N90-16221
554753 p10

A KNOWLEDGE-BASED MACHINE VISION SYSTEM FOR SPACE STATION AUTOMATION[1]

Laure J. Chipman and H. S. Ranganath
Computer Science Department
University of Alabama in Huntsville

INTRODUCTION

A simple knowledge-based approach to the recognition of objects in man-made scenes is being developed. Specifically, the system under development is a proposed enhancement to a robot arm for use in the space station laboratory module. The system will take a request from a user to find a specific object, and locate that object by using its camera input and information from a knowledge base describing the scene layout and attributes of the object types included in the scene.

In order to use realistic test images in developing the system, we are using photographs of actual NASA simulator panels, which provide similar types of scenes to those expected in the space station environment. Figure 1 shows one of these photographs.

In traditional approaches to image analysis, the image is transformed step by step into a symbolic representation of the scene. Often the first steps of the transformation are done without any reference to knowledge of the scene or objects. Segmentation of an image into regions generally produces a counterintuitive result in which regions do not correspond to objects in the image. After segmentation, a merging procedure attempts to group regions into meaningful units that will more nearly correspond to objects.

Rather than taking this approach, we avoid segmenting the image as a whole, and instead use a knowledge-directed approach to locate objects expected in the scene. Constraints on the spatial relationships among objects and on attribute measurements of object types are used in obtaining a matching between regions of the input image and object descriptions in the knowledge base.

Section 2 describes the knowledge-based approach to scene analysis. Section 3 discusses the categories of knowledge used in our system. The remainder of the paper is a step by step description of the system under development.

KNOWLEDGE-BASED APPROACH

The use of a knowledge-based approach to object recognition is a growing area of research in image analysis. Use of knowledge improves recognition accuracy. We seek to avoid embedding this knowledge in the code, in order to create a more flexible system.

---

1 This research is being supported by NASA Contract NCC8-16.

Knowledge of objects is traditionally used in the later stages of image analysis to match regions of an image with known objects. We are exploring the use of knowledge at earlier stages of the processing to help guide the search for objects.

A goal of our work is to provide a flexible system for locating objects, which could be updated for new scenes by simply adding to the knowledge base. The knowledge of scenes and objects is stored explicitly, rather than being embedded in the system's code. Objects and scenes are described in a general way, so that the system will not be overly sensitive to changes in the camera position or illumination. It is desirable to avoid exact models of the objects of interest. Some of the objects on the panels may be difficult to describe with a precise geometrical model. For example, the panel in Figure 1 contains switches that are enclosed in protective brackets. Because of their complicated structure and the existence of shadows, objects such as these will show up in the gradient image as a tangle of lines, easy to recognize but difficult to model geometrically.

There are many systems designed to match regions of an image to descriptions of objects stored in a knowledge base. McKeown's SPAM (System for Photo interpretation of Airports using MAPS) is one example [1]. This system takes the result of a traditional region-growing segmentation and attempts to group segments into meaningful objects. Levine and Shaheen describe a system in which segmentation is based on color, and regions are merged to form objects based on a long list of constraints on attribute measures of different object regions [2].

CATEGORIES OF KNOWLEDGE

For our application, the following categories of knowledge are used:
1) Knowledge of primitive, scene-identifying features
2) Measurement ranges of attributes of object types
3) Knowledge of spatial layout of scenes

In the first category, information about features consists of a list of procedures to be used to find the features, and parameters for these procedures. The scenes are described as lists of features that are present and absent from them. The information in this category was obtained through experimentation with input images. There is a need to develop an automated method for finding discriminating features for any new scene presented to the system.

The second category of knowledge consists of object types and ranges of acceptable values for attributes of those object types. The attributes used will preferably be invariant to scale or illumination changes and relatively insensitive to rotation. Such attributes as a texture measure, circularity, rectangularity, or ratio of length to width are good possibilities. A significant, but manageable, programming project would be to automate the gathering of these object attribute ranges, using a teacher to draw windows around several objects of a given type, and having the system automatically make and record measurements.

The third knowledge category contains information that aids in finding the starting points of probable objects. It contains the layouts of regions of the scenes to which input images will be matched. The knowledge in this category can help resolve ambiguities in the classification of objects by using spatial constraints. In other systems, this category could be expanded to include other types of constraints on the relationships among objects, such as adjacency or inclusion.

STRATEGY

The steps of the processing in our system are shown in Figure 2. The user indicates an object of interest. The system first verifies that the input image is a view that should contain that object. In the knowledge base, inclusion lists specify what panel contains each of the possible objects of interest. From this, we can determine which panel should be present in the input image. There is also a list of primitive features and parameters for procedures to find them listed in the knowledge base. The second step is then to identify which scene, or panel, is represented by the input image. The image is searched for distinguishing features to determine which scene is present. If the input image contains the scene of interest, we proceed to locate the object of interest. If not, the camera would be relocated to find the desired scene.

Once the proper scene is present, we find a region of interest within the scene. This region will contain the object of interest. The region of interest can be found relative to the location of the features found in the image in the scene identification step. Once the region is found, the camera can be made to zoom in on this area.

Within the region of interest, probable starting points to locate objects are found. Then, the boundaries of probable objects are found by searching windows around these starting points. Attributes of these probable objects will be measured. The knowledge base will list attributes of the different object types that are easy to recognize and identify. For each probable object in the region of interest, we obtain a list of object types for which the attribute measures match. Then, a matching between the input image and the scene layout described in the knowledge base must be found.

Although the actual procedures used for finding seed points, measuring attributes, and matching objects are specific to our application, these three steps could provide a useful starting point for other applications. For other image types, there could be other procedures developed for performing essentially the same three steps.

Preprocessing and Scene Identification

The first step in our processing is to obtain an edge image using the Sobel edge operators. This is done to facilitate locating boundaries of objects.

In our application, it is not likely that any significant rotation of the image will occur, since the camera will be mounted on a robot arm attached to a rail that runs the length of the module. Since the robot can know which end is "up," rotation is not a problem. In other cases, a system may need to deal with this possibility. For images of man-made objects such as control panels, a possible approach is to search Hough transform space for lines of maximum intensity. In scenes of control panels these are generally horizontal and vertical lines. Knowledge of the expected scene could also be used to determine at what angle the lines of maximum intensity should appear in the input image. This can be used to rotation-normalize the image.

Next, we identify which scene is present. We are assuming that an input image will contain one of a number of separate scenes. If the image contains parts of more than one scene, the process will generally not produce useful results. This goes along with the assumption that a camera attached to a robot arm could be positioned at a number of discrete, although approximate, positions along the length of the space station lab module.

To identify the scene, the system searches for primitive distinguishing features. The presence or absence of the features in the input image is matched with lists of features present for each scene in the knowledge base. Presently, a scene must have features that match exactly with one of the scenes in the knowledge base. It would be possible to allow for closest matches by computing the string distance between a binary string denoting presence and absence of features in the input image with the strings in the knowledge base, and assume the scene to be the one with the closest match.

The features used for scene identification are sets of lines in the gradient image with certain characteristics. These lines correspond to edges in the original image. A line in our system is defined in terms of a merit measure which is a linear combination of average intensity and average difference between successive pixels along the line. A row of pixels of high intensity and low average difference is a "good" line. The characteristics of intensity and average difference can be useful taken separately. The average difference measure provides a good measure of texture which is easy to compute. Some of the features used for scene identification are lines of high average difference.

Figure 3 depicts the scene identification process for the panel of Figure 1. In this example, lines of high texture, as measured by high average difference, are found through the columns of an array of lights, and also through a row of switches. The diagonal line and the set of lines in the upper left corner of the image represent the best matches for two additional features that are present on other panels but not on this panel.

We use primitive features to keep processing for scene identification to a minimum, but any features could be used, as long as the process for finding them could be listed in the knowledge base.

Object Seed Points

Given the location of features in an input image, it is possible to compute coordinates for a region of interest of the scene that contains the desired object.

Once an image of the region of interest of the scene is obtained, we find starting points of probable objects. In scenes consisting of well-separated blobs on a background, a method that has proven useful is to search for a specified number of horizontal and vertical lines of high texture, with some minimum spacing between them. Figure 4 shows the result of this process on one of our regions of interest. Most of the intersection points pass through objects on the image. There are some false lines, since there is some printing on the control panels that results in high-texture lines.

The minimum spacing chosen is large enough to prevent the appearance of more than one line in the same direction through the same objects. Only a minimum is given so that the object seed points may be found for images that are translated or scaled differently.

The intersection points of lines found are possible object locations. For other types of images, other methods for finding seed points of objects could be used. If an object's color is known, the image could be searched to find a patch of that color as a starting point for a region-growing routine. Likewise, any other attribute of an object, such as intensity or texture could be used to find a patch from which to start a region-growing routine. This may be better than performing a global segmentation and growing all possible regions in the image, which probably do not correspond to objects.

We are experimenting with methods of finding object boundaries within windows centered on the seed points. The use of these seed points can reduce computation by limiting the search area for objects.

Measurement of Attributes

The control panels contain instances of a finite number of object types, e.g. switches, buttons, knobs, etc. For each object type, the knowledge base contains an acceptable range of values for each attribute. The attributes used may differ depending on the object type. For example, circularity may be a good attribute to use for knobs, but texture may be better for switches enclosed in brackets. Since the possible objects in the input image may be processed in parallel, it may be worthwhile to measure all attributes, even though some results may be not be used. Once the attribute measures have been determined, the knowledge base is consulted to determine for each possible object the set of object types consistent with its measurements. For example, Object 1 may "look like" a switch or a button. Some possible objects will not match to any object types.

The result of this process is a grid showing the possible objects which could be located at each point.

Matching Scene Layout

Our system will match the input image with a grid layout of the region of interest in the knowledge base. The points on the grid correspond to intersection points of lines passing through the objects. Some points will not correspond to any object, but pass through empty space.

Figure 5 shows examples of knowledge base and input grids. The input image will be processed to produce a grid layout of what is found. The matching routine will find a consistent match between the knowledge base grid and the input grid. In general, the input grid may have more rows or columns than the knowledge base grid. There may be non-object points in the input that happen to look like a certain object type based on their attribute measures. The constraint of the layout given in the knowledge base will help to find a consistent matching. In theory, there could be more than one consistent matching for a given scene, but the fact that both attribute measures and scene layout constraints are used will reduce the chance of an incorrect matching.

We are producing a deterministic matching routine, but this may be expanded to find a closest match, thus enabling the system to handle partially occluded objects or problems with glare.

The constraint of scene layout, meaning left-right, above-below relationships is not the only constraint that could be used to find a consistent match. There are other scene attributes that can be represented in graph form that constrain the interpretations of the scene. Adjacency and inclusion relationships are two examples.

There has been some work done to develop a theoretical basis for graph matching. Shapiro and Haralick [3,4] have developed a graph theoretic method of partial matching, using distances between graphical descriptions of input images and those on file for known images. They apply this approach to matching relational descriptions of objects with their descriptions stored in a knowledge base. The same idea can be applied to matching relational descriptions of scenes in which the objects are stationary. This is a promising approach for handling problems of missing or occluded objects or variations due to noise. More flexibility is provided by the matching of attributed graphs. Sanfelieu and Fu [5] have described a distance measure between attributed graphs which may be useful. Their work is applied to syntactic recognition of objects, but could also be applied to graphical descriptions of scenes.

CONCLUSION

A system that uses knowledge of scenes and objects to aid in segmentation and location of desired objects is being developed. The system is not based on any geometrical modeling of objects or on precise measurements of object location. A goal was to make the system relatively insensitive to changes in camera position and illumination, taking into account the fact that a robot's positioning system will not be perfect. The approach used is most applicable to scenes in which objects are stationary and well-spaced, such as control panels.

The usual approach to object recognition is to segment the entire image and then try to make sense out of all the segments by matching them to known objects. In our approach we eliminate needless processing of segments that do not correspond to known objects. We change the focus of attention of the system based on information about the scene layout, to match up only objects that will assist in finding the object of interest. Once we have a consistent mapping of areas of the image to known objects, we have completed processing. Other features in the image are ignored.

Although this system is designed specifically to process man-made scenes such as control panels, in which objects are usually well-separated on a background, the basic idea can be generalized to other applications. In any application in which the scenes consist of fixed objects or regions, knowledge of scene layout can be used to direct the segmentation process and to constrain possible interpretations of the objects found in the scenes. Different approaches can be found for determining object seed points, and then for growing regions from points identified as being likely parts of objects of interest. Different attributes of these regions can be measured for different applications. Constraints on relationships among objects other than the simple spatial layout can also be used.

REFERENCES

[1] D. McKeown, Jr., W.A. Harvey, Jr., and J. McDermott, "Rule-based Interpretation of aerial imagery," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-7, No. 5, 1985, pp. 570-585.

[2] M. Levine and S. Shaheen, "A modular computer vision system for picture segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-3, No. 5, 1981, pp. 540-556.

[3] L.G. Shapiro, R.M. Haralick, "A metric for comparing relational descriptions," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-7, No. 1, 1985, pp. 90-94.

[4] L.G. Shapiro, R.M. Haralick, "Structural descriptions and inexact matching," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-3, No. 5, 1981, pp. 504-519.

[5] A. Sanfelieu, and K.S. Fu, "A distance measure between attributed relational graphs for pattern recognition," IEEE Trans. Systems Man Cybernetics, SMC-13, No. 3, 1983, pp. 353-362.

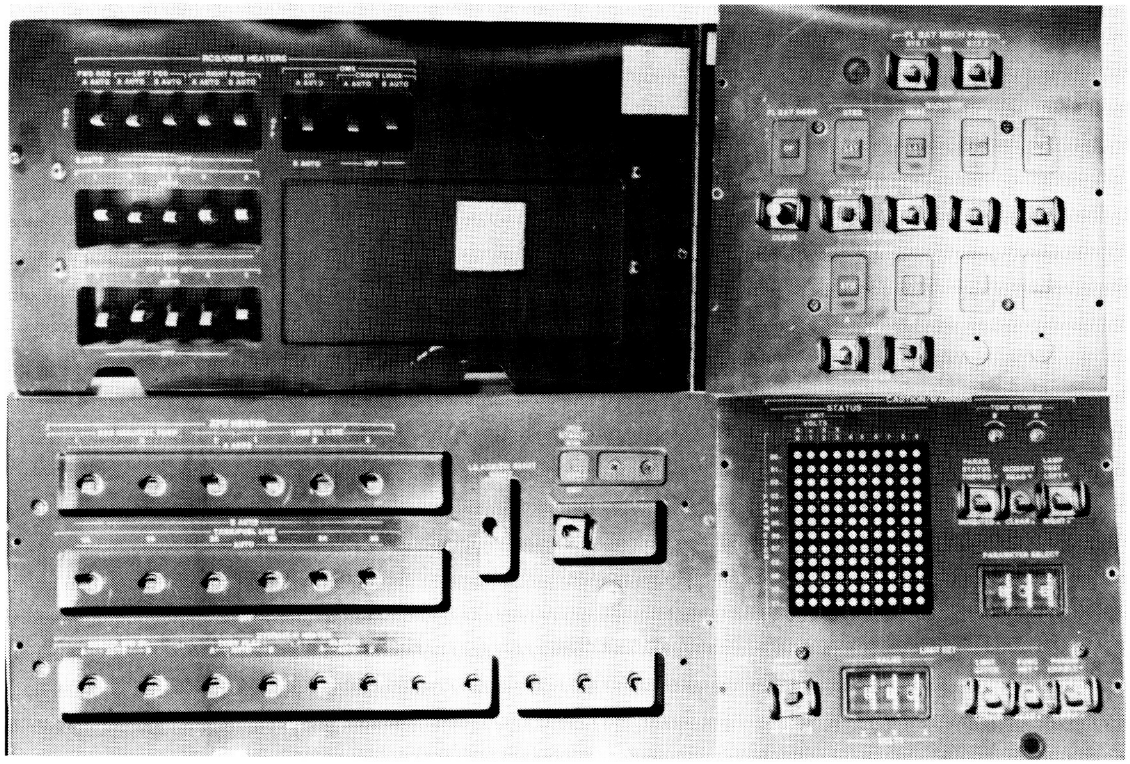

Figure 1: One of the scenes used as a realistic test image for the system.

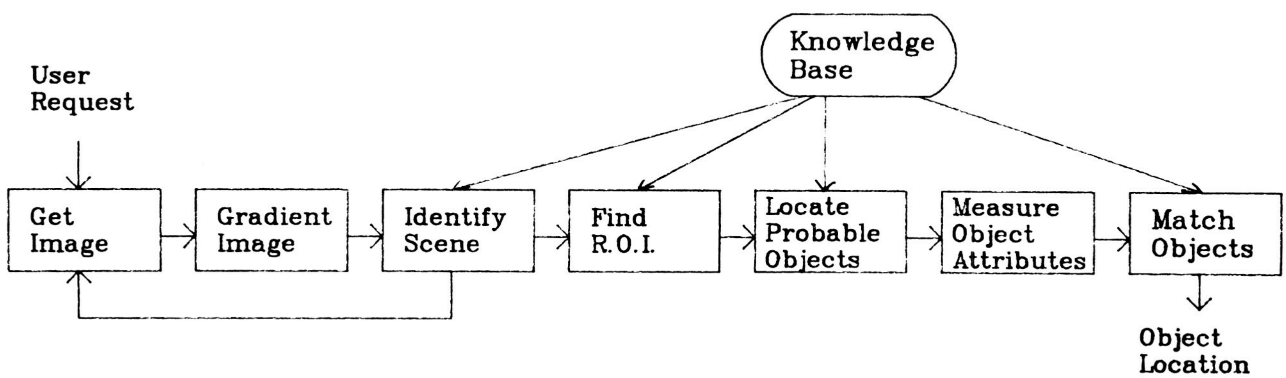

Figure 2: The steps in the object recognition process.

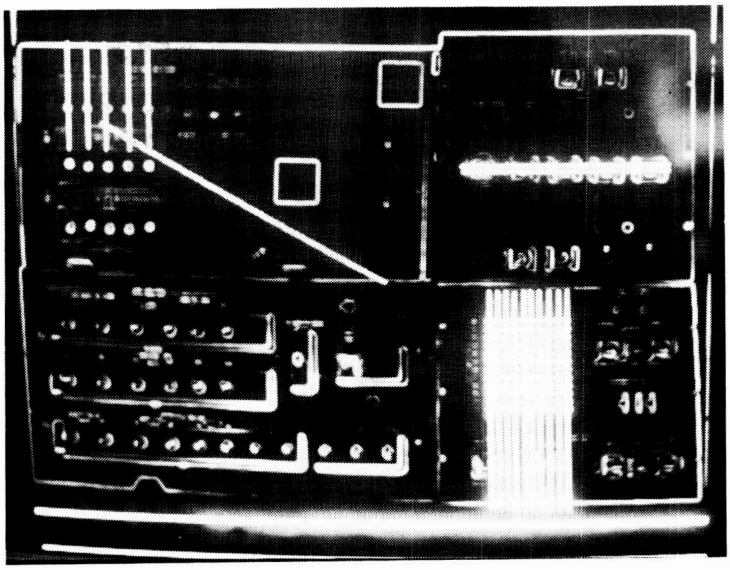

Figure 3: The scene identification process performed on the gradient image of the panel in Figure 1.

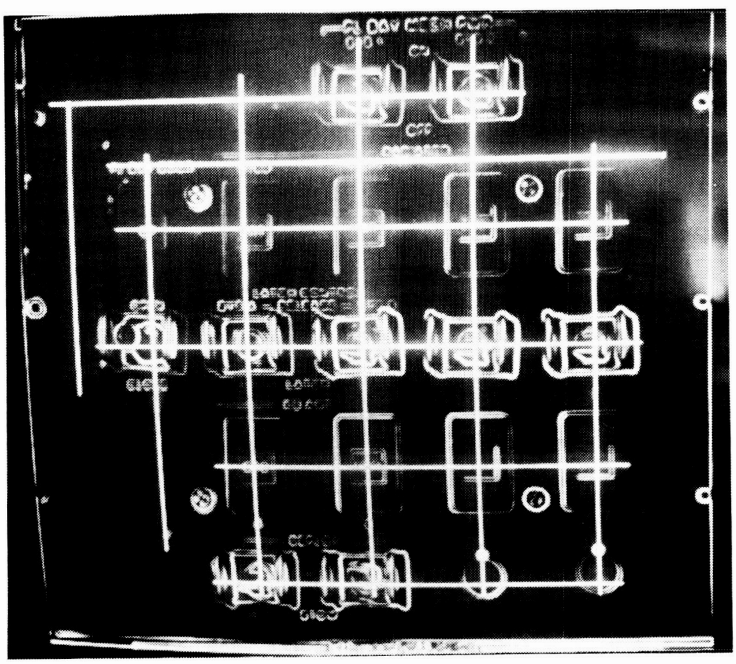

Figure 4: Determination of likely starting points for objects, performed on a sub-panel of Figure 1.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

INPUT GRID

--	--	Switch, Knob
Button	--	--
Rectangle	Rectangle	Rectangle
Switch, Button	Switch	Switch, Button

KNOWLEDGE BASE GRID

--	--	Switch 1
Rectangle 1	Rectangle 2	Rectangle 3
Switch 2	Switch 3	Switch 4

Figure 5: A hypothetical grid representing possible object layout in an input image, and a grid from the knowledge base to be matched to it.