appearances.

propagation from sensory measurements to geometric features. Finally, we investigate the way to use these sensor models for automatic generation of object recognition programs.

SENSORS IN OBJECT RECOGNITION

Various kinds of sensors are used in object recognition. For our purpose, "sensors" are transducers which transform "object features" into "image features". For example, an edge detector detects edges of an object as lines in an image. Photometric stereo measures surface orientations of surface patches of an object. There are both passive and active sensors. Binocular stereo is passive, while a light-stripe range finder is an active sensor using actively controlled lighting. Table 1 gives a summary of various sensors in terms of what object features are detected in what forms.

Table 1 Summary of Sensors

Sensor	Vertex	Edge	Face	active/passive
Edge Detector [33, 24, 6]	-	line	-	passive
Shape-from-shading [14, 17]	-	-	region	passive
Synthetic Aperture Radar [35]	point	point/line	point	active
Time-of-Flight Range Finder [19, 13]	-	-	region	active
Light-stripe Range Finder [1]	-	-	region	active
Binocular Stereo [25, 30]	-	line	-	passive
Trinocular Stereo [28]	-	line	-	passive
Photometric Stereo [36, 16]	-	-	region	active
Polarimetric light detector [23]	-	-	point	active

In addition to qualitative descriptions of a sensor, a sensor model must describe two characteristics quantitatively: *detectability* and *reliability*. Detectability specifies what kind of features can be detected in what conditions. Reliability specifies the expected uncertainty in sensory measurement and geometric features derived from measurement. Since these two characteristics depend on how the sensor is located relative to an object feature, we will first define a feature configuration space to represent the spatial relationship between the sensor and the feature. Then, we will investigate the way to specify detectability and reliability over the space.

FEATURE CONFIGURATION SPACE

Whether or not a sensor detects an object feature and the reliability of this detected feature depend upon various factors which include: distance to a feature, attitude of a feature, reflectivity of a feature, transparency of air, ambient lighting, and so forth. In most object recogntion problems, the attitude of a feature, that is, angular freedom in the relationship between a feature and a sensor, affects sensor characteristics the greatest. In order to specify this freedom explicitly, we attach a coordinate system to an object feature and consider the relationship between the sensor coordinate system and the feature coordinate system. For example, in a face feature, we define a coordinate system so that the z axis of the feature coordinate system agrees with the surface normal and the x-y axes lie on the face, but are defined arbitrarily otherwise. For other features, we can define feature coordinates appropriately. See Appendix I for more details.

Since angular relationships between the two coordinate systems are relative, for the sake of convenience we fix the sensor coordinate system and discuss how to specify feature coordinates with respect to it. The angular relation from the sensor coordinate system to a feature coordinate system can be specified by three degrees of freedoms: two degrees of freedom in the direction of the z axis of a feature coordinate system, and one degree of freedom in the rotation about the z axis. See Figure 2 (a).

Since we will consider the angular relationship, we can translate the feature coordinates so that the two coordinate systems share the origin. We will then define a sphere whose origin is located at the origin of the sensor coordinate system, and whose north pole is on the z axis of the sensor coordinate. We will specify a feature coordinate as a point in the sphere. Referring to Figure 2 (b), the north pole of the sphere corresponds to the case when the feature coordinates are aligned completely with the sensor coordinates. For other feature coordinates, the direction from the sphere center to the point coincides with the z axis of the feature. The distance from the spherical surface to the point is determined by the angle of rotation (modulo 360°) around the z axis from the coordinate on the spherical surface. A point on the spherical surface represents a feature coordinate obtained by rotating the sensor coordinate around the axis perpendicular to a plane given by the sphere center, the spherical point, and the north pole. We will refer to this sphere as the feature configuration space.[1] Figure 2(c) shows various feature coordinates corresponding to spherical points, while Figure 2(d) shows those corresponding to points on a radical axis.

DETECTION CONSTRAINTS and ASPECTS

In the previous section, we defined the way to represent the relationship between sensor coordinates and feature coordinates. In this section, we will develop a constraint to determine whether a feature can be detected at each point of the configuration space.

Detection Constraints

Each sensor has two components: illuminators and detectors. For example, both a time-of-flight range finder and a light-stripe range finder have one light source and one TV camera. Binocular stereo has one light source (without light sources you cannot observe anything) and two TV cameras; photometric stereo has three light sources and one TV camera.

One illuminator only illuminates one part of an object; one

[1]This representation will not create discontinuities around the north pole as opposed to the case in which Euler angles from the sensor coordinate frame to the feature coordinate frame are used to specify spherical points; this representation will instead create discontinuities at the center of the sphere and at the south pole. However, this is advantageous because we mostly use the area around the north pole to discuss detectability and reliability.

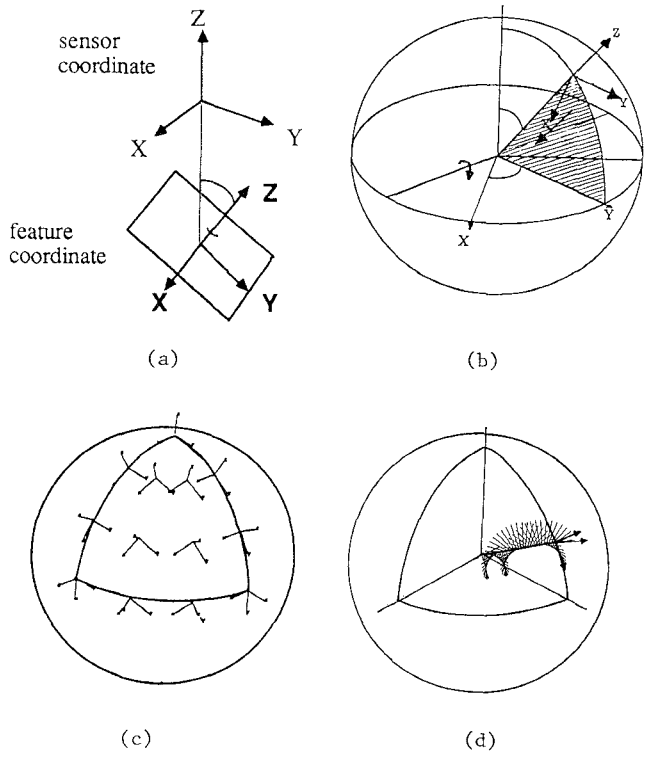


Figure 2: Feature configuration space: (a) A feature coordinate with respect to the sensor coordinate; (b) A feature configuration space; (c) Feature coordinates on the spherical surface; (d) Feature coordinates along a radial axis.

detector only observes one part of the object. Each sensor, which consists of illuminators and detectors, only detects one part of the object. In order for a feature to be detected by a sensor, the feature must satisfy certain conditions on being illuminated by its illuminators and being visible from its detectors.

Once we define a local coordinate system on a feature, we can compute configuration of a feature in which it is illuminated by each illuminator, and configurations in which it is visible by each detector. In the following discussion, we will consider both illuminators and detectors as generalized sources (G-sources). Each G-source has two properties: the illumination direction and the illuminated configurations. In the source case, these two terminologies are the same as the nominal meanings. In the case of detectors, the illumination direction corresponds to the line of sight of the detector, and the illuminated configurations correspond to the visible configurations from the detector.

G-source illumination direction can be represented in the feature configuration space as a radial line from the sphere center. G-source illuminated configurations can be specified as a volume in the configuration space. Finally, we can obtain the constraints in which the feature is detectable by the sensor with AND operations on illumination directions and illuminated configurations of all component G-sources of the sensor.

Figure 3(b) shows an example analysis of a face feature for a light-stripe range finder in Figure 3(a). A light-stripe range finder has two G-sources (a TV camera and a light source): the direction denoted by $V1$ indicates the line of sight of the TV camera; $V2$ indicates the illumination direction of the light source. The illuminated configurations of a face from the light source are determined by the z axis (ie, its surface normal) of a face feature coordinate, and are not dependent on its rotation. Therefore, illuminated configurations of a feature form a spherical cone whose axis is $V2$ and whose apex angle is $d2$. Similarly, the configurations of a feature visible from the TV camera form a spherical cone whose center direction is $V1$ and whose apex angle is $d1$. Since a light-stripe range finder detects the faces which are illuminated from the source and visible from the TV camera, the detectable configurations (necessary condition) are the intersection of the two cones. Thus, the resulting detection constraints in Figure 3(b) contain three constraint components; two illumination directions and one volume of detectable configurations.

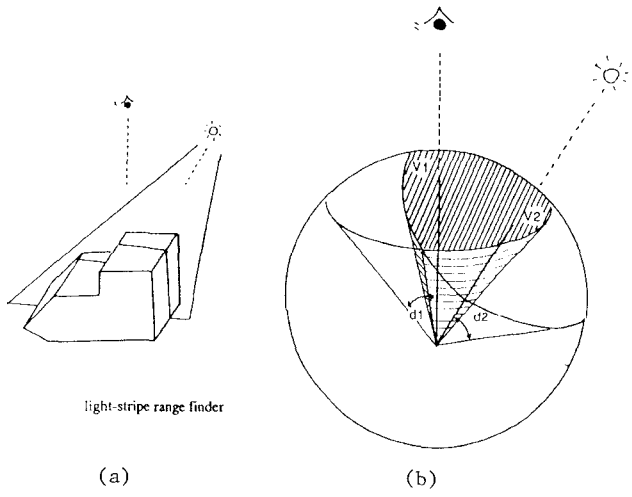


Figure 3: Detection constraints of a light-stripe range finder: (a) Our hypothetical light-stripe range finder; (b) Constraints in the feature configuration space; Note that there are two kinds of constraints; illuminated configurations and illumination directions.

Similarly we can analyze the detection constraints for various sensors in Table 1. The results of the analysis are summarized in [23].

Use of Detection Constraints

In order to predict object appearances, we apply the constraints to each feature of the object. Each feature is detectable by the sensor if it satisfies the following two conditions:

1. None of the illumination directions are occluded by any other parts of the object
2. The detectable configurations contain the configuration of the feature.

To check these conditions we use the constraints with a geometric modeler. We rotate the object into a certain attitude to be examined, and then see whether its features satisfy the previous constraints. Figure 4 illustrates the process of predicting object appearances for a light-stripe range finder. Suppose an object is placed like Figure 4 (a). Figure 4 (b) shows the detection constraints on a face for a light-stripe range finder. We will put this configuration space on each candidate face to examine whether the face is detectable. See Figure 4(c). This amounts to checking the following conditions:

1. The light source direction, $V2$ is not occluded by other faces.
2. The line of sight of the TV camera, $V1$ is not occluded by other faces.
3. The local coordinate of an face, defined by the surface normal (z axis) and the tangential plan (x-y axis), is contained in the detectable configurations.

Figure 4(d) shows the result of this operation. The shaded areas indicate those which satisfy the conditions and thus are detectable by the light-stripe range finder.

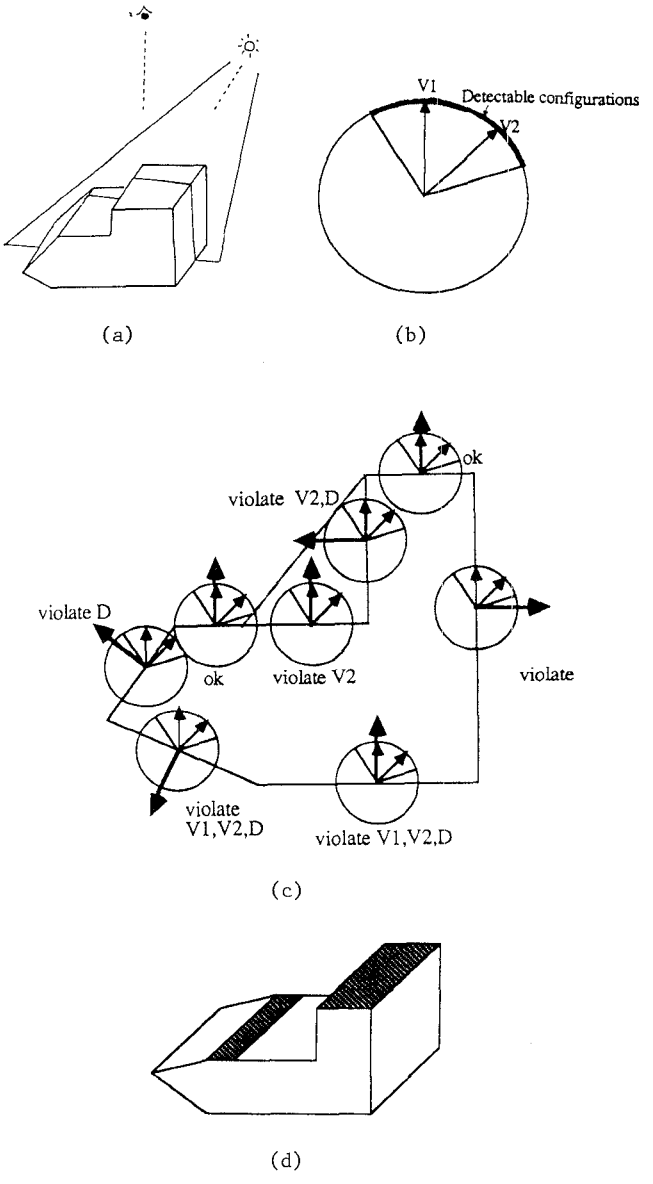


Figure 4: How to use the constraints: (a) A light-stripe range finder; (b) The detection constraints; (c) Applying the detection constraints to object features; (d) Detectable faces determined.

Appropriate descriptions of object appearances must be defined so that they can be used in automatic generation of object recognition programs. The description of an object appearance should include a set of visible faces, a set of the expected feature values. A geometric modeler generates a possible appearance of an object under a given attitude. We will convert output data from the geometric modeler into representations in Framekit+. One appearance, for example *I0* in Figure 5, is represented by one frame, which points to several appearance component frames representing visible 2D faces, *IMAGE-COMP01*, and *IMAGE-COMP02*[2].

[2]In this example, one 2D face corresponds to one image component. If several 2D faces have C^1 continuity across the edges, these faces are grouped and stored as one single image component. In this case, face area and face moment are calculated over the group of faces.

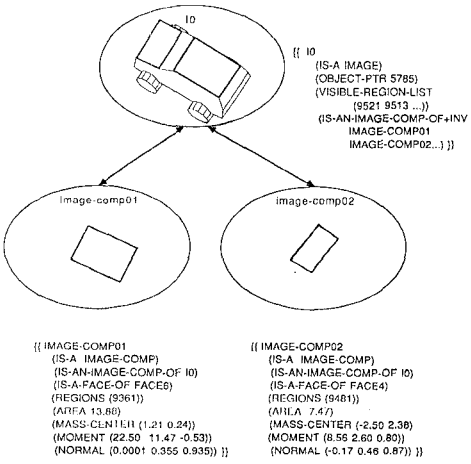


Figure 5: Object appearance represented in frames.

Each frame corresponding to one visible 2D face maintains various geometric properties of the face in slots. For example, face area and face moment are maintained in slots *AREA* and *MOMENT*. The values of these features are obtained by using output data from a geometric modeler. Each frame representing a 2D visible face has a backpointer to the 3D face from which the 2D face is projected. For example, the *IS-A-FACE-OF* slot of *IMAGE-COMP01* frame has a value *FACE6*.[3] An image structure consists of an image frame and image component frames.

Aspect

By using object appearances by the detection constraints, we can generate aspects defined by a particular sensor. Aspects are originally defined as topologically equivalent classes with respect to the object features [24]. Since each sensor has particular features to be detected, we have to modify the definition of aspects according to features detected by a sensor. We will consider aspects given from appearances by a light-stripe range finder.

We can define aspects for a light-stripe range finder by those faces detectable in each aspect. Suppose we have n faces, $S_1,S_2,...,S_n$. Let the variable X_i denotes the condition whether or not the face S_i is detected, that is

$$X_i=\begin{cases}1 & \text{face } S_i \text{ is detectable;}\\ 0 & \text{otherwise.}\end{cases}$$

An n-tuple $(X_1,X_2,...,X_n)$ represents a label of an object appearance in terms of face detectability. This label will be referred to as a *shape label*, and we can characterize each object attitude with this label. The set of contiguous object attitudes that have the same *shape label* forms an *aspect* in this example [6].

Appropriate descriptions of aspects must be defined so that they can be used in automatic generation of object recognition programs. The description of an aspect should include constituent appearances, a set of features extractable for the aspect, and the expected feature values. This description should have flexible and convenient forms for applying generation rules to them and for use in execution. Since an aspect is an abstract concept for a group of images (appearances), an aspect structure is similar to its constituent image structures.

[3]Each frame also contains array addresses of various geometric items such as 2D FACE, 2D EDGE and 2D VERTEX in the data base of the geometric modeler; for example, *9361* in *REGIONS* slot of *IMAGE-COMP01* frame. These allow us to access the original geometric data, if necessary.

In order to construct aspect structures, shape labels of all image structures are examined one by one, where a shape label is the combination of visible 3D faces. The visible 3D faces among a 2D appearance can be retrieved by backpointers of 2D faces to 3D faces such as *FACE6* in *IS-A-FACE-OF* slot of *IMAGE-COMP01* frame in Figure 5, where *FACE6* is the frame name of a 3D face of the object.

If an image structure cannot find any aspect structure with the same shape label among the already established ones, a new aspect frame is created together with aspect component frames which correspond to image component frames: therefore, the aspect structure has the same structure as the image structure. Also, frames to represent the relationships between pairs of aspect components are created. If an image structure can find an aspect structure with the same shape label, the image frame is registered to the aspect frame as an instance and its frames of 2D faces are registered to corresponding aspect component frames.

An example of an aspect structure is shown in Figure 6. Aspect frame *ASPECT1* points to several aspect component frames, *ASPECT-COMP10*, *ASPECT-COMP11* with the *IS-AN-ASPECT-COMP-OF+INV* slot. It also points to its instance images *I0*, *I1* with *IS-AN-IMAGE-OF-ASPECT-OF+INV* slot, while its aspect component frame, *ASPECT-COMP10* points to its instance 2D faces *IMAGE-COMP01*, *IMAGE-COMP12*. Frame *ASPECT-COMP-RELATION-11-10* is a relation frame which represents the relationship between *ASPECT-COMP10* and *ASPECT-COMP11*.

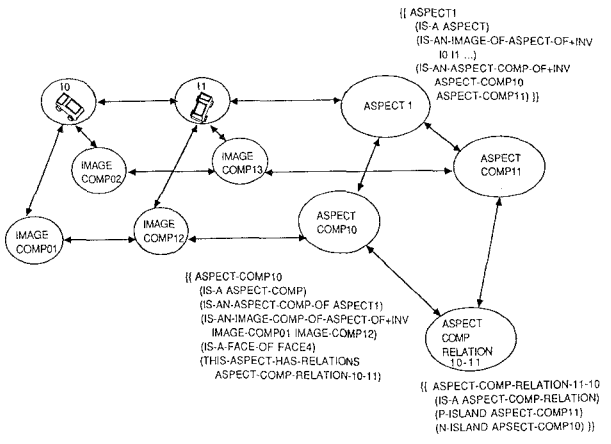


Figure 6: Frame representation of aspects: Each aspect structure consists of an aspect frame, aspect component frames, and aspect component relation frames. An aspect frame also points to its instance images *I0*, *I1*, ..., while its aspect component frame, *ASPECT-COMP10* points to its instance 2D faces *IMAGE-COMP01*, *IMAGE-COMP12*.

RELIABILITY OF SENSORS

Once a sensor feature is detected, the next question to ask is how reliable the sensor feature is or to determine uncertainty in the sensor feature. This section discusses two issues of uncertainty in feature values. The first issue is uncertainty in sensory measurement; any sensory measurement detected by a sensor always contains measurement uncertainty. Determining the bound of the uncertainty is important for model based vision. Suppose there is a sensor feature for which the geometric model takes two nominal values, 100 and 90, for two distinct situations. If a sensor has an uncertainty range of plus/minus 1 for the sensor feature, we can use the feature from that sensor as one of reliable discriminators in the recognition stage. On the other hand, if a sensor has an uncertainty range of plus/minus 20, we cannot use the feature from that sensor.

The second issue is propagation of uncertainty from sensor features to geometric features, hence the resulting uncertainty of those geometric features. In some cases, a detected sensor feature from a sensor is used directly as a feature; in most cases, however, geometric features are derived from sensor features. Thus, it is necessary to determine the uncertainty propagation mechanism for determining resulting uncertainty in geometric features.

Uncertainty in Sensory Measurement

Uncertainty in sensor measurement is caused by various reasons which include; variance in brightness values, variance in light source direction, and various digitization mechanisms. However, the major uncertainty of an intensity-based sensor comes from brightness variance, and the major uncertainty of a position-based sensor comes from variance in light source as showin in Table 2.

Table 2 Main factor of uncertainty

Sensor	Type	Factor
Edge Detector	Int	brightness variance
Shape-from-shading	Int	brightness variance
SAR	Int	flight direction
Time-of-Flight Range Finder	Pos	mirror direction
Light-stripe Range Finder	Pos	mirror direction
Binocular Stereo	Pos	camera directions
Trinocular Stereo	Pos	camera directions
Photometric Stereo	Int	brightness variance
Polarimetric light detector	Int	polarimetric variance

Since uncertainty in sensory measurement depends on the sensor, we will analyze the light-stripe range finder as a representative position-based sensor and photometric stereo as a representative intensity-based sensor.

Light-stripe range finder We will consider a depth measurement by a hypothetical light-stripe range finder. Let us assume that the main source of uncertainty in depth measurement comes from the ambiguity of the slit position on a surface due to the width of the light beam and angular errors in setting the light directions. The uncertainty model can be obtained analytically.

As shown in Figure 7 (a), let us denote the angular uncertainty of the light stripe by $\delta\theta$. The light is intercepted by an object surface, creating a slit pattern on it. The angular uncertainty $\delta\theta$ of the light direction results in uncertainty δy in the position on the surface:

$$\delta y = \frac{r\delta\theta}{\cos\alpha},$$

where r is the distance of the surface from the light source, and α is the angle between the light direction **S** and the surface normal **N**. This positional uncertainty on the surface is observed as the slit position uncertainty (or "slit width") δi in the camera image. If β is the angle between the surface normal **N** and the viewer direction **V**, then

$$\delta i = (\cos\beta)\delta y,$$

Finally, this uncertainty is transferred into the uncertainty in depth measurement by triangulation. For simplicity, if we assume orthographic projection for the camera, the uncertainty in the image δi creates uncertainty in depth δz,

$$\delta z = \frac{\delta i}{\tan\gamma},$$

where γ is the angle between **V** and **S**.

By representing the angles α, β, and γ in terms of **V**, **N**, and **S**, we

obtain

$$\delta z = \frac{\cos\beta}{\cos\alpha\tan\gamma} r\delta\theta = \frac{(\mathbf{N}\cdot\mathbf{V})(\mathbf{S}\cdot\mathbf{V})}{(\mathbf{N}\cdot\mathbf{S})\sqrt{1-\mathbf{S}\cdot\mathbf{V}}} r\delta\theta.$$

Since r is roughly constant, the uncertainty distribution of this light-stripe range finder over the detectable configurations is governed by the factor $\frac{(\mathbf{N}\cdot\mathbf{V})(\mathbf{S}\cdot\mathbf{V})}{(\mathbf{N}\cdot\mathbf{S})\sqrt{1-\mathbf{S}\cdot\mathbf{V}}}$.

Figure 7 (b) plots this function. The left sphere represent the feature configuration space. The detectable configurations are enclosed by two great circles. Since the light-stripe range finder is independent on the rotation of the z axis of the feature coordinate system, we only plot uncertainty for the configurations on the spherical surface. The detectable configurations are projected onto the tangential plane at the north pole. The right diagram represents the distribution of uncertainty over the plane. The larger the angle between the surface normal and the illuminator direction, the larger the uncertainty.

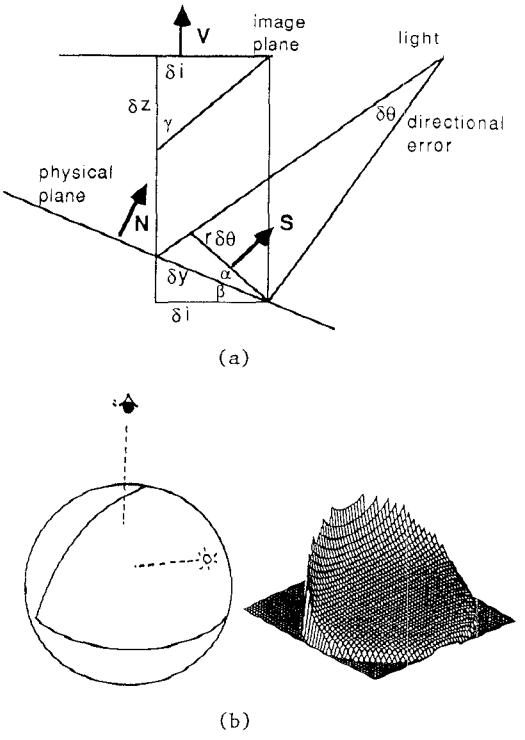


Figure 7: Predicted uncertainty in depth measurement by a light-stripe range finder: (a) Detection mechanism; (b) Predicted uncertainty in depth measurement.

Photometric stereo Let us consider the uncertainty in surface orientation by photometric stereo. Our photometric stereo can be described as a two step process. First, a brightness triple **I** is converted to a normalized brightness triple **E**.

$$\mathbf{E}=\mathbf{F}(\mathbf{I}).$$

Then, the normalized brightness triple is converted to a surface orientation **N**.

$$\mathbf{N}=\mathbf{A}\mathbf{E}.$$

Thus, we can obtain the uncertainty of surface orientation

$$U=(d\mathbf{N})^t d\mathbf{N}=(d\mathbf{I})^t\mathbf{J}^t\mathbf{A}^t\mathbf{A}\mathbf{J}d\mathbf{I},$$

where **J** is the jacobian matrix of **F**.

We now examine the jacobian matrix over the detectable configurations. Figure 8 (a) shows the distribution of $\frac{\partial f_1}{\partial i_1}$ over the detectable configurations, where the detectable configurations are plotted at the tangential plane at the north pole of the feature configuration space. Although it is possible to approximate the distribution with a polynomial, we assume it is constant 0.004 over the detectable configurations for the sake of simplicity. We follow the same method for the other component of the jacobian matrix.

$$J=\begin{bmatrix}0.004 & 0.002 & 0.002\\ 0.002 & 0.004 & 0.002\\ 0.002 & 0.002 & 0.004\end{bmatrix}$$

We determine $\mathbf{A}d\mathbf{E}$ from the real data because **A** is represented as a lookup table. From our experiment, our TV camera (8 bit) has a standard deviation 3. Thus, taking two sigma di=6. Suppose only one light source causes error at one time. Then, normalized brightness uncertainty, $\sqrt{d\mathbf{E}^t d\mathbf{E}}$=0.03. Our lookup table has 16 cells in one normalized brightness axis and the distance between the maximum normalized brightness and the minimum normalized brightness is roughly 0.8 over the detectable configurations. 0.03 uncertainty in normalized brightness corresponds to 1.5 mesh uncertainty in the lookup table.

Next, we examine the relationship between one cell distance and the surface orientation difference. Figure 8 (b) shows the angular distance between two adjacent cells in the lookup table. By using this result and a 1.5 mesh uncertainty in the normalized brightness, the total uncertainty in surface orientation becomes 5 degrees over the detectable configurations. In Figure 5c, the horizontal broken lines indicate plus/minus 5 degrees uncertainty, while vertical thick lines indicate uncertainty directly measured from our system. Both results agree over the detectable configurations.

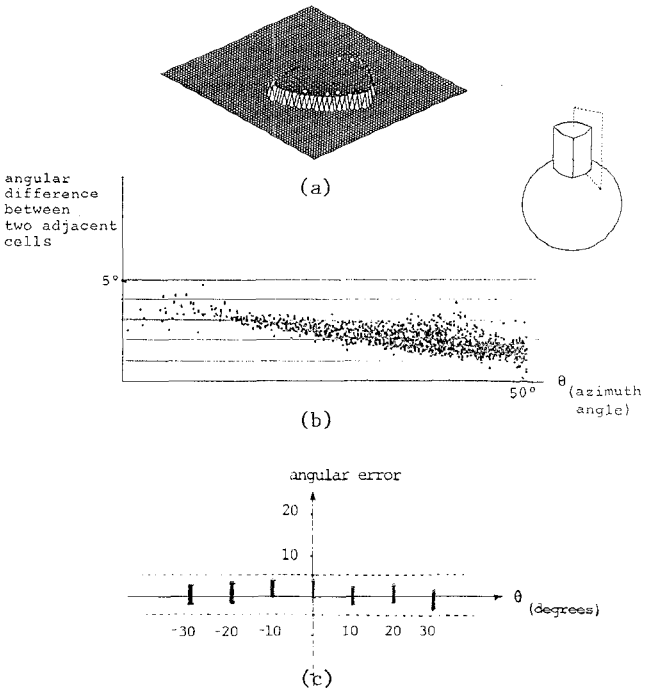


Figure 8: Uncertainty in surface orientation by photometric stereo: (a) Distribution of $\frac{\partial f_1}{\partial i_1}$ over the detectable configurations; (b) Angular difference between two adjacent cells in the lookup table; (c) Uncertainty in surface orientation over the detectable configurations.

Uncertainty in Geometric Features

Usually sensory measurements such as depth detected by a range finder, are further converted into object features such as area and inertia of a face. This process involves grouping pixels into regions, extracting some feature values and transforming them into object features. Modeling uncertainty propagation in this process is difficult in general. However, as an example of predicting the uncertainty range of a geometric feature, let us consider an area feature of a face detected as a region by our hypothetical light-stripe range finder. Figure 9(a) shows the conversion process from depth values to the area of a face. The process includes three parts: grouping pixels into a region to obtain the area in the image, computing the surface orientation of the region, and finally converting the image area into the surface area by the affine transform determined by the surface orientation. We will analyze how the uncertainty is introduced and propagated in these three parts.

Suppose a surface under consideration has the real area A and the surface orientation β (angle between the surface normal and the viewing direction). It should create a region of size n pixels where

$$n = A\cos\beta.$$

However, because of the imperfect detectability of the sensor, the sensor fails to find some of them so that the measured area is different from the nominal area n. Let P_d denote the detectability for this surface [23]. Then, the process of measuring the area by sampling n pixels can be modeled by a binomial distribution with mean nP_d and variance $nP_d(1-P_d)$. Assuming two standard deviations, the discrepancy in area measurement will be

$$\delta n = n-(nP_d-2\sqrt{nP_d(1-P_d)}) = n(1-P_d)+2\sqrt{nP_d(1-P_d)}.$$

Another uncertainty is introduced in obtaining the surface orientation β from measured depths due to uncertainty in depth measurement δz. If we estimate the surface orientation at a pixel by differentiating depths of neighboring pixels, then the uncertainty in surface orientation will be $\cos^2\beta\delta z$. However, since we have roughly n pixels in the region, the surface orientation will be averaged, reducing the uncertainty by a factor $\sqrt{n}$. Thus

$$\delta\beta = \frac{\cos^2\beta}{\sqrt{n}}\delta z$$

Finally, the estimation of area of the face, $A+\delta A$, is obtained by converting the image area into 3D surface area.

$$A+\delta A = \frac{n+\delta n}{\cos(\beta+\delta\beta)}$$

Thus, assuming that $\delta\beta$ is small, we see that

$$\delta A = A(1-P_d)+2\sqrt{\frac{AP_d(1-P_d)}{\cos\beta}}+A\tan\beta\delta\beta$$

$$= A(1-P_d)+\sqrt{\frac{A}{\cos\beta}}(2\sqrt{P_d(1-P_d)}+\frac{\sin 2\beta}{2}\delta z)$$

By this method, we can predict what deviations from the nominal value of the area feature should be expected, once we model the sensor and know its intrinsic detectability P_d and uncertainty in depth δz.

In order to examine the validity of the propagation model, we execute an experiment using our photometric stereo. We first observed a face five times by our photometric stereo, where the face was inclined 30° from the direction of TV camera's axis. From the measured surface orientation and projected pixel number, we recovered the face area and face inertia by the affine transform and obtained the results shown in the column observed in Table 5.

Our photometric stereo has plus/minus 5°,(or 0.0872 radian) uncertainty in surface orientation and observed $n=506$ pixels from the face. This gives $\delta\beta=0.00387$. Since we measure the face in the ideal condition, $P_d=1$. We insert these parameters in the above equation and obtain the uncertainty in area. In order to predict the uncertainty range in inertia, we double the uncertainty range in area. These values are shown in the column predicted in Table 3.

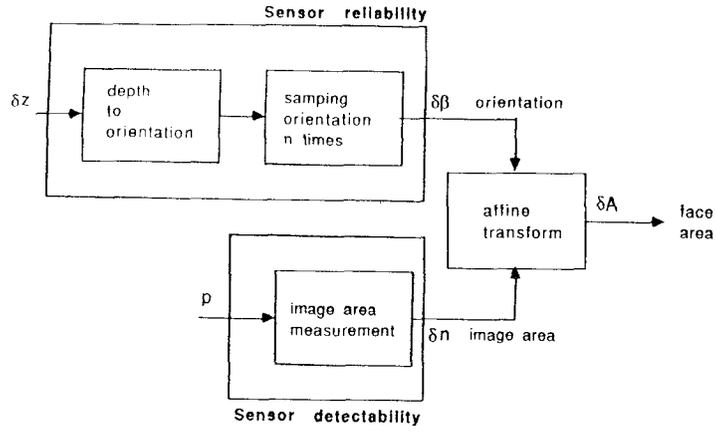


Figure 9: Conversion process from depth value values to the area of a face.

The result gives the consistency between the predicted uncertainty and the real uncertainty.

Table 3 Reliability of Geometric Features

Feature	Observed	Predicted
Area	0.0023	0.0022
Inertia	0.0044	0.0043

Use of Uncertainty for Tree Generation

By using sensor model, we can predict the ranges of various feature values at each aspect. At each image, since a nominal value of a feature and its configuration with respect to sensor coordinates are given, we can predict the range of the feature value for each 2D face of the image by using the formula described above. Then, the range of the feature value at an aspect component is obtained as a sum of ranges of the feature values over its registered image components which can be reachable along *IS-AN-IMAGE-COMP-OF-ASPECT-OF+INV*. The predicted range will be stored in the slot of an aspect component frame.

Firgure 10 shows slots for this purpose. For example, area ranges, moment ranges, and moment ratio ranges are calculated at each image components, *IMAGE-COMP01*, *IMAGE-COMP12* which can be retrieved along the link stored in slot *IS-AN-IMAGE-COMP-OF-ASPECT-OF+INV* of *ASPECT-COMP10* frame in figure 6. The sum of the ranges are stored in slot *AREA-VARIANCE*, *MOMENT-VARIANCE*, and *MOMENT-RATIO-VARIANCE* of *ASPECT-COMP10* frame. Similarly, feature ranges of aspect component relations, such as *DISTANCE-VARIANCE*, *MOMENT-ANGLE-P-TO-N-VARIANCE*, *SURFACE-ORIENTATION-ANGLE-VARIANCE*, are obtained and stored. These ranges of features will be retrieved by generation rules at compile time to generate an interpretation tree and by the execution process at run time in recognizing a scene.

AUTOMATIC GENERATION OF OBJECT RECOGNITION PROGRAM

Figure 11 presents an example of how to apply the sensor model to the automatic generation of an object recognition program. A geometric model as shown in Figure 11(b) is obtained from a toy wagon in Figure 11(a). From the geometric model and sensor

```
{{ ASPECT-COMP10
  ....
    (AREA-VARIANCE (13.94 14.85 15.75))
    (MOMENT-VARIANCE (22.77 25.06 27.34))
    (MOMENT-RATIO-VARIANCE (0.53 0.65 0.76))
    (VISIBLE-EDGE-LIST ASPECT-COMP10-VISIBLE-EDGE-LIST)
  ....
   }}

{{ ASPECT-COMP-RELATION-11-10
  ....
    (DISTANCE-VARIANCE (5.04 5.38 5.69))
    (MOMENT-ANGLE-P-TO-N-VARIANCE (1.29 1.53 1.8))
    (MOMENT-ANGLE-N-TO-P-VARIANCE NIL)
    (SURFACE-ORIENTATION-ANGLE-VARIANCE (0.04 0.21 0.40))
  ....
   }}
```

Figure 10: Slots for storing uncertainty ranges of features

detectability, we can generate various possible appearances as shown in Figure 11(c). We can classify and categorize various appearances into possible aspects, where each aspect shares the common detectable faces. One aspect structure is constructed at each aspect group (Figure 11(d)). Predicted ranges of uncertainty of geometric features are determined using the sensor reliability. Generation rules generate the recognition strategy automatically based on the predicted ranges of uncertainty (Figure 11(e)) [28, 29]. Once the recognition strategy is obtained, the strategy is converted into an executable program (Figure 11(f)) [30].

The generated program is applied to the scene as shown in Figure 12. Figure 12 (b) shows the needle map given by our photometric stereo. Figure 12(c) shows segmented regions using shadows and surface orientation discontinuities. The arrow indicates the highest region, which is given to the program for recognition, while Figure 12(d) shows edge distributions superimposed on the region map. The black nodes in Figure 12 (e) indicates the follow of control in the real run. The program classifies the region to the corresponding aspect as shown in Figure 12(e). Then, the generated program determines the precise attitude of the object as shown in Figure 12(f), where obtained position and attitude of the object is superimposed over the scene.

CONCLUDING REMARKS

In this paper we discussed how to apply sensor modeling towards automatic generation of object recognition programs. Our sensor model consists of two characteristics: sensor detectability and sensor reliability. Sensor detectability specifies under what conditions a sensor can detect a feature, while sensor reliability is a measure of confidence for the detected features over the detectable configurations.

We have defined the configuration space which represents the relationship between sensor coordinates and object coordinates. The sensor detectability and the sensor reliability are expressed in this configuration space. Constraints in the configuration space involved in detecting features have been developed by using G-source illuminated configurations and G-source illumination directions. We have shown how to compute the sensor detectability distribution and the sensor reliability distribution for two representative sensors: a light-stripe range finder and photometric stereo.

In model based vision, expected values of various features can be computed from 3D geometric models. These values are, however, nominal values that should be taken in ideal cases or sensed by ideal sensors. On the other hand, actual observed sensor data contains noise and should be used accordingly. The sensor model bridges the discrepancy between these two values by modeling the distribution of the sensed value based upon the characteristics of a given sensor.

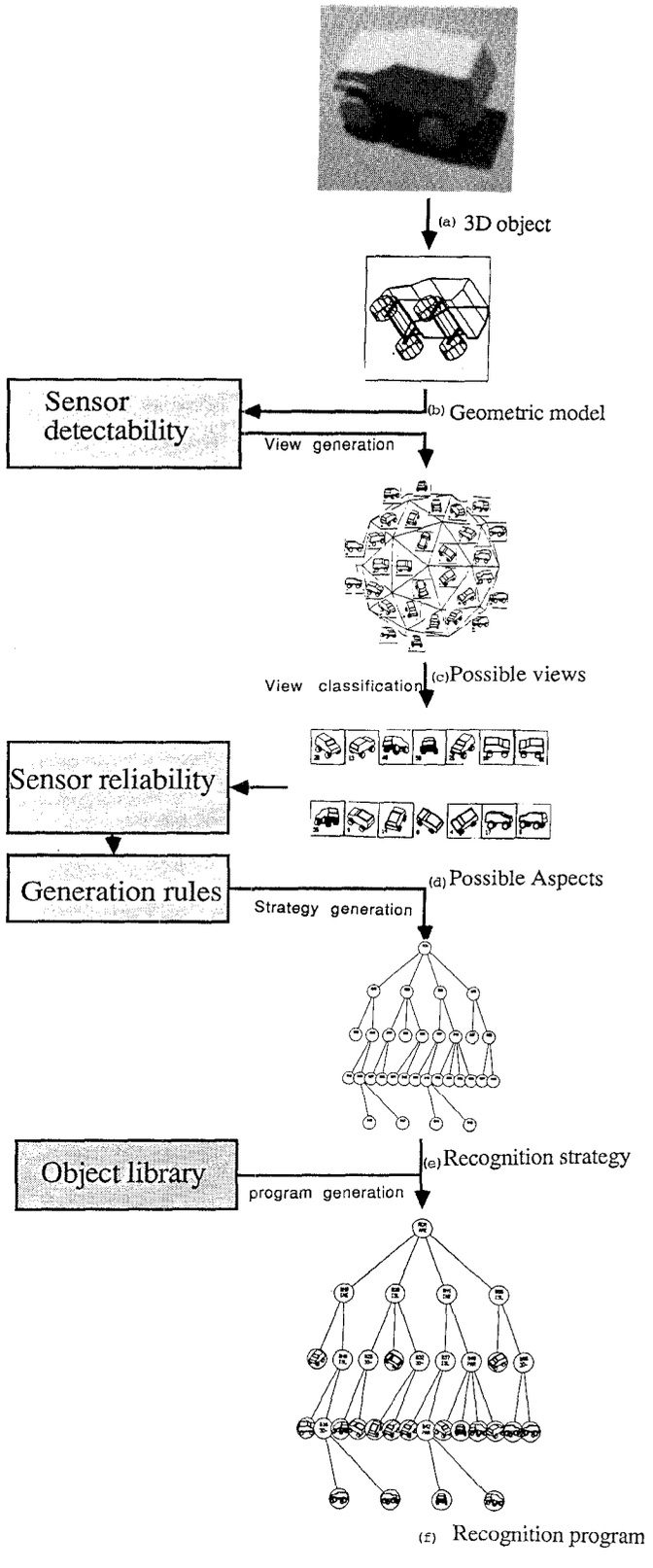
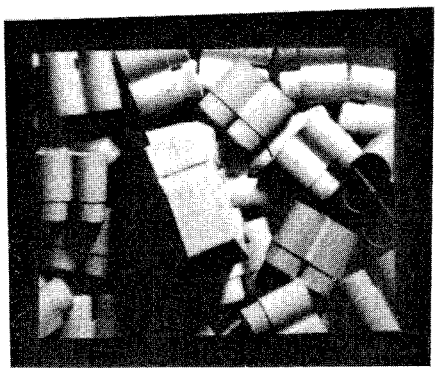
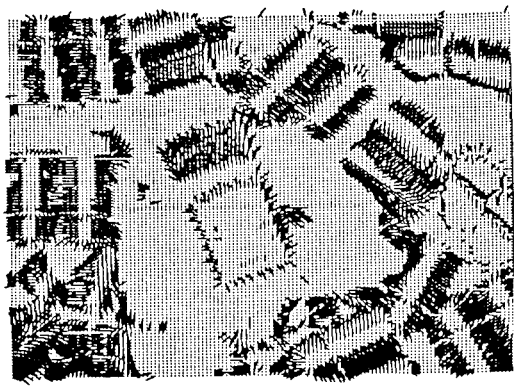


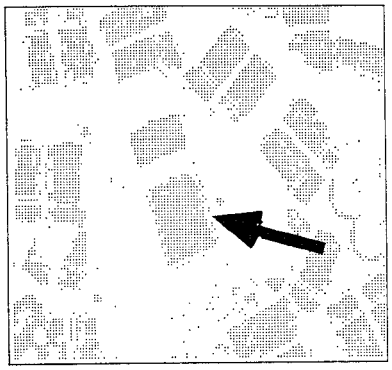
Figure 11: Sensor model and automatic generation of object recognition program.



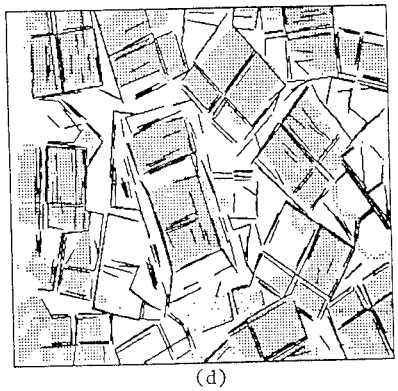
(a)



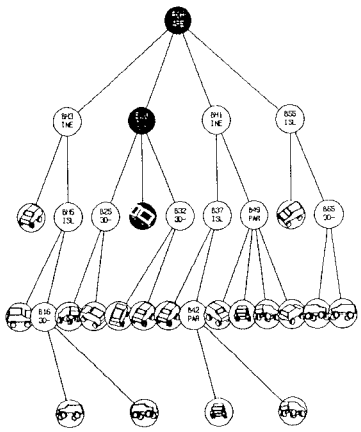
(b)



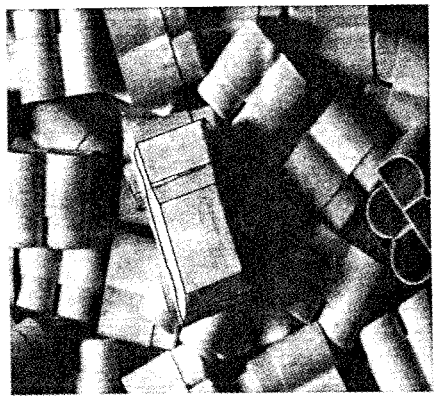
(c)



(d)



(e)



(f)

Figure 12: Tree execution: (a) Scene; (b) Surface orientation distribution of the scene; (c) Segmented regions using shadows and surface orientation discontinuities. The arrow indicates the target region selected by the system. (d) Edge distributions superimposed on the region map; (e) Execution result. Black nodes indicates the trace of the control. The target region is classifed into the corresponding aspect. (f) Obtained position and attitude of the object superimposed over the scene.

We have also analyzed the uncertainty propagation mechanism from sensory measurements to geometric features. This is important because quite often we are interested in geometric features derived from detected measurement. Once we establish the method to model the uncertainty propagation, we can also assess the uncertainty in the geometric features, hence we can construct a recognition system more systematically and reliably. Further study is required in this area.

Calculating detectable features of an object under the constraints of various sensors is a tedious job when we use a conventional geometric modeler. A better way is to interface a geometric modeler with the sensor model proposed. We call this a sensor modeler. The traditional geometric modeler only allows users to generate a 3D object by combining primitive objects and to display its views. In this sense, the traditional modeling system has only one sensor model, which is projection. The sensor modeler we propose can generate various 2D representations under given sensor specifications. Part of this facility is being implemented in our new geometric/sensor modeler VANTAGE [31].

ACKNOWLEDGEMENT

The authors thank Keith Gremban, Yoshinori Kuno, Ki Sang Hong, Shree Nayar, Purushotohaman Balakumar, Jean-Christophe Robert and the member of VASC (Vision and Autonomous System Center) of Carnegie Mellon University for their valuable comments and discussions.

I. Feature coordinate system

Face We define the z axis of the feature coordinate system to agree with the surface normal, and the x-y axes lie on the face, but are defined arbitrarily otherwise.

Edge We define the z axis to agree with the average direction of the two normals of the two adjacent faces incident to the edge. We define the x axis of the feature coordinate system to agree with the edge direction. The y axis is determined accordingly.

Vertex We define the z axis to agree with the average direction of the normals of the adjacent faces incident to the vertex. The x-y axes lie on the plane perpendicular to the z axis, but are defined arbitrarily otherwise.

References

[1] Binford, T.O., "Survey of model-based image analysis systems", *The International Journal of Robotics Research*, Vol. 1, No. 1, 1981, pp. 18-64.

[2] Chin, R.T. and Dyer, C.R., "Model-based recognition in robot vision", *ACM Computing Surveys*, Vol. 18, No. 1, March 1986, pp. 67-108.

[3] Goad, C., "Special purpose automatic programming for 3D model-based vision", *Proc. of DARPA Image Understanding Workshop*, DARPA, 1983, pp. 94-104.

[4] Bolles, R.C. and Horaud, P., "3DPO: A three-dimensional part orientation system", in *Three-Dimensional Machine Vision*, Kanade, T., ed., Kluwer, Boston MA, 1987, pp. 399-450.

[5] Koezuka, T. and Kanade, T., "A technique of pre-comiling relationship between lines for 3D object recognition", *Proc. Intern. Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, IEEE Industrial Electronics Society, February 1987, pp. 144-149.

[6] Ikeuchi, K., "Generating an Interpretation Tree from a CAD Model for 3-D Object Recognition in Bin-Picking Tasks", *International Journal of Computer Vision*, Vol. 1, No. 2, 1987, pp. 145-165.

[7] Roberts, L.G., "Machine perception of three-dimensional solids", in *Optical and Electro-Optical Information Processing*, Tipplett, J.T., ed., MIT Press, Cambridge, MA, 1965, pp. 159-197.

[8] Marr, D. and Hildreth, E., "Theory of edge detection", *Proc. of the Royal Society of London B*, Vol. 207, 1980, pp. 187-217.

[9] Canny, J.F., "Finding edges and lines in images", Tech. report AI-TR-720, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1983.

[10] Horn, B.K.P., "Obaining Shape from Shading", in *The Psychology of Computer Vision*, Winston, P.H., ed., McGraw-Hill, New York, 1975, pp. 115-155.

[11] Ikeuchi, K. and Horn, B.K.P., "Numerical shape from shading and occluding boundaries", in *Computer Vision*, Brady, M.J., ed., North-Holland, Amsterdam, 1981, pp. 141-184.

[12] Tomiyasu, K., "Tutorial review of Synthetic-Aperture Rader(SAR) with applications to imaging of the ocean surface", *Proc. of the IEEE*, Vol. 66, No. 5, May 1978, pp. 563-583.

[13] Kuno, Y. Ikeuchi, K. and Kanade, T., "Model-based Object Recognition in SAR Images using Configuration Spaces", Tech. report, Carnegie Mellon University, Computer Science Department, September 1988.

[14] Jarvis, R.A., "A laser time-of-flight range scanner for robotic vision", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 5, 1983, pp. 505-512.

[15] Hebert, M. and Kanade, T., "Outdoor scene analysis using range data", *Proc. of Intern. Conf. on Robotics and Automation*, IEEE Computer Society, San Francisco, April 1986, pp. 1426-1432.

[16] Agin, G.J. and Binford, T.O., "Computer description of curved objects", *International Joint Conf. on Artificial Intelligence*, Stanford, CA, August 1973, pp. 629-640.

[17] Marr, D. and Poggio. T., "A computational theory of human stereo vision", *Proc. of the Royal Society of London B*, Vol. 204, 1979, pp. 301-328.

[18] Ohta, Y. and Kanade, T., "Stereo by intra- and inter-scanline search using dynamic programming", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, 1985, pp. 139-154.

[19] Milenkovic, V.J. and Kanade, T., "Trinocular vision: using photometric and edge orientation constraints", *Proc. of DARPA Image Understanding Workshop*, DARPA, Miami Beach, FL, December 1985, pp. 163-175.

[20] Woodham, R.J., "Reflectance Map Techniques for Analyzing Surface Defects in Metal Castings", Tech. report AI-TR-457, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1978.

[21] Ikeuchi, K. Nishihara, H.K. Horn, B.K.P. Sobalvarro, P. and Nagata, S., "Determining grasp points using photometric stereo and the PRISM binocular stereo system", *The International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 46-65.

[22] Koshikawa, K., "A polarimetric approach to shape understanding of glossy objects", *Proc. of 6th Intern. Joint Conf. on Artificial Intelligence*, 1979, pp. 493-495.

[23] Ikeuchi, K. and Kanade, T., "Towards automatic generation of object recognition program", *Proc. of IEEE*, No. 11, November 1988, (accepted for publication, a slightly longer version is avaiable as CMU-CS-88-138)

[24] Koenderink, J. J. and Van Doorn, A. J., "Geometry of binocular vision and a model for stereopsis", *Biological Cybernetics*, Vol. 21, No. 1, 1976, pp. 29-35.

[25] Bajcsy, R.,Krotkov, E., and Mintz, M., "Models of errors and mistakes in machine perception", *Proc. of Image Understanding Workshop*, DARPA, 1987, pp. 194-204.

[26] Faugeras, O.D., Ayache, N., Faverjon, B. and Lustmán, F., "Building visual maps by combining noisy stereo measurement", *Proc. of Intern. Conf. on Robotics and Automation*, IEEE computer society, San Fransisco, April 1986, pp. 1433-1438.

[27] Matthies, L. and Shafer, S.A., "Error modelling in stereo naviagtion", Tech. report CMU-CS-86-140, Carnegie-Mellon University, Computer Science Department, 1986.

[28] K. Ikeuchi, "Toward Automatic Generation of Object Recognition Program -- Aspect Classification and Object Library", Tech. report, Carnegie Mellon University, Computer Science Department, 1988, (in preparation)

[29] K. Ikeuchi and K. S. Hong, "Toward Automatic Generation of Object Recognition Program -- Linear Shape Change Determination", Tech. report, Carnegie Mellon University, Computer Science Department, 1988, (in preparation)

[30] Chang, H. Ikeuchi, K. and Kanade, T., "Model-based vision system by object-oriented programming", Tech. report CMU-RI-TR-88-3, Carnegie Mellon University, The Robotics Institute, March 1988.

[31] Kanade, T., Balakumar, P., Robert, J.C., Hoffman, R., and Ikeuchi, K., "Overview of geometric/sensor modeler VANTAGE", *Proc. The International Symposuim of Exposition on Robots*, The Australian Robot Association, Sydney, Australia, November 1988.