REPRESENTING WORKSPACE AND MODEL KNOWLEDGE FOR A ROBOT WITH MOBILE SENSORS

M. Shneier, E. Kent and P. Mansbach

Reprinted from IEEE SEVENTH INTERNATIONAL
CONFERENCE ON PATTERN RECOGNITION Proceedings
July 30 - August 2, 1984 Montreal, Canada

IEEE COMPUTER SOCIETY
1109 Spring Street, Suite 300
Silver Spring, MD 20910

REPRESENTING WORKSPACE AND MODEL KNOWLEDGE FOR A ROBOT WITH MOBILE SENSORS

M. Shneier, E. Kent, and P. Mansbach

National Bureau of Standards
Washington, D.C. 20234.

Abstract

A representation is described for supplying a robot manipulator with information about its workspace. Information is obtained from sensors that move with the manipulator. Spatial information is stored in an octree, allowing fast computation of which parts of the workspace are occupied and which are navigable. Information about properties and features of objects is stored in a set of tables or attribute lists. This information is used to match objects in the world with stored models and to assign names to instances of objects and features. Recognized objects are stored in the same way as unrecognized objects, so that all operations on the workspace model are uniform. The two representations are linked to enable objects to be located in space by name, by description, or by position, and to facilitate finding out what object occupies a particular volume in the workspace.

1. Introduction

A representation that models a robot's workspace and the objects in it should represent all information known about the locations of objects, the volumes they occupy, and the uncertainties in their positions and sizes. It should be able to integrate information obtained by sensing the world with that obtained from hypotheses and expectations. It should enable uncertainties in positions to be reduced, and object identities to be established with increasing certainty as more sensor data is acquired. It should also allow answers to questions about free space as well as space that is occupied.

The representation is designed to operate in an environment in which the robot and sensors move about. Information about the world is obtained from sensors whose locations are known. This simplifies some of the problems of 3D reconstruction from 2D projections. The representation comprises a spatial representation, describing what parts of the workspace are occupied, empty, or about which no information is known, and a representation of the objects and features in the space. Constraints explicit in one of the representations simplify descriptions in the other, and make answering questions easier using either representation.

The representation has the properties of uniformly representing expected and unexpected information, and enabling rapid updating of information as more data become available. It can be viewed at several levels of resolution, so that questions that require only fairly general answers may be answered rapidly if the answer is clear-cut, but if not, searches for a solution can be directed to critical areas which determine the answer. The representation uses octrees for representing spatially-indexed information, and tables for storing information about objects and their features and properties.

The sensor that is currently of primary interest is a camera mounted on the wrist of the robot. Since the position of the camera is fixed with respect to the robot arm, and the robot's position in space can be obtained at all times, the position of the camera can be computed, and is used in constructing the workspace representation. As the robot moves, the sensor observes different parts of the workspace, and must integrate information over many views to construct a representation of the whole scene.

The spatial representation is constructed from a sequence of two-dimensional images. Initially, it consists of a large cube enclosing the whole workspace, with known objects or regions, such as the work surface or a machine tool, already represented. As pictures are taken, the objects discovered are projected into the cube as generalized cones. The cones describe the possible locations of the objects that gave rise to each component in the image. When pictures are taken from different viewpoints of the same region in space, the cones are intersected to constrain the possible locations of the objects (Figure 1). As data accumulate, the shapes and locations of objects approach their true values more and more closely. The representation always contains all the information known about each region of space, and includes the uncertainties in object positions and shapes.

At the same time that the spatial representation is constructed, a parallel process extracts features from the sequence of images and stores

them in tables. The features are used in matching the objects with a database of models, and as descriptions of unrecognized objects. When an object matches with a model, a lot of information is made available by instantiating the model. This information is used to fill the table, and to refine the spatial representation of the object by projecting the model into the octree. The two representations are linked to allow spatial indexing of objects, as well as locating features and objects in space by name or description.

2. The Representations

The spatial representation views the workspace as enclosed by a cube. The contents of the cube are initially unknown, except for fixed volumes such as the base of the robot or a machine tool. As sensor information is obtained, parts of the cube become known to be empty, or to have objects in them. Objects can be of any size, and it is not practicable to represent the whole cube at the resolution of the smallest possible object. As a result, some structure is placed on the cube, in the form of an octree (Srihari, 1981, Meagher, 1980).

An octree is constructed as follows. Initially, the region of space to be described is represented by a single node in a tree, corresponding to a cube surrounding the space. This cube is examined, and if it is homogeneous, the process terminates. Otherwise, the cube is divided into 8 equal subcubes (octants), which are the children of the node, and the process is repeated for each octant. When all (sub-) cubes are homogeneous (according to some rule), the octree is complete (Figure 2).

For the spatial representation, the original node represents the workspace, and has a standard orientation and position. When the first picture is taken, a set of generalized cones is projected into the cube from a point corresponding to the optical center of the camera. Each cone arises from a separate component in the picture. The space inside the cones reflects the possible locations of objects in the world, while the space outside the cones is background. To decide which octants of the cube to expand involves intersecting the cones with the cube. If an octant intersects a cone, a further check is made to see if it is totally contained in the cone, in which case, its color is simply changed from "unknown" to "object", and it is not further subdivided. If it only partly intersects the cone, it is subdivided, and the same tests are applied to its children. The result is a tree representing the current state of knowledge about the world. Note that labelling a node "object" does not mean that the object actually occupies the corresponding volume in space, but only that the volume lies inside cones arising from an object visible in all views of the region so far.

The intersection process works as follows. A cone arising from an object in the image can intersect with three kinds of terminal nodes in the octree representing the workspace. If a node in the workspace octree is labeled "unknown", its label is changed to "object", reflecting the possibility that there is an object in that location. If a node is labeled "empty" it remains "empty", because the region corresponding to the node must already have been seen from another viewpoint, and must have been seen to be empty. If a node is labeled "object" it retains its label, because no new constraints have been found for the node. The intersection process for cones corresponding to empty space in the 2D image is simple. Projections of empty regions change the labels of all nodes that they intersect in the workspace octree to "empty".

Computing which nodes in the octree intersect with a cone is non-trivial. The difficulties arise because the camera can view the workspace from any position or orientation, and from inside or outside the workspace cube. The approach currently implemented involves first approximating the boundaries of objects in the 2-D images with straight-line segments. These segments are then projected as planes into the octree, and the set of cubes is found that lies inside the volume enclosed by the planes.

The projection approach is similar to that described by Martin and Aggarwal (1983). Their goal was to describe the volumes of individual objects, however, rather than whole scenes. Also, they assumed orthogonal projection rather than perspective projection, an assumption that reduces the complexity of the construction process. Martin and Aggarwal used a representation for the volumes that appears less suited to the task of representing the workspace than does the octree. Connolly (1984) also used a projection technique to construct octrees. In his application range data were used to construct octree models of objects from multiple views. Again, only a single object was modelled in each tree. Connolly first constructed a quadtree from the image, and then projected the quadtree blocks into the octree. It is not clear that this method provides any speedup in the projection process, because the quadtree nodes do not map into octree nodes except in rare instances.

It is not necessary or cost-effective to store shape information at full resolution in the tree. If an octant is small enough and contains mostly object points, it should be considered as being filled by the object for spatial representation purposes. This does not cause a loss of information because of extra information in the tabular representation, which is indexed by the nodes in the tree.

With each node in the tree is associated a set of pointers. The pointers address information concerning the contents of the node. The objects contained in the region represented by the node may have names and features associated with them, or exact geometric descriptions. Such information is obtained both from sensor data and from object models. Pointers to these data serve several purposes. They enable finer discriminations of the space to be made than that set by the resolution of the octree. In addition, they provide spatial

indexing into the sets of object and feature descriptions.

The object and feature descriptions are organized as tables, which store all non-spatially-indexed knowledge. Each entry has slots for object names, locations, properties, and features. There may be more than one name for each object in the world, and more than one instance of each kind of object. Each name entry has a confidence associated with it. When this confidence goes below some threshold, the name is removed. If all confidences go below threshold, the entry in the table is removed if the object has disappeared from the scene (e.g., if it is a noise region that no longer appears in an image). Otherwise, it is labeled as "unexpected".

Two table-based representations are used, one for data describing models, and one for scene data and instantiated model data (hypotheses and recognitions). The major difference between the tables is that the model table holds generic information in object coordinates, while the scene table holds information about particular instances in world coordinates. The scene table has confidences in various matches, and pointers to particular nodes in the octree and to entries in the model table. Each object expected to appear in a scene has a row in the model table, with columns for feature types and values, to be used in recognizing the objects. There is also a pointer to the geometric descriptions of the objects, obtained from a CAD database.

Each feature entry points to a list of the particular values for that feature (e.g., positions and angles for corners, in object coordinates). The table can easily be set up at the beginning of a task by examining models of all the objects expected to be seen during execution of the task.

The table constructed for data extracted from the scene is similar to that for the model data, but has extra columns for pointers to the octree and for tentative identifications and their confidences. Each row of the table corresponds to an object in the world. Objects can be single regions in space, groups of regions, or hypothesized objects. The columns in the table hold pointers to particular instances of features or instantiated models.

Each identification is either an index into the table of models or the label "unexpected". Confidences are obtained from the model matching process. The entries for pointers to the octree address lists of octree nodes in which the objects appear. The entries for features have the same form as those for model features, except that their values are instantiated using information from the scene.

3. Using the Representations

There are two main aspects to using the representations. The first is updating it and ensuring that it contains current information. The second is answering questions about the world and the objects in it.

Updating the tree involves moving objects as they are picked up by the robot, (and in later versions of the system, as they move), and adding new objects as they appear. Removing an object is simple. The leaf nodes that represent it are located in the tree from their addresses in the table entry for the object. Their color is changed from object to background, and any merging that can be done as a result is performed. The pointers of all parents of the nodes are updated to reflect the disappearance of the object, and its entry in the table is removed. New objects are added automatically by the process of projecting from components in the 2-D images.

When an object first appears in the spatial model, an entry is set up for it in the table. If the object arises from an hypothesis, the slots are all filled in immediately. Otherwise, slots are filled in using values obtained by feature extraction techniques. Before an object has been recognized, the name list contains a flag indicating that its identity is unknown. As objects are recognized, or more information is discovered from sensory input, the remaining slots are filled. When an object disappears from the world, its entries are erased and the space is made available for later use. Objects can also be merged if they are found to be connected. Merging is a simple operation in the table, involving the coalescing of the various table entries. A problem arises when the same set of features lends support to more than one hypothesis. For example, a region in space may tentatively be identified as one of a number of known objects. The confidences in each of the identifications depends on the features that match with the corresponding object models. To decide which is the best identification requires a relaxation labelling process to establish the most globally-consistent set of matches (Mackworth, 1977).

Answering questions is fairly straightforward. To find out about the world it is not necessary or desirable to interrogate the sensors directly. The representation contains both predictions about the world and information obtained from sensors. There is no direct way of distinguishing between information from different sources. Answers to questions contain both empirical and hypothetical elements. Answering questions about space involves checking the octree for the color of the node corresponding to the requested region. Questions that refer to specific features or surfaces are answered by reference to the table. Finding all occurrences of a particular feature involves scanning a column of the table. Depending on the generality of the question, this can be very simple or can require searches of lists of features stored at each position. In either case, though, the search is limited to a small, well-defined subset of the known information. To find out what features or objects occur at a given location, the octree is traversed to find which nodes span that location. The pointers from the nodes are followed, and an answer found in the tables.

Identifying (naming) features and objects is also simplified. If a sensor perceives an object, and the projection of the object intersects with an already-identified or an hypothesized object, then simple feature checking can establish a confidence in the identity of the object. If there is no intersection, the properties and features extracted by the sensor can still be matched with those from the object models, by comparing corresponding entries in the table.

4. Conclusions

The system described above is currently being implemented as part of an hierarchical sensory-processing system that interacts with an hierarchical robot control system to perform tasks requiring real-time sensory guidance. The lower-level image-analysis and feature-extraction algorithms have been implemented using a network of microprocessors that operate independently and asynchronously (Kent, 1982, Shneier, 1982). The output of the lower levels forms the input necessary for constructing the spatial and feature-based representations, and for matching objects with their models. The models currently in use are hand-crafted, but a computer-aided design system is now available, and a database of models will soon be generated and interfaced with the workspace modelling system. The result will be a flexible representation scheme that should allow real-time responses to questions of significant complexity.

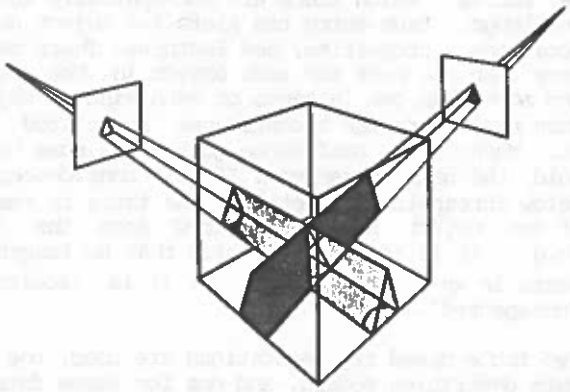

Figure 1. Intersecting cones from two views of an object.

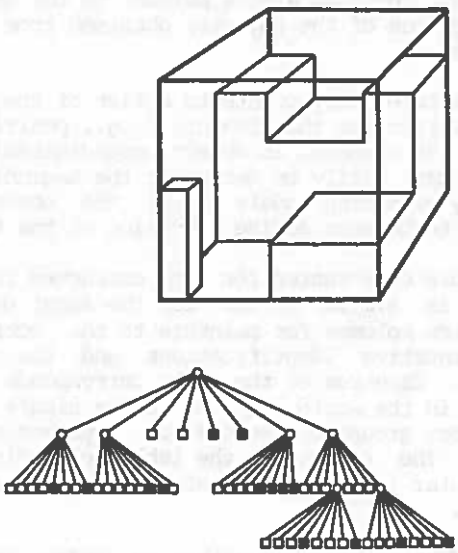

Figure 2. Objects enclosed within a cube, and the octree representing the volume of the cube.

References

1. C. I. Connolly, Cumulative generation of octree models from range data. Proc. International Conference on Robotics, Atlanta, Ga, March 1984, 25-32.

2. E. W. Kent, A hierarchical, model-driven, vision system for sensory-interactive robotics. Proc. Compsac '82, Chicago, November 1982.

3. A. K. Mackworth, Consistency in networks of relations. Artificial Intelligence 8, 1, 1977, 99-131.

4. W. N. Martin and J. K. Aggarwal, Volumetric descriptions of objects from multiple views. IEEE Trans. PAMI 5, 2, March 1983, 150-158.

5. D. Meagher, Octree encoding: a new technique for the representation, manipulation and display of arbitrary 3D objects by computer. Tech. Rep. TR-IPL-111, Dept. Electrical Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, 1980.

6. M. Shneier, 3-D robot vision, Proc.International Conference on Cybernetics and Society, Seattle, Wa, October 1982, 332-336.

7. S. N. Srihari, Representation of three-dimensional digital images. ACM Computing Surveys 13, 4, December 1981, 399-424.