Relative Depth for Behavior Based Recognition

Ehud Rivlin and Liuqing Huang
Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742

Abstract

The problem of object recognition from sensory data is traditionally a problem of finding structure from an array of intensity functions. To solve the problem, a depth map is recovered to match against a set of possible structures to suggest the most likely objects. However, few schemes of this nature have been able to perform beyond laboratory assumptions.

We propose to study object recognition by asking the question in the context of an agent performing the recognition in an environment where the agent is performing a behavior. In our paradigm, the problem becomes a problem of action from intensity functions. In accomplishing a behavior, we are determining our next step of action from the images. Acquiring the information for action is a solution for a recognition task. The recognition task is agent and behavior dependent and can use the output of different visual modules. One possible visual module for the object recognition process can be a module that gives qualitative depth information. We discuss the way such a module can operate.

We conclude that many recognition tasks become easy under this paradigm, as recognition is reduced to qualitative judgements of the scene. We proof our point of view with a set of real world examples based on visual information from relative depth visual module.

1 Behavior based recognition

The problem of object recognition from sensory data is defined in the literature as the association of visual input with a name or a symbol. Although very good research on the topic has been published, we still lack vision systems that can recognize in real time a large number of objects (natural or man-made). This is because in order to recognize an object one would have to visually recover it (i.e. its shape and various properties) and then match this recovered information against a database of known objects [2]. However, full recovery is hard to achieve to date, and matching suffers from combinatorial explosion.

Model-based recognition on the other hand has been suggested as a remedy to these problems. By requiring that the objects to be recognized are specific instances of a generic model (e.g., polyhedral, generalized cylinders, cones, superquadrics, etc.), the problems of surface recovery and matching become easier. However, model-based approaches obviously employ strong assumptions about the nature of the scene and thus they lack generality.

We proposed [3] to study the problem of object recognition by asking a different question, i.e. by considering it in the context of an agent performing it in an environment, where the agent's intentions translate into a set of behaviors. What are objects for? An object can suit a purpose, fulfill a function. If the agent recognizes this, it has in effect recognized the object. To perform this type of recognition we need on one hand a definition of the desired function, and on the other the means of determining whether the object can fulfill that function. To find out if an object can fulfill a function we need to perform various *partial* recovery tasks.

An agent is defined as a set of intentions, $I_1, I_2, \ldots, I_n$. Each intention I_k is translated into a set of behaviors, $B_{k1}, B_{k2}, \ldots, B_{km}$. Each behavior B_{ki} calls for the completion of recognition tasks $T_{ki1}, T_{ki2}, \ldots, T_{kij}$. The agent acts in behavior B_{ki} under intention I_k. The behavior calls for the completion of recognition tasks $T_{ki1}, \ldots, T_{kin}$. The behavior sets parameters for the recognition tasks. Note that the same object can answer positively to several recognition tasks. Under one behavior a chair will answer yes to some recognition task that is asking for obstacles, under another behavior it will answer yes to a recognition task that is asking for a sitting place, and under another it will answer yes to a task that is ask-

0-7803-0720-8/92 $3.00

ing for an assault weapon.

We view the recognition process along the axis intention, behavior, recognition task. For a theory of purposive object recognition we should be able to make two basic transformations: first from the desired intention to the set of behaviors that achieve it, second from a specific behavior to some needed recognition task(s). In [3] we showed that the intention-to-behaviors problem with a finite number of behaviors is undecidable. We believe that a general automatic transformation from behaviors to recognition tasks is also hard. A possible solution is to use compiled knowledge and build a set of useful translations.

2 Recognition tasks under a specific behavior

In order for us to build working systems, a natural direction will be to use a set of translations. It seems that a certain class of robots will share some common, useful, functional translations. We can define categories like animate, inanimate, prey, predator, obstacle, etc. that will belong to some hierarchical structure. The hierarchies are functional and have perceptual substance. They must have perceptual characteristics that make them discriminable. These functional relationships (here functional is been used in the utilitarian sense), can be translated, for example, into surface characteristics and geometric properties in a crude qualitative way.

Each recognition task activates a different collection of basic perceptual modules. Each module finds a generic object property which is a result of one or a combination of direct low-level computations on some sensory data (possibly done by other modules). The result of a module's operation is given as a qualitative value. Each module has its own neighboring open intervals which are parameter-specific. Such modules, for example, might provide information about the size of the object under consideration (very small, small, medium, large, very large) relative to the observer , various shape features of the object under consideration, etc. More complex modules might answer, for example, questions as is the object graspable? (based on low-level modules of size, shape, etc.), is the object a possible container? etc.

A recognition task for a "cup concept" might be activated and performed in the following manner. A drinking intention could activate a searching behavior for "something that it is possible to drink from" [1]. This behavior calls for a recognition task with this definition as a source for the translation process to functional properties. This definition asks for a container that is open at one end, of reasonable size and graspable. These sub-tasks will be answered by the different visual modules.

As another example, in a defense scenario we might have the intention of throwing an object at an attacker. This calls for a behavior of searching for "something that it is possible to throw and create damage". This behavior will activate a recognition task for something that is rigid, mobile, graspable, and not too light or heavy. The following modules might be activated under this recognition task: Is it animate? mobile? graspable? hard? [2]

Note that the cup gives positive answers to all these questions. Under the defense intention the cup can be used as a missile. A cup and a stone will give the same values, and are equally good for the current intention and behavior.

As a final example, a frog's feeding intention calls for a prey-catching behavior which will include a recognition task for a "bug concept". This task could make use of the following modules: Is it moving? Is the motion rapid (by the agent's scale)? small (by the agent's scale)? dark? reachable?

3 Integrating visual modules for recognition

Under our framework an agent acts in behavior B_{ki} under intention I_k. The behavior calls for the completion of recognition tasks $T_{ki1}, \ldots, T_{kin}$. The behavior sets parameters for the recognition tasks. Each recognition task activates a different collection of basic perceptual modules. Each module finds a generic object property which is a result of one or a combination of direct low-level computations on some sensory data (possibly done by other modules). The result of a module's operation is given as a qualitative value. Each module has its own neighboring open intervals which are parameter-specific. The i^{th} module can take one of $q_{i1}, \ldots, q_{in}$ qualitative values [3]

The state of our recognition system, denoted by Q_i, is a tuple of all the qualitative values of our modules $(q_1, \ldots, q_m)$ under recognition task T_{kij}. Each recognition task T_{kij} defines a system state that will

[1] See for a similar approach [4].

[2] This, as well as the rigidity requirement for graspability, requires tactile sensory information.

[3] Such qualitative values represent a partial recovery of the scene.

constitute a positive answer to that recognition task. Recognition is done when we complete our task, which means a stable answer from our modules. The required system state is a setting of parameters in a model. From that point of view our paradigm can be considered as model-based, though the model consist of qualitative values ranging from time, motion, color, shape to infra-red values. A common recognition task can be defined as a new module.

¿From this point of view we see the recognition system as a a collection of processes or modules which solve particular recognition tasks. These processes are connected dynamically to sensory modules, as well as to other kinds of input. We have recognition modules working in parallel, and we differentiate between two basic levels of modules: low and high. The low level modules get their inputs directly from the sensory data. The high level basic perceptual modules get their inputs either from the low level modules or from the raw sensory data (or both). In this sense the low and high level modules are just simple and complex modules respectively. The complex ones are given a task that requires a combination. These two types of modules work in parallel and are controlled in a mixed top-down/bottom-up manner.

In what follows we describe the implementation of one visual module and its use for purposive recognition. We will show how to achieve robustly relative depth from a stereo setup without correspondence and calibration, and how this visual module can be used under some intentions and behaviors.

4 A visual module - relative depth

Consider two cameras in a stereo configuration. Let $f^L(x,y)$ and $f^R(x,y)$ be the intensity functions of the left and the right images respectively. If point (x,y) in the left image corresponds to point $(x+\Delta x, y+\Delta y)$ (with $(\Delta x, \Delta y)$, the disparity vector) in the right, then the following equation holds:

$$f^L(x,y) = f^R(x+\Delta x, y+\Delta y) \qquad (1)$$

Expanding Equation (1) in a Taylor series and keeping the linear terms we have:

$$f^L(x,y) = f^R(x,y) + f_x^R \Delta x + f_y^R \Delta y$$

or

$$(f_x^R, f_y^R)(\Delta x, \Delta y) = f^L(x,y) - f^R(x,y) \qquad (2)$$

Equation (2) shows that we can determine the projection of the disparity vector $(\Delta x, \Delta y)$ on the direction of the gradient (f_x^R, f_y^R). This projection is what we call "normal disparity" and this is the input to our algorithm.

Assume that the camera is moving with a translational velocity of $(T_x, T_y, T_z)^T$ and a rotational velocity of $(R_x, R_y, R_z)^T$, the velocity of a world point $P = (X, Y, Z)$ is:

$$\mathbf{V} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + R \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The world point P is projected on the image plane as $p(x,y)$. The optic flow is the velocity of image point p can be found as:

$$v_x = \frac{T_x}{Z} - x\frac{T_z}{Z} - xyR_x + (1+x^2)R_y - yR_z$$

$$v_y = \frac{T_y}{Z} - y\frac{T_z}{Z} - (1+y^2)R_x + xyR_y + xR_z$$

Note that the unit normal vector (i.e. the direction of the image gradient) at $p(x,y)$ is (n_x, n_y). The normal disparity vector is the project of the optic flow on the unit normal vector. Thus the relationship between the motion velocity and the normal disparity is $v_n = v_x n_x + v_y n_y$, thus

$$\begin{aligned} v_n = {} & n_x\frac{T_x}{Z} + n_y\frac{T_y}{Z} - \frac{T_z}{Z}(n_x x + n_y y) \\ & -R_x y(x n_x + y n_y) \\ & +R_y x(x n_x + y n_y) \\ & +R_z(x n_y - y n_x) \\ & +R_y n_x - R_x n_y \end{aligned}$$

In the stereo setup, all motion parameters are zero except for T_x and R_y. Note that $T_x = b$ and $R_y = \alpha$, we have:

$$v_n = \frac{b n_x}{Z} + [(1+x^2)n_x + xy n_y]\alpha \qquad (3)$$

Note that the terms involving x^2 and xy are resulted from V_z, and it is a by-product of the rotation R_y around the Y-axis. However, the effect is very small. It is dominated by the translation along the X-axis and the rotation around the Y-axis. Human vision has a small instantaneous field of view. A large field of view is achieved by head and eye movements and registration. Indeed, from Equation (3), when the field of view is small ($< 30°$), $\| x^2 \| \ll 1$ and $\| xy \| \ll 1$. Terms involving x^2 and xy can be omitted in a qualitative decision. Thus

$$\frac{v_n}{n_x} = \frac{b}{Z} + \alpha \qquad (4)$$

Equation (4) describes the relationship between the depth Z, the vergence angle α, the baseline b and the normal disparity. As α and b is generally unknown, the depth Z can not be readily recovered. However, for any two given points $p1 = (x1, y1)$ and $p2 = (x2, y2)$ whose normal disparity can be found, we obtain two equations of the form of Equation (4). By the subtraction of the two equations we obtain:

$$(\frac{1}{Z_1} - \frac{1}{Z_2})b = \frac{v_{n1}}{n_{x1}} - \frac{v_{n2}}{n_{x2}} \quad (5)$$

¿From the sign of the right hand side of Equation (5), we can determine whether the 3D point of $p1$ is closer than that of $p2$.

5 Experiments - relative depth for behavior based recognition

We used relative depth as a major visual module in two basic behaviors. Under basic navigation behavior an agent is moving in a known environment. In our experiment we used a corridor. The agent should avoid obstacles while it is moving. While it is keeping the center of the corridor, it is checking for anomalies. The motion can continue as long as no unexpected anomalies are detected. When an anomaly was detected we use the relative depth module to estimate the position of the obstacle. The relative depth module segments the obstacle, and gives the actual recognition. Figure 1 shows a corridor scene where an obstacle (a chair) is present in the middle of the robot's path. The values of the computed normal disparity are presented in figure 2. The qualitative depth results are presented in figure 3. Threshold on their values isolate the chair from the environment (figure 4).

Another basic level behavior that uses relative depth is grasping. To reach a target object we need an estimate of the relative depth between the hand and the object. Under a grasping behavior we use relative depth to control the grasping process. The recognition here is of a situation: is the object before the hand? after it? Figure 5 shows the basic setup. Normal flow values are computed (see figure 6). These results are the input for the relative depth computation that enables the robot to evaluate its hand position (relative to the object).

6 Conclusions

We have presented an alternative approach to the problem of object recognition. Instead of the common two-stage, bottom-up process of complete scene recovery, followed by fitting to a model and matching in a database, we have formulated the problem as a top-down process. Recognition is studied in terms of the agent performing it, under its intentions and the behaviors triggered by them. It involves a verification process that checks for the existence of physical properties that provide needed functionality. We have used relative depth to solve a class of behavior based recognition problems. Experimental results were achieved for some basic behaviors. For recognition under navigation behavior, relative depth was used to recognize obstacles by isolating unexpected objects in close range. For grasping behavior relative depth was used to recognize the different stages in the process.

References

[1] Huang, L., Aloimonos, J. Y. (1991). "Relative depth from motion using normal flow: An active and purposive solution" *Proc. IEEE Workshop on Visual Motion.*

[2] Marr, D. (1982). "VISION: Computational Investigation into the Human Representation and Processing of Visual Information", Freeman, San Francisco.

[3] Rivlin, E., Aloimonos, J. Y. & Rosenfeld, A. (1991). "Purposive recognition: an active and qualitative approach " *SPIE Proc.*, Vol 1611, Boston, MA.

[4] Winston, P.H., Binford, T.O., Katz, B., & Lowry, M. "Learning physical description from functional descriptions, examples, and precedents," *Proc. of the AAAI*, pp. 433–439, 1983.

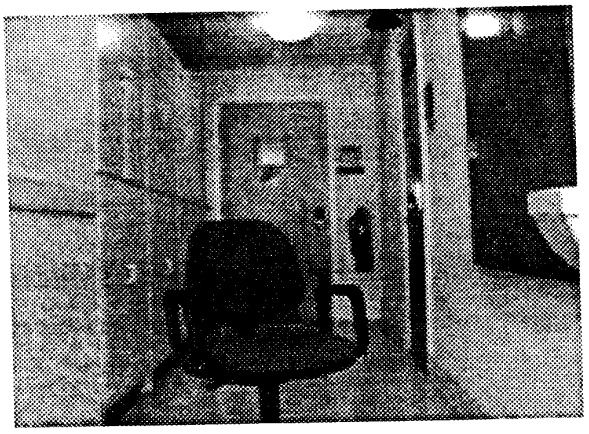

Figure 1: A chair is present in the corridor.

Figure 2: Normal disparity is computed from a stereo pair.

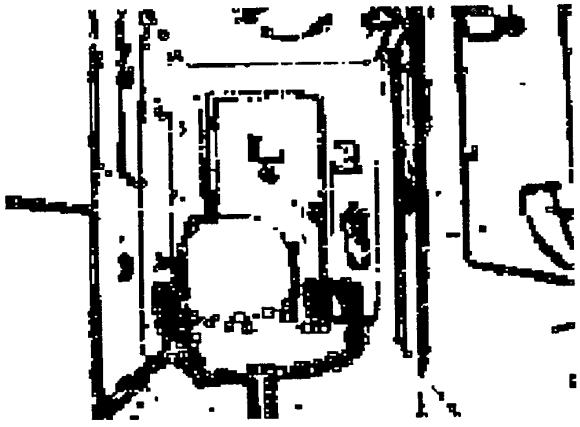

Figure 3: Qualitative depth is prepresented with size of the square.

Figure 4: The chair is isolated as an obstacle.

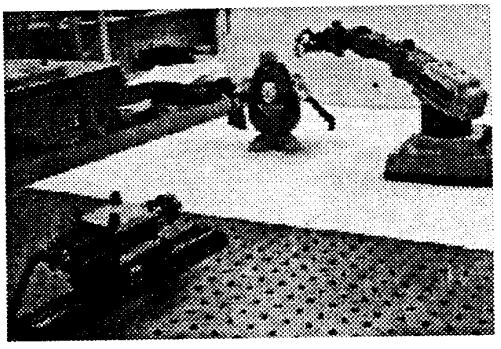

Figure 5: The setup of a robot arm reaching an object.

Figure 6: The result of normal flow is used to compute relative depth.

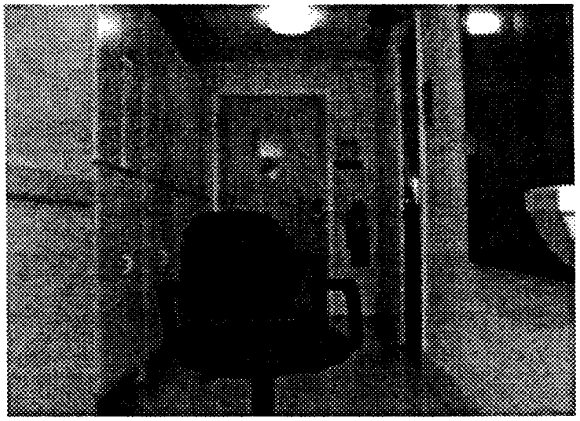

Figure 1: A chair is present in the corridor.

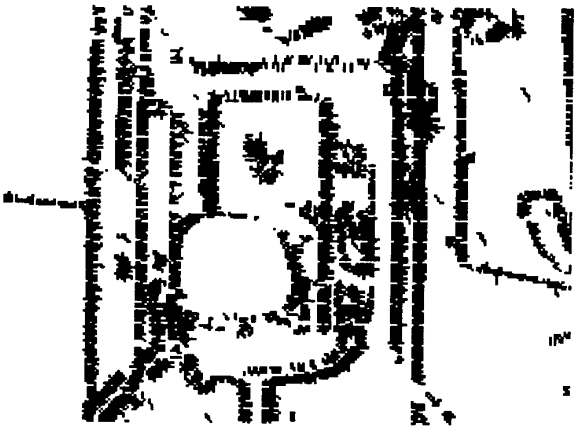

Figure 2: Normal disparity is computed from a stereo pair.

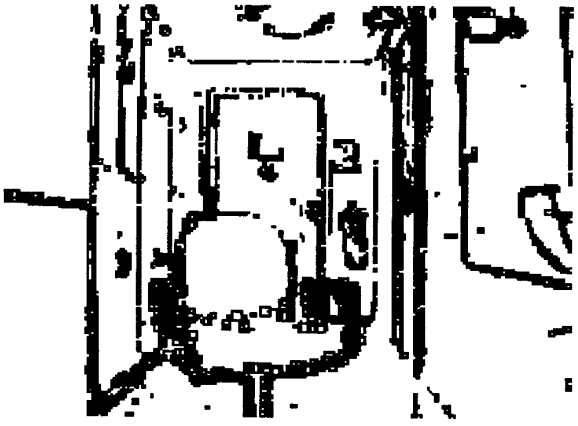

Figure 3: Qualitative depth is prepresented with size of the square.

Figure 4: The chair is isolated as an obstacle.

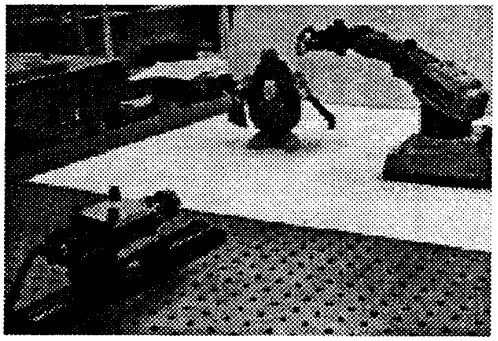

Figure 5: The setup of a robot arm reaching an object.

Figure 6: The result of normal flow is used to compute relative depth.