RECOGNITION AND SPATIAL ORGANIZATION OF OBJECTS IN NATURAL SCENES

by

Robert J. Douglass

Computer Sciences Technical Report #317

February 1978

R E C O G N I T I O N A N D S P A T I A L O R G A N I Z A T I O N O F O B J E C T S I N N A T U R A L S C E N E S *

Robert J. Douglass
Computer Sciences Department
University of Wisconsin-Madison

This work is partially supported by the National Science Foundation, NSF Grant MCS76-07333, and the University of Wisconsin Graduate School.

Abstract

A computer vision model for recognizing objects in real world scenes and locating them in three-dimensional space is described. It is intended for applications which require navigation through an environment and interaction with objects in it. The system uses a recognition cone for feature extraction and preliminary recognition, a segmentation algorithm, and a routine for constructing a three-dimensional world model. Several types of visual knowledge are incorporated into the system including long-term object models, short-term object representations, and general routines for interpreting perspective, shadows, highlights, occlusions, and texture gradients. The vision model is currently implemented in SIMULA and is being tested on near views of outdoor scenes. Results are presented for one scene of a house and yard. The design of a monocular motion parallax routine is given which will permit the system to integrate several views of one scene over time.

Key words: computer vision, recognition, natural scenes, scene analysis, depth perception, image processing.

Introduction

A number of computer vision applications require a perceptual system capable of identifying salient objects in a scene, developing an understanding of their spatial relationships, and maintaining continuity from one view to the next as either the system's camera or objects move through the scene. Typical of such applications are the computer controlled chauffeur proposed by McCarthy in Baumgart (1), an extra-terrestrial explorer as outlined by Lewis and Bejczy (2), a navigational aid for the blind, or an industrial robot which must move through and interact with its environment as in Finkel et al. (3). Recently work has begun on a number of complete, relatively general vision systems which will interpret complex natural scenes in terms of object names from a digitized picture of a scene (4,5,6,7,8,9,10). These systems are a first step toward the type of perceptual front end required by the above applications. However, most of these programs either develop only a weak notion of spatial organization or ignore the problem entirely. Marr proposes a system with strong spatial vision but separates spatial organization entirely from object identification (6). In goals and design the system described in this paper is most similar to the models of Riseman and Hanson (9) and of Sakai et al. (10).

This vision system extends the work of Uhr (4), Riseman and Hanson (9), and Sakai et al. (10) by combining

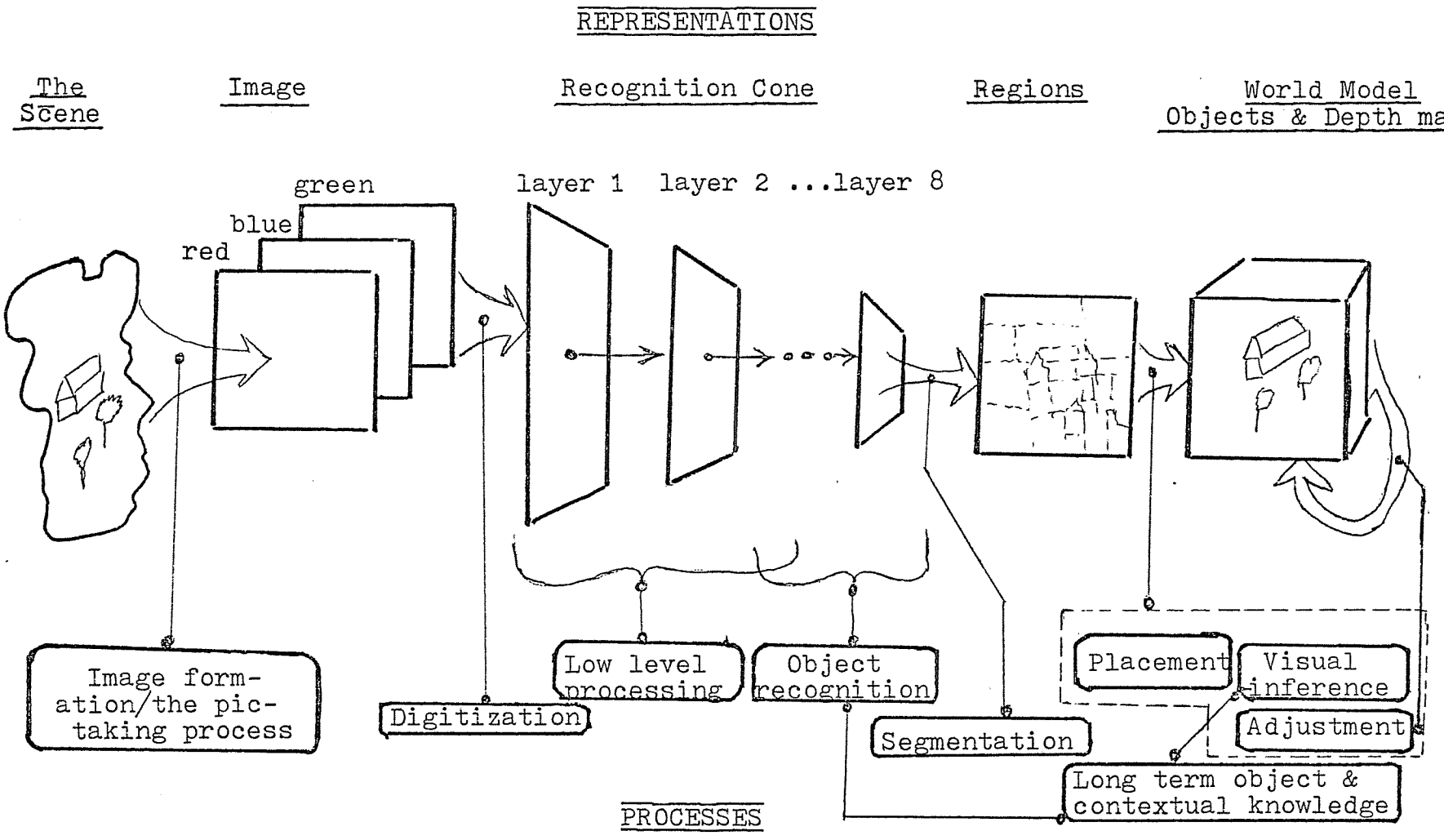
several different depth cues to build a three-dimensional world model from an image of a scene. The system incorporates the recognition system of Uhr (4,11) and a segmentation algorithm to divide a scene into regions. Visual knowledge routines acting as depth cues and a relaxation like adjustment routine for combining the cues cycle over the regions to stitch them together into distinct instances of objects in the world model. The world model provides a spatial understanding of a scene as well as representing it in terms of object names. In addition, the world model is a basis for integrating multiple views of a scene over time. The overall design of the vision model is given schematically in figure 1 with data representations listed across the top and processes across the bottom.

Overview of the Scene Description System

The vision model described in this paper, with the exception of the motion parallax routine, is coded in SIMULA and is running on a UNIVAC 1110 computer at the University of Wisconsin-Madison. It accepts digitized pictures and produces a description of the names of the objects in view, their visible surface characteristics, and a three-dimensional model or map of the scene. The system's analysis and feature extraction program is designed to process a 600X800 pixel (picture element) image where each pixel is a triple of eight bit values of red, green, and blue light intensity. The system is currently being tested on near views of outdoor scenes

F I G U R E 1 .

A MODEL FOR COMPUTER VISION



containing objects such as houses, cars, streets, grass, and trees. Figure 2 is a typical input scene; it is the house scene used by Ohlander in his thesis (12) and more recently by Schacter et al. (13). When the motion parallax routine is completed, the program's input will be a series of snapshots or views of the same scene taken from different camera positions like the succession of views an automaton would receive while moving down a street.

The output from the system consists of a three-dimensional model and a set of objects and their descriptions. For example, figure 3 shows a depth plot of the house scene from figure 2 produced from the world model. Figure 4 represents the area of the scene identified by the program as forming a single house.

The program consists of a recognition cone after Uhr (4), for feature extraction and an initial assigment of object names to areas of the scene, a segmentation routine, a short-term model builder, a set of long-term object models, and a set of routines embodying general visual knowledge such as information about perspective, shading, and occlusions.

The recognition cone is a parallel-serial cone structure consisting of a number of processing layers. The first layer or retina of the cone contains the three color pixels of light intensity as digitized from the system's camera. Successive layers of the cone average the picture, compute several measures of texture and color, and detect edges and angles.

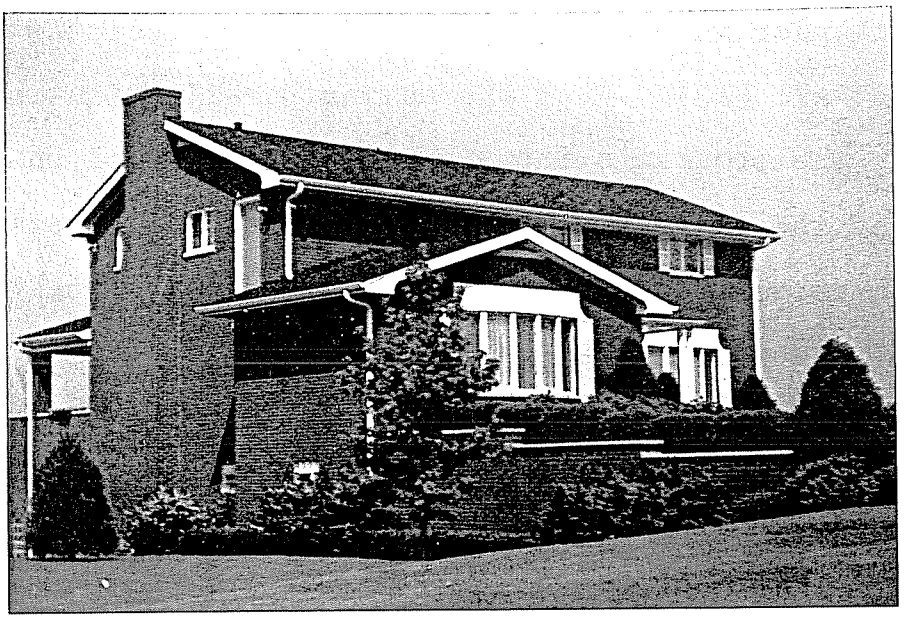


Figure 2. A typical input scene for the vision program.

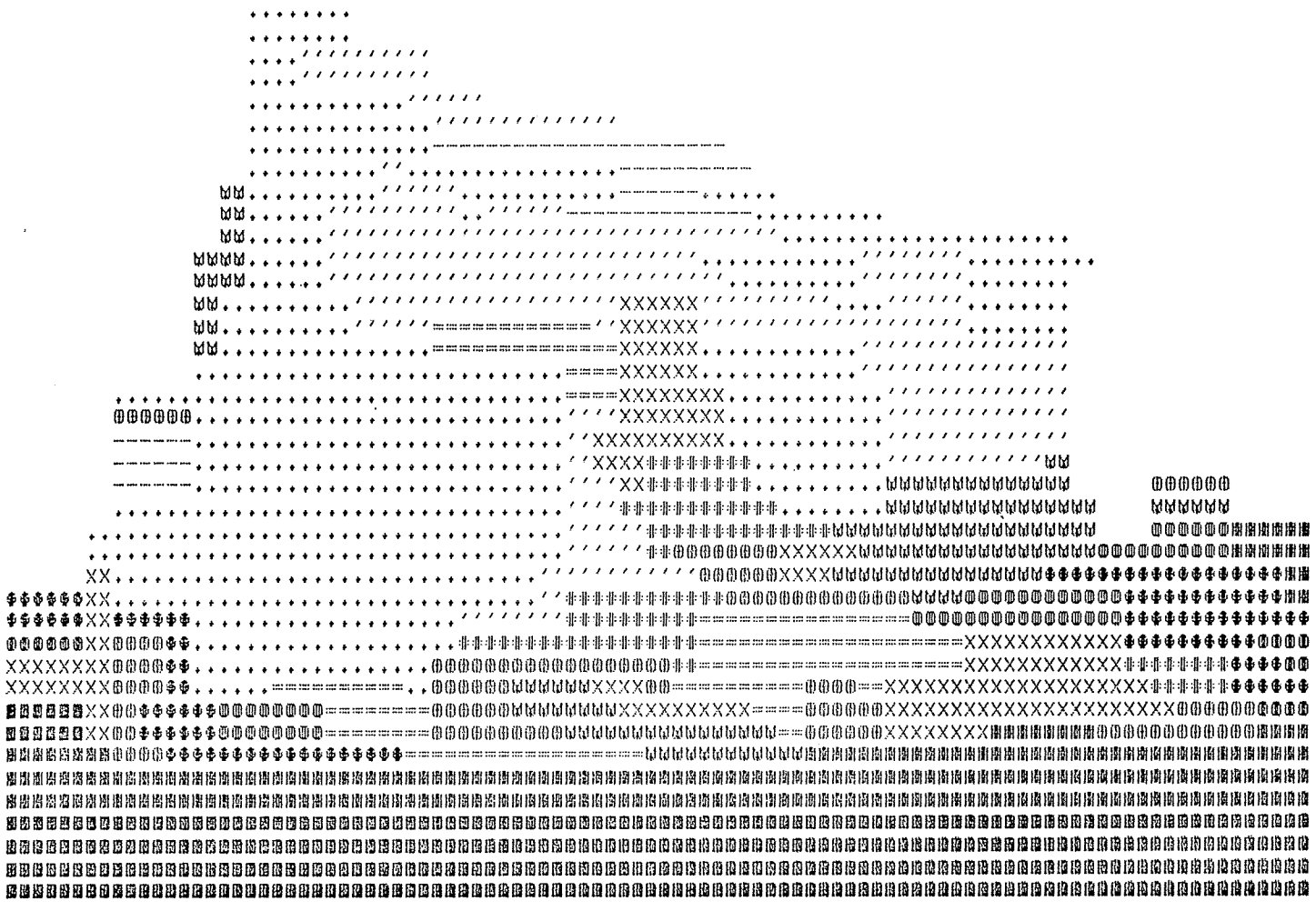


Figure 3. A depth plot of the house scene shown in figure 2, produced from the world model. The darker the shade of a point in the plot, the closer it is to the camera.

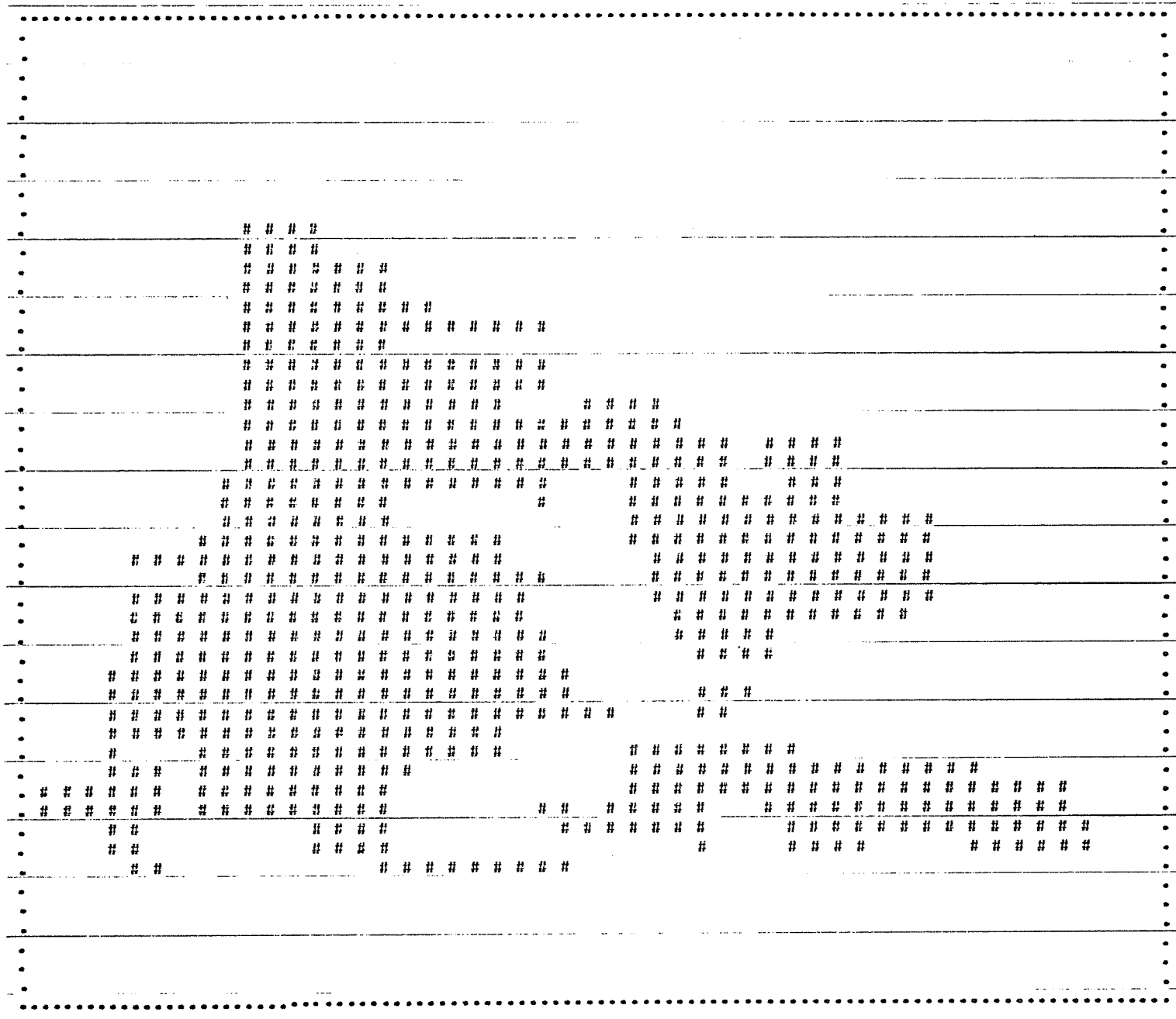


FIGURE 4. AREA OF THE SCENE IN FIGURE 2 IDENTIFIED AS ONE HOUSE BY THE PROGRAM.

In the higher layers of the cone, object names are assigned to various areas based upon the presence of certain configurations of edges and lower level features. The last layer is a 50X50 array containing a number of textural, color, and edge descriptors at each point of the array and one or more possible interpretations for that point.

The recognition cone's output is coarsely segmented by first using the Zobrist-Thompson grouping operator (14) to estimate the probability of an edge between two points in the picture array, and then using Yakimovsky's (15) one pass segmentation algorithm as modified by Nagel (16). A description is formed for each segment including texture, color, size, adjacent segments, brightness, and a list of possible interpretations for that segment.

The short-term model builder is a routine which places the segments produced by Yakimovsky's algorithm into a three-dimensional model of the scene. The model builder forms the heart of the vision system. It uses a long-term memory model of objects, the general visual knowledge routines, the segment descriptions, and the previous contents of short-term memory, if any, to form the segments into three-dimensional surfaces representing objects in the scene. The short-term memory model also contains the description of the visual surface characteristics of the objects and parameters specifying the location and orientation of the system's camera and source of illumination.

Long-term memory models of objects include two types of information. Most of the specific data on allowable shape, color, texture, and relationships between the pieces of an objectare distributed throughout the recognition cone in the form of configurational transforms. The other kind of object information is individual models for each object which provide a very general description of shape, size, orientation, and expected context for the object within a scene.

Barrow and Tenenbaum (17) say that a scene analyzer must have stored explicitly or implicitly information about the picture taking process, that is, the relationship between the visual properties of the physical surfaces in the real world and their manifestations on the program's digitized input array. This knowledge is represented explicitly in the present system with general visual knowledge or visual inference routines. These contain information used by the model building routine to resolve problems of occlusion, shadows, highlights, texture gradients, and perspective distortions in terms of the objects' positions in the three-dimensional space of the scene.

This system is designed to handle a sequence of slowly changing views of one scene over time. The processing of one view at a particular instant of time can affect future processing in several ways. First, the type and amount of processing that the recognition cone does can be altered by adding or deleting transforms that look for features characteristic of particular objects. Second, the short-term memory model of

the scene built up over previous views is used to change the interpretation of transforms in the recognition cone and adjust the weights in the Zobrist-Thompson grouping function to improve segmentation. The model is also used to help decide on one object name where the recognition cone has provided several alternatives. Finally, the previous contents of the model are used to compute an approximate depth based upon a camera model and motion parallax.

Recognition Cone: Scene Analysis and Initial Recognition

This system uses a recognition cone developed by Uhr to characterize the initial light intensity data in terms of several textural, color, and edge measurements, and to generate a set of possible interpretations or object names for points in the cone based on the presence of specific shape and combinations of surface features. Its structure and operation are described in Uhr and Douglass (18) and Uhr (11) and will only be summarized here.

The cone is a sequence of layers or arrays of cells where each cell is itself a list of data and transforms. The type of data contained in a cell depends upon the layer of the cone. For example, cells at the first layer contain the brightness values for each of the three primary colors while cells at layer five contain several textural measures, edges, and combinations of edges. Transforms are procedures which compute a value or search for a combination of features in a set

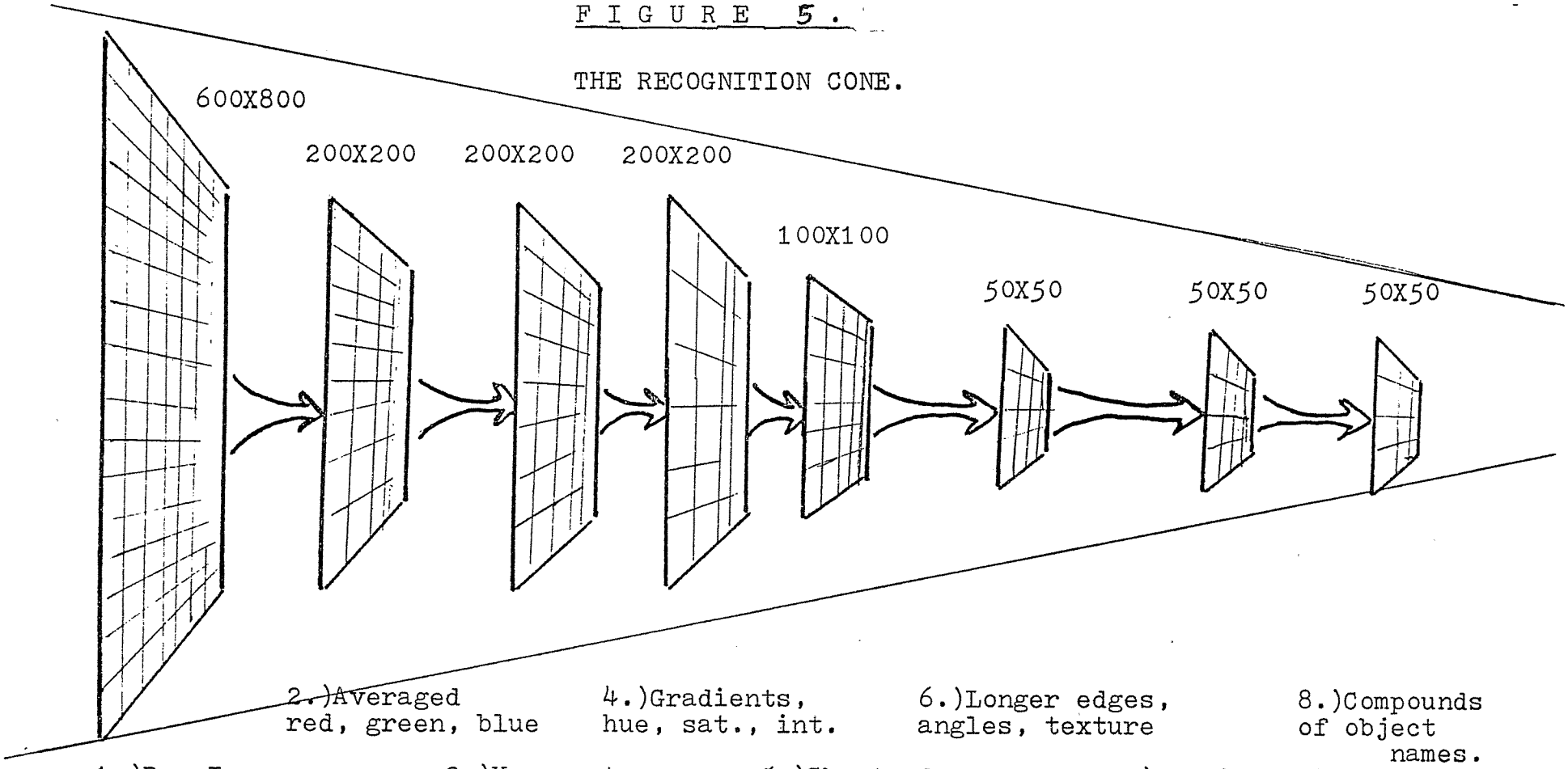
of cells in one layer and output a value or implied name into the next layer. All transforms in any one layer operate logically in parallel and independently of each other. The layers are shown graphically in figure 5, and the construction of one layer and of a cell is shown in figure 6.

The low level feature extraction performed by the cone includes color, texture, and edge detection. Color is measured by approximations to hue, saturation, and brightness (19). Mean values of hue, saturation and brightness over a 4X4 window are used as approximate first order statistics of texture (see 20). Several methods of computing gradients of edge point strength have been used, but a simple Roberts cross gradient (12) on the brightness values alone is currently used along with a threshold to find edge points. Mean values of edge strength, a count of edge points above the threshold, and an approximation to the standard deviation of edge point strength about the mean over a window are used as second order statistics for measuring texture (20).

Edge points that are nearly colinear are grouped into short line segments and the short line segments are in turn compounded to form longer edges and angles. The short line segments are used to compute two further measures of texture. These are the mean number of edges per 4X4 window and the first two most numerous edge orientations on a window. Principle line orientation was found to be a strong texture discriminator by Zobrist and Thompson for their data (14).

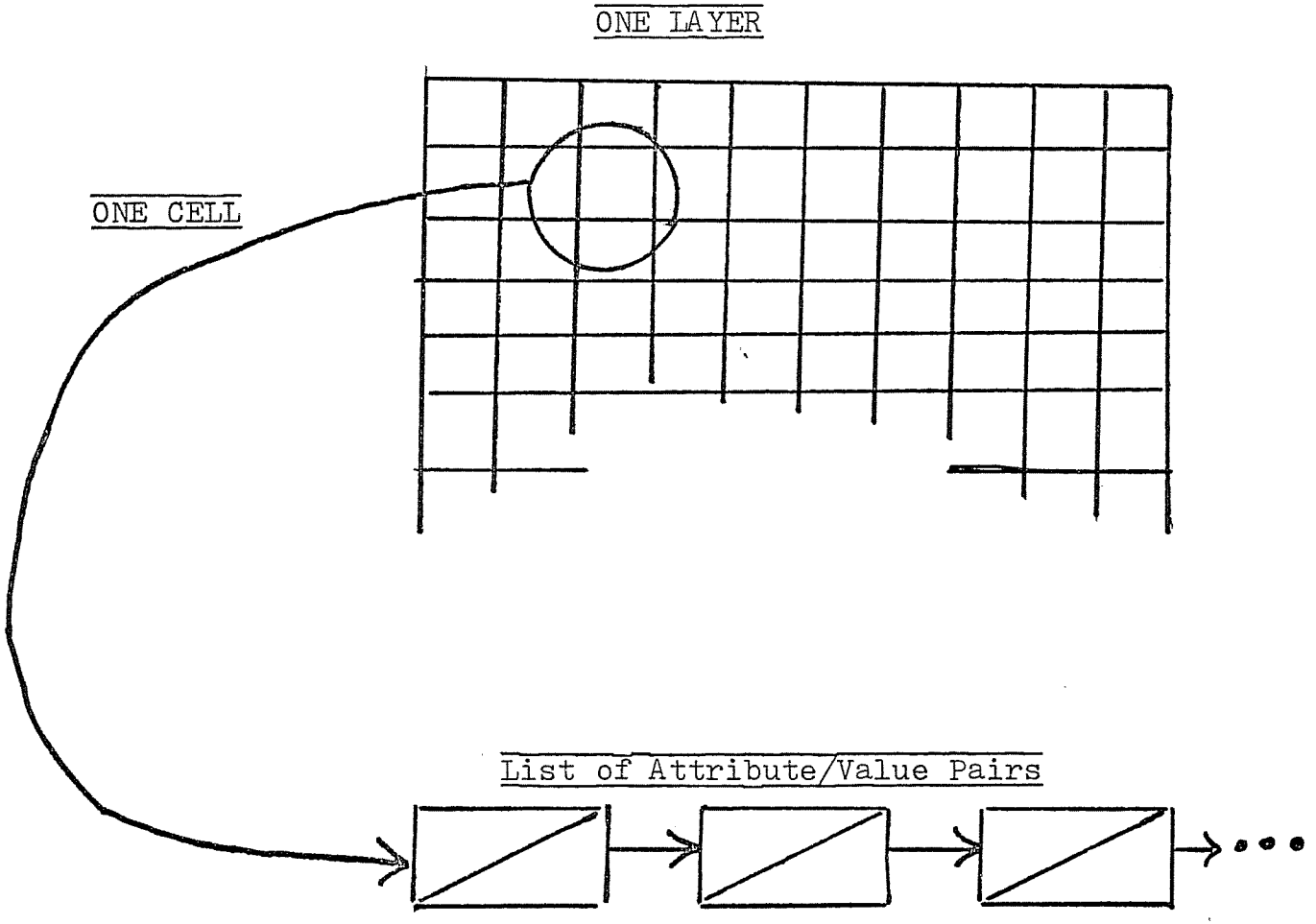
F I G U R E 5 .

THE RECOGNITION CONE.



F I G U R E 6 .

THE MAKE UP OF ONE LAYER OF THE RECOGNITION CONE:



The upper levels of the recognition cone use transforms that look for configurations of lines, angles, colors, and textures. When a transform finds a combination of features matching its input descrition it can imply both object names and further transforms to apply. For example, a transform that finds a red hue and a brick like texture will add the interpretation of "brick wall" for that area to the next layer and also add transforms to check for the outline of a house. At a higher level, a transform might look to see if one area has "sky" as a possible interpretation with a linear edge below it and an area that has "roof" as one interpretation below the edge. If that transform finds such a configuration, it will increase the weight or confidence in those interpretations and possibly add new transforms to search for an area with a "wall" interpretation below the "roof". Thus transforms embody most of the system's knowledge about shapes, surface properties, and contextual constraints on the specific objects that the system can recognize.

The output of the recognition cone for this system is a layer containing for each point in the layer several measures of texture, hue, saturation, brightness, and a set (possibly empty) of interpretations or names. If the highest weighted interpretations are selected and recognition is the primary task to be accomplished, then the system is already a complete scene description program at this point (see Uhr and Douglass, 18). But, if a three-dimensional understanding of the scene is

required then some additional processing is necessary. The remainder of this paper describes segmentation of this output and the construction of a short-term memory world model.

Segmentation of the Scene

The segmentation routine partitions the picture (actually the last layer of the recognition cone) into regions which are similar to one another in multidimensional feature space. The assumption behind segmentation is that each region will represent a distinct object or part of an object. The primary reason for performing segmentation in this system is to reduce the amount of data which must be processed by the short term memory model builder.

A number of quite different segmentation techniques have been developed from histogramming and thresholding used by Ohlander (14) to the interpretation guided segmentation program of Tenenbaum and Barrow (21). Arbib and Riseman (22) provide a relatively complete review of the literature on image segmentation.

Yakimovsky's algorithm (15) has been selected for this system because it combines the advantages of edge detection and region growing and because it is a fast, one pass algorithm. A modified form of the Zobrist-Thompson Gestalt distance function (14) is used instead of Yakimovsky's local edge detector. It provides a simple, adjustable way to combine a number of feature

measures into one edge operator.

The Gestalt distance function, D, referred to as the Zobrist-Thompson grouping operator, is a linear weighted sum of n elementary distance functions, d_i, i = 1,2,...,n. An elementary function measures the likelihood that two points or neighborhoods of points in an image will be visually grouped together based upon the difference in the value of their i^{th} feature. In the present system, an elementary distance function, d_i, represents a scaled difference between values of the i^{th} feature for two points in the recognition cone's output layer. Since the cone is gradually reduced from a 600X800 array to a 50X50 array, each point in the 50X50 layer represents a neighborhood of points in the initial layer. If c_i, i = 1,2,...,n represents a weight expressing the relative importance of the i^{th} feature in grouping two points together, then the strength of an edge element, that is the probability that the points will not be grouped together in the same segment, is D, where:

$D = c_1d_1+c_2d_2+\ldots+c_nd_n$ if $\sum_{i=1}^{n} c_i d_i > t$ (a threshold)

$= 0$ otherwise.

The weights, c_i, in this system are obtained empirically by adjusting weights and threshold and subjectively evaluating the resulting segmentation. The weights can be readjusted over time based upon the world model in short term memory. In this system the first of Yakimovsky's two assumptions behind segmentation, that the image of an object must be approximately uniform or smoothly changing in its local properties (15), is relaxed by using first and second order statis-

tical texture measures in the Zobrist-Thompson grouping operator.

The output from the Zobrist-Thompson grouping operator is formed into two arrays in a suitable form for input to Yakimovsky's algorithm. One array expresses the probability of an edge between point (i,j) and (i,j-1) and the other array the probability of an edge between (i,j) and (i-1,j). These arrays are then input to Yakimovsky's algorithm as modified by Nagel (16). This algorithm defines segment boundaries by searching for "valleys" and "peaks" of edge values. The output is an array expressing the segment that each point in the array belongs to and the boundary points of the segment.

For each segment a description is compiled. This includes mean hue, saturation, texture values, and an indication of the variability and range of these values. Also included are the size of the segment, the names of adjacent segments, the order or number of adjacent segments, and a minimum bounding rectangle for the segment in the output array. A list is made for each segment of all the interpretations for points in the segment along with a total confidence value for those interpretations over the segment as a whole. The segmented picture and the segment descriptions form the input to the world model construction routine.

Figure 7 shows the digitized version of the house scene shown in figure 2. It is averaged down to a 50X50 array in ten gray levels. Figure 8 presents the stronger edge points

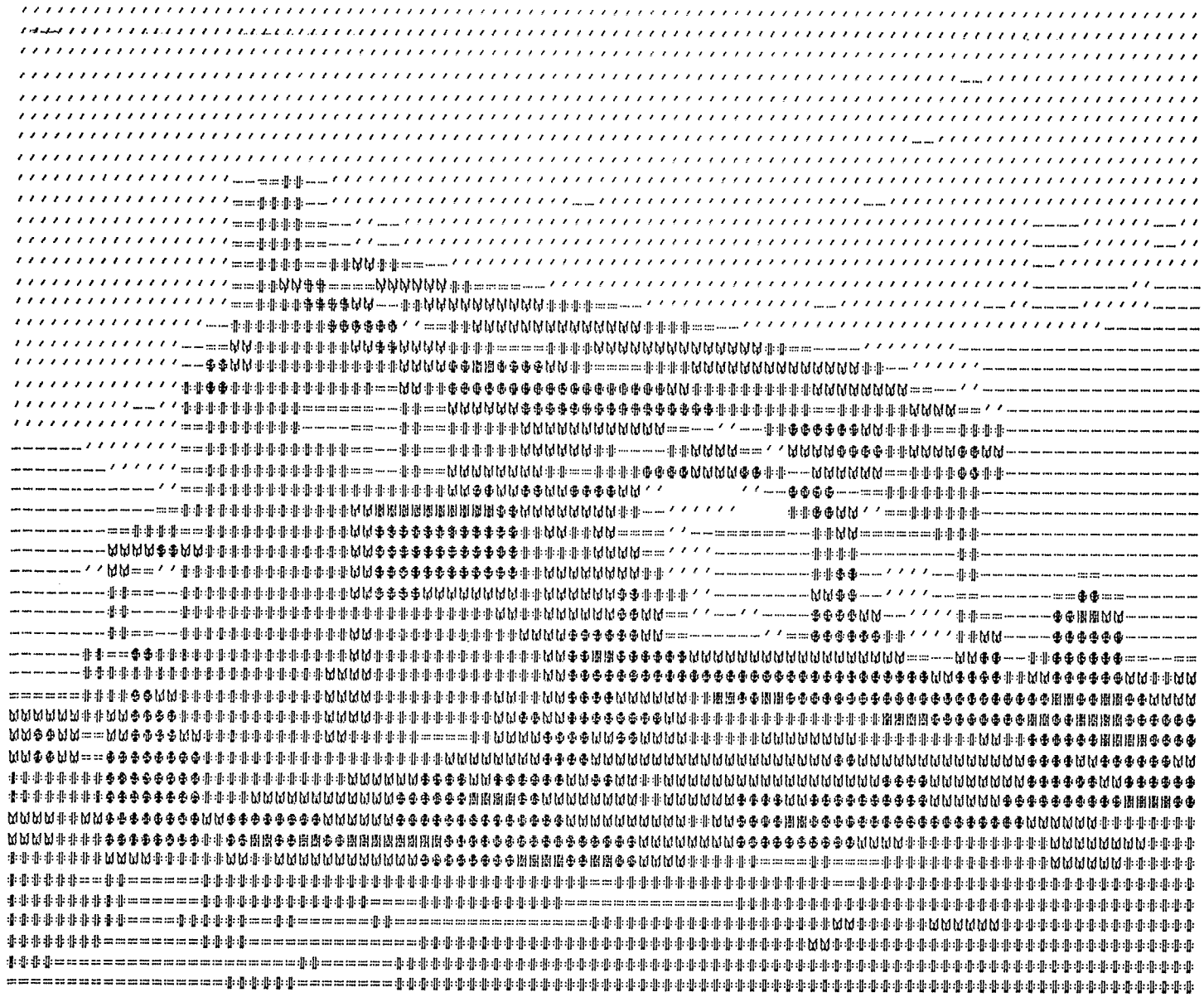


Figure 7 . A 50X50 averaged version of the digitized house scene of figure 2.

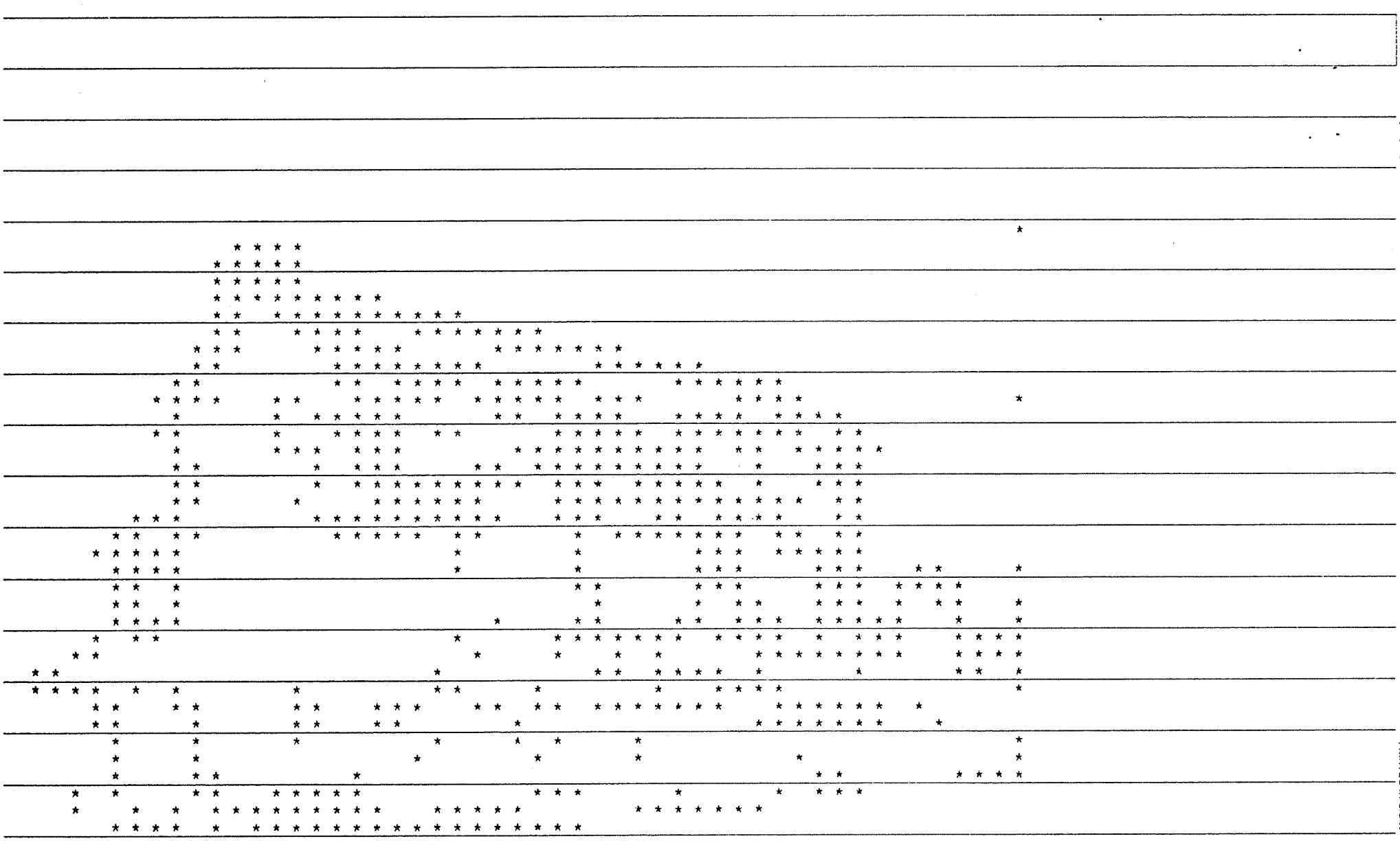


FIGURE 8. The stronger edge points for the house scene produced by the Zoorist-Thompson distance function applied to layer 8. The operator used the following terms: hue, weight 10; saturation, weight 2; intensity, weight 1; texture measure T1, weight 1; texture measure T3, weight 3; threshold ,50.

which result from applying the grouping operator to a 50X50 array of the house scene using hue, saturation, and intensity and two texture measures as terms for the operator.

After the initial segmentation by the algorithm, the output is improved by merging some of the smaller segments into larger neighbors if they are similar in their characteristics. The Zobrist-Thompson operator is used again to decide on their similarity, this time with the segment descriptors as terms for the operator. Merging is conservative in this vision model. Unlike the approach of Tanimoto and Pavlidis (23), which tries to merge out highlights and regions produced by shading gradients, this model views those regions as importan spatial and lighting cues for the construction of the world model. For the house scene, 104 regions were produced by the first pass of the segmentation routine and 76 regions were left after merging. The boundaries of the final segmentation are given in figure 9.

Construction of the Short-Term World Model

The model construction routine merges segments into a three-dimensional model and selects between alternative interpretations for the segments to form a single, internally

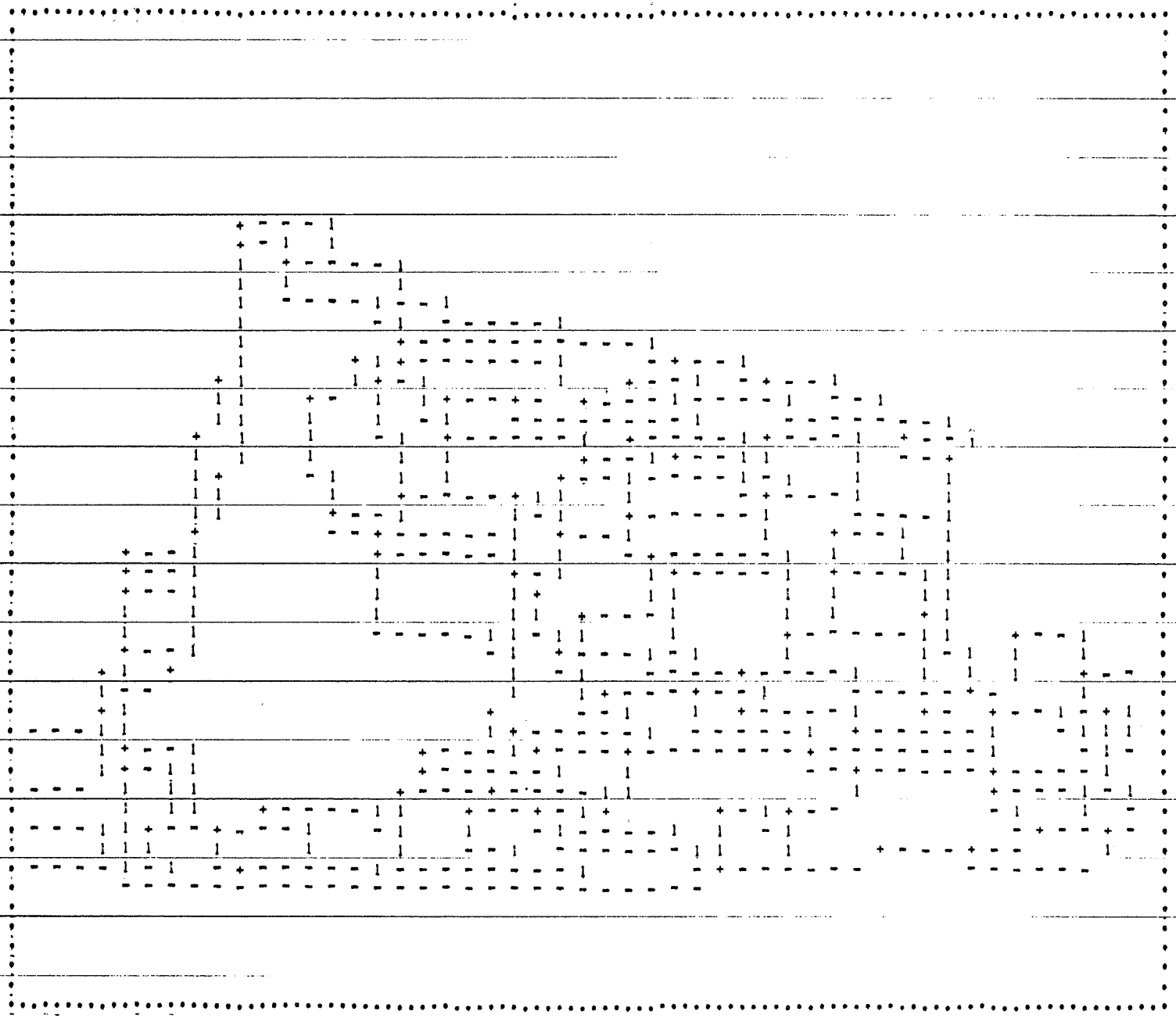


FIGURE 9. The final segmentation for the house scene shown in figure 2 using the edge points of figure 8.

consistent view of the environment as it is visually manifested in a scene. The model forms what would be the output of the system to a high level planner. The world model is built from the segment descriptions, the previous short-term world model if any, the long-term models of specific objects and collections of objects, and general visual heuristics or vision routines concerning aspects of the picture taking process.

A placement algorithm is responsible for initially assigning an object name and a depth to a segment. An adjustment procedure refines the initial depth based upon information from the general visual knowledge routines. An object formation routine decides which segments should be grouped together to form specific instances of an object in the scene. The representation of the world model consists of three parts: a two-dimensional array of picture points, a set of segments or region descriptions, and a set of object descriptions. These parts form a short--term world model because their contents are updated with each new view of the scene that the system gets.

As in the layers of the recognition cone, each point of the two-dimensional array of the world model corresponds to a small area of the digitized scene. Each point of the array contains a value representing a depth for that point and the name of the segment that that point lies in. The region descriptions consist of the segment description formed during segmentation and a depth and orientation for a region. Object

descriptions consist of an object name such as "house" or "tree" and an object number to distinguish between different instances of the same object, such as a scene with several houses. An object description also has a list of the regions which belong to that object.

A segment or region is the basic building block of the world model. It is an area of the scene which is approximately uniform in its visual attributes. It is treated as a two-dimensional surface which can be flat or curved or locally distorted. Each region has a depth assigned to it by the placement routine which expresses the distance of the center of gravity of the region from the system's camera. In addition, each region has a roll, tilt, and pan angle indicating the orientation of the region's surface with respect to the image plane. Finally, each point on the region's surface can be locally distorted from the average depth and orientation of the surface as a whole. These local distortions are expressed as depths in the two-dimensional model array. For example, a crumpled newspaper lying on a desk in front of the system's camera would be spatially represented in the world model by a region with a depth equal to the depth of the center point of the paper plus a tilt angle indicating that the paper was lying flat and by the displacements in depth of each point on the paper due to the fact that it is crinkled and not a flat plane.

The placement routine is responsible for putting the regions produced by the segmentation algorithm into the world

model. The routine places a region in the model by selecting a single interpretation for it, assigning it a depth and spatial orientation, and marking the corresponding area of the two-dimensional array as belonging to that region.

The placement routine selects an interpretation from the set of object names output by the recognition cone. It orders the set and selects the highest weighted name. If there are no names associated with the region or if the confidence weights for all the names are low, then the region will be labeled as having no interpretation. The interpretation given to a region by the placement routine can be reweighted or changed by the visual knowledge routines as explained below.

The placement routine uses one of several heuristics to compute an initial estimate of a region's depth. It starts by examining the information in the long-term object model corresponding to the region's interpretation. If the long-term object model indicated that the object is typically on the ground (as is the case for grass or pavement), then the placement routine will assign a depth to the region based on the ground plane hypothesis (see Duda and Hart, 24). If the object is not on the ground, then the placement routine will attempt to use any available information about the expected size of an object to estimate the region's distance from the camera. If the region is uninterpreted or has no information useful for estimating its depth in its associated long-term model, then the placement routine will assign it a default

depth.

All depth assignments have an associated confidence weight. This weight is initially set low and gradually increased as improvements are made in the initial depth estimate by the adjustment routine. This initial weight reflects the type and quality of information used. For example, an estimate made using the ground plane hypothesis generally has a higher weight than one using the expected size of an object, and the confidence in the expected size of some objects, such as a person, is higher than the confidence in the expected size of a tree.

The placement routine, like the adjustment and visual inference routines, is designed to operate logically in parallel on all regions at the same time. In actual implementation on a serial computer, it places the largest interpreted regions first, using a strategy similar to that of Sakai et al. (7).

After a region is initially placed, the general visual knowledge routines are called to improve upon the depth estimate and possibly correct or strengthen the confidence in the region's interpretation. These routines are a collection of heuristics about occlusion, shadow, perspective and the ground plane hypothesis. They generate hypotheses about a region's depth and orientation with respect to the surrounding regions and store these hypotheses in the form of a relative displacement between two regions and a confidence

weight. An example of the type of inference that these routines make is "if a region labeled tree is surrounded on three sides by a region labeled house, then the tree region is in front of the house region". The content of these routines is described more fully in the section below on visual knowledge routines.

An adjustment routine examines the various hypotheses generated by the visual knowledge routines and resolves them in an improved estimate of a region's depth and orientation. An error is computed for the depth of every point on the boundary of the region being adjusted by comparing the depth of that point with the depth of the points in the neighboring regions and looking at the spatial relationship between the regions as hypothesized by the visual knowledge routines.

The adjustment routine weights the errors by the confidence in the hypotheses it used and the confidence in the depth of the neighboring regions. The weighted error is then used to calculate a new depth and orientation for the region being adjusted so as to minimize the error. Correcting a region's placement causes its confidence weight to be increased. Regions which have a confidence above 80 per cent will not be adjusted.

Adjusting a region's position in turn indicates corrections in surrounding regions to be made by the adjustment routine. This pattern of successive corrections by the adjustment routine causes the routine to be cycled repeatedly

over all the regions. The cycling using confidence weights has the effect of propagating reliable information about the depth of one region to other regions. For example, if the system knows with some certainty that a region labeled as a tree lies about 100 feet from the camera and the occlusion heuristics indicate that the region is in front of a second region labeled as a house then the tree's depth can be used to set a minimum bound on the distance of the house. Improving the estimate of the house's depth can, in turn, be used to improve estimates for the depth of any regions known to be behind the house such as other trees.

The cycling of the adjustment routine stops when a stable interpretation of the scene has been achieved. A stable interpretation is one in which all the regions in the world model either have a placement confidence above 80 per cent or have a depth and orientation which is in agreement with the depth and orientation of the surrounding regions. The spread of information between the regions by a repeated application of of an adjustment routine is similar to the relaxation techniques used by Tanenbaum and Barrow (21) and Rosenfeld et al. (25) to constrain the interpretations of a scene or match a template to a portion of an image.

Figure 10 shows a depth plot of the house scene of figure 2 after initial placement of all the regions but before an adjustment. It represents the distance of each point in the

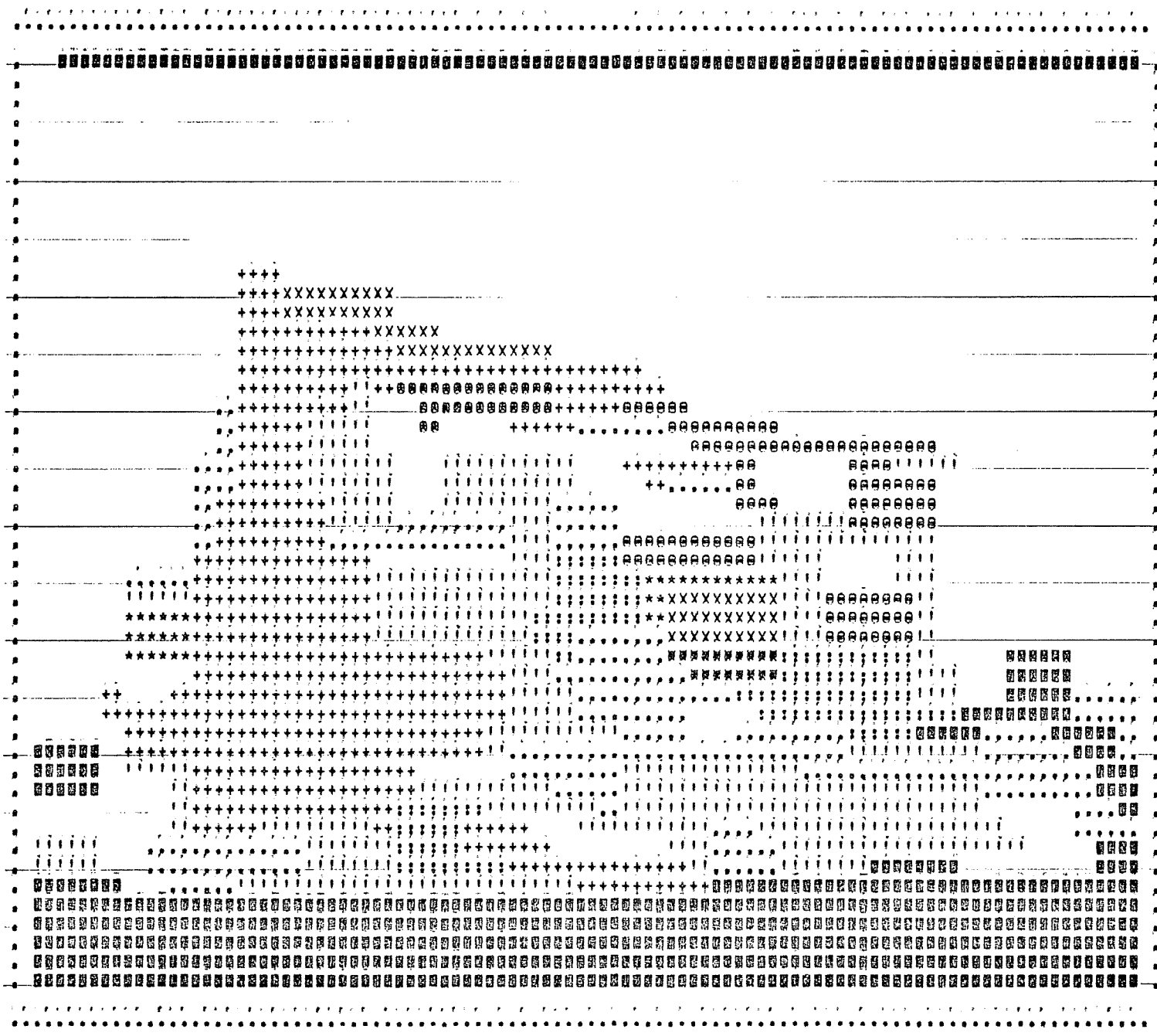


FIGURE 10. A depth plot generated from the world model for the house scene of figure 2. In this plot all regions have been placed into the world model but there has been no adjustment of the regions. The darker the shade of a point the closer it is to the camera.

scene from the camera by a scale of gray where the darker the point, the closer it is to the camera. Figure 11 plots the depth of the scene after one cycle of adjustment for all regions. Notice how just one update has registered many of the regions of the house and trees. The perceived depth of the scene after five cycles of the adjustment routine is shown in figure 12. The region boundaries produced by the segmentation algorithm are drawn in on the depth plot to indicate how the location of regions in depth changes.

The construction of the short term world model is completed by designating regions as parts of particular objects. Assigning a region to an object in the object list should not be confused with assigning an object interpretation to a region. For example, the placement routine using information from the recognition cone on shape, texture, etc., might assign the interpretation of "house" to a region which forms the roof of a house in a scene. But an object formation routine must decide what regions should be taken together with the roof region to form one particular instance of a house in the scene.

The object formation routine assigns regions to objects in the following way. For each region it checks to see if the region has any neighboring regions with the same interpretation. If there are such regions, then the region under consideration is assigned to the same object as its neighbors. If no neighbor shares a common interpretation, the object formation routine looks at the world model to see if the region might

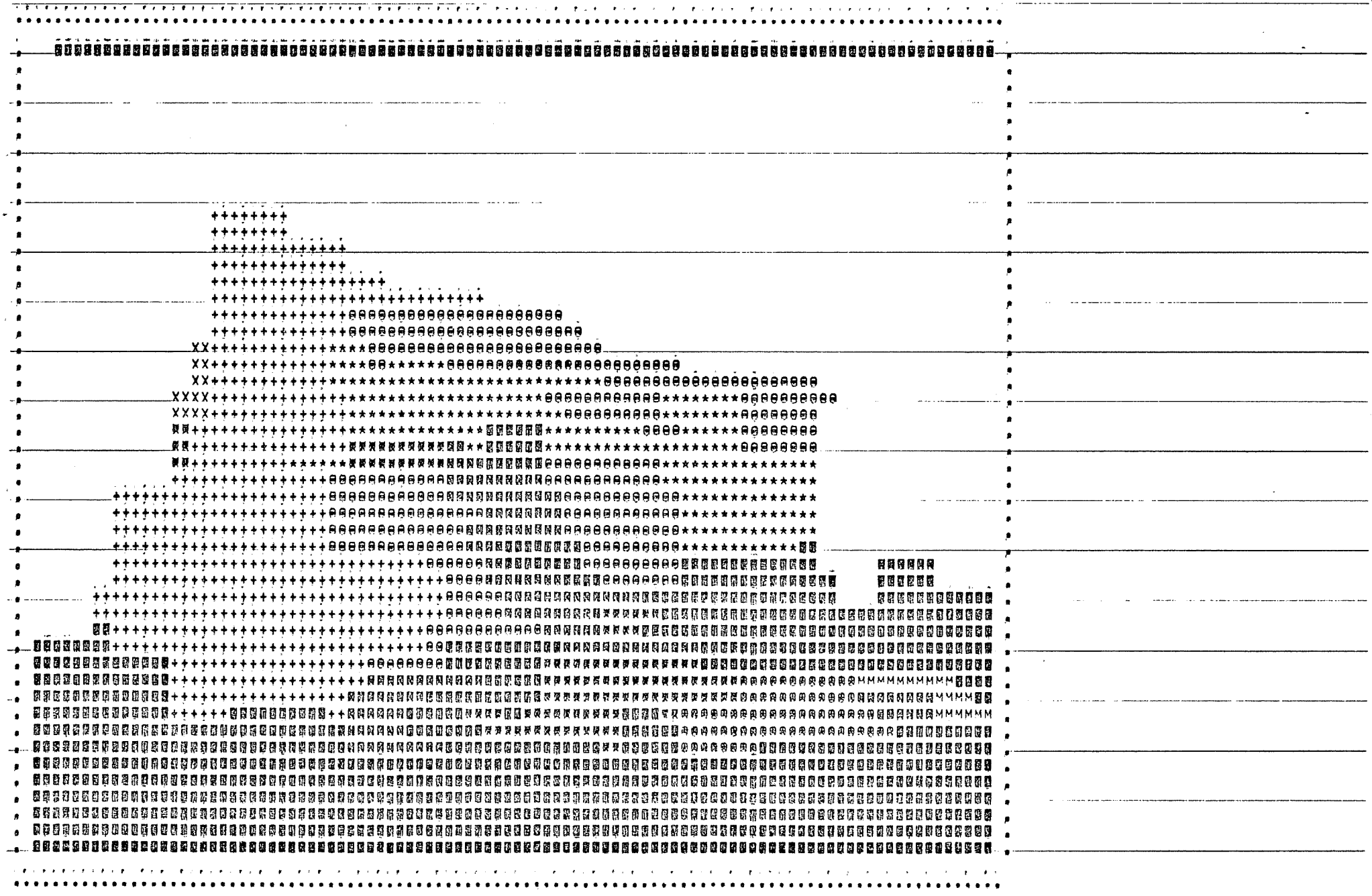


FIGURE 11. A second depth plot generated from the world model after all regions have been adjusted once.

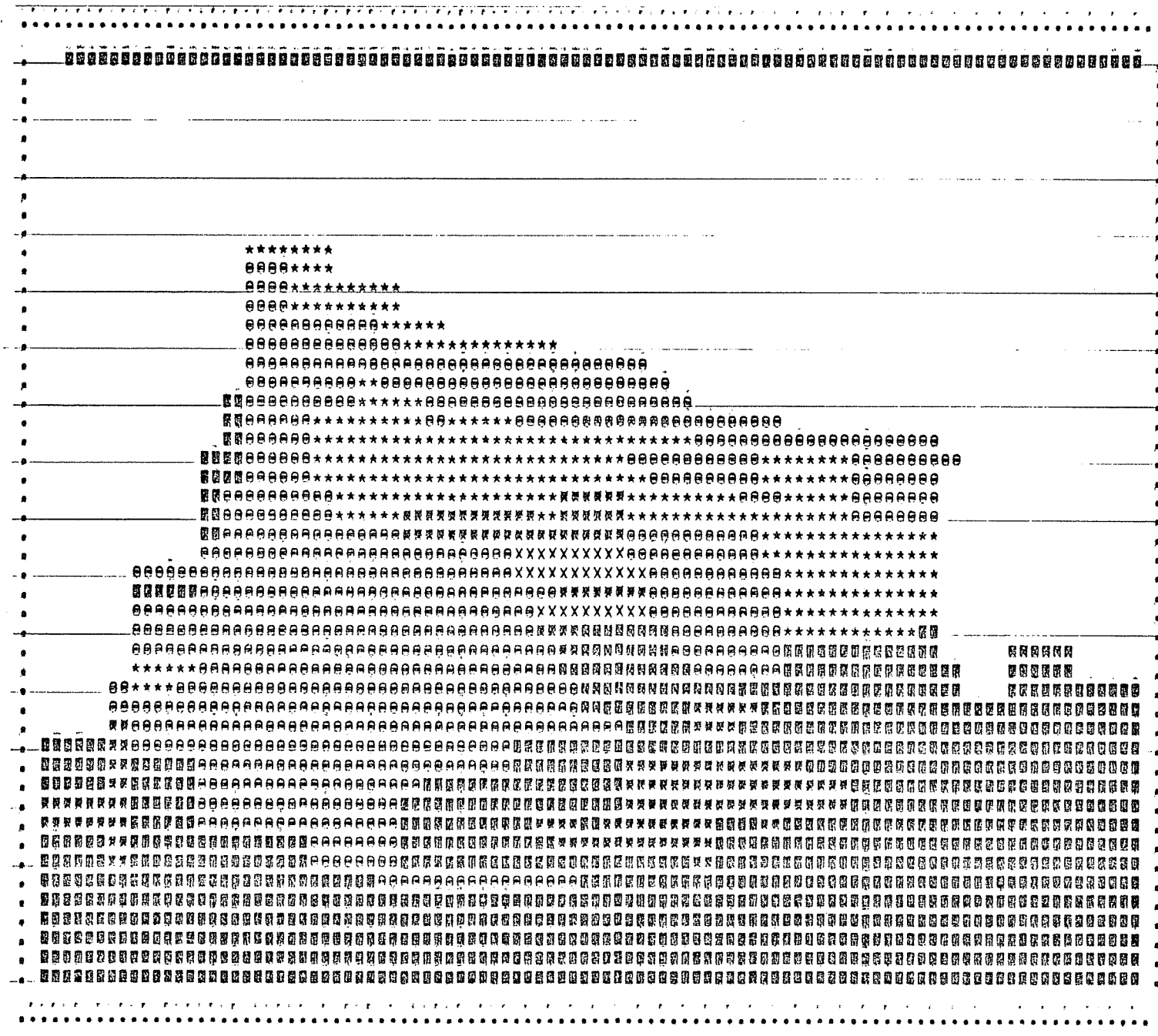


FIGURE 12. Depth plot for the house scene after five cycles of the adjustment routine over all regions in the world model.

belong to an object which is occluded by another object in front of it. If the region still can not be grouped with other regions to form a single object, then a new instance of the object is formed. Separate instances of an object may later be merged into a single instance as the placement routine places more regions into the world model and as the visual knowledge and adjustment routines produce a better understanding of spatial relationships in the scene.

Figure 13 shows a simple stylized example of the problem. Here a tree in the foreground divides a house into several different regions. By using texture or shape cues it is possible to determine the presence of a house and a tree but only by developing a spatial understanding of the scene can the program determine that all the house regions belong to the same house with a tree in front rather than to two separate houses with a view of the background between them. The next section covers some of the specific heuristics used to obtain this type of scene understanding and their organization.

General Visual Knowledge Routines

The general visual knowledge routines contain the knowledge used to interpret a scene in three-dimensions. The types of information they use correspond roughly to the major monocular depth or perspective cues identified by psychologists

Figure 13. A simple example of one instance of an object, the house, divided into separate parts by an occluding object in the foreground.

(26). The cues used by the present program include texture gradients, shadows and highlights, expected size, ground plane, occlusion or interposition, linear perspective, and motion parallax. These routines are best thought of as general knowledge because they contain information which is largely independent of the specific objects which might be in a scene. These routines process regions as they are merged into the short-term world model using as data the region descriptions and the long-term object models.

The program's general visual knowledge consists of sets of heuristics. Each set is grouped under the heading of one of the depth cues and is associated with a routine. The routine applies its heuristics to the regions as they are being placed into the world model. All routines operate logically in parallel and independently of one another. Normally, the routines hypothesize a spatial relationship between a region and one of its neighbors and compute a weight expressing the routine's confidence in the correctness of that hypothesis. They can also change the object interpretation of a region or raise or lower the confidence weight in an interpretation. The hypotheses about a region's orientation are all represented in a common form and associated with the region's description. The adjustment routine interprets the hypotheses to obtain an improved estimate of the region's depth as explained above.

Of the depth cues included in this vision model,

occlusion is the most completely developed. Occlusion, usually referred to as interposition in the psychology literature, (Massaro, 26) occurs when one object is positioned in front of another object in such a way that the projection of the object in front onto the image covers up part of the image of the object behind. Except for scenes of polyhedra, occlusion has been used very little as a depth cue. In scenes of straight-polyhedra, it is possible to unravel overlapping objects by categorizing vertices into several types as Waltz's program did (27). Ohlander (12) is one of the very few who has developed heuristics for resolving occlusion problems in natural scenes. His approach compares the characteristics of regions as opposed to the vertices.

The occlusion routine in this program uses about 30 different heuristics to decide if a region is overlapping, is overlapped, or is touching another region. There are two general categories of heuristics, semantic and nonsemantic. The nonsemantic ones include a number of questions about the positioning and characteristics of one region with respect to its neighbors. These heuristics are an extension of Ohlander's proposed techniques. An example of a nonsemantic heuristic would be the rule that if one region is surrounded on three sides by another region, then the surrounded region occludes the other region. Many of the positional relationships between regions are calculated using a minimum bounding rectangle around a region. Implications of occlusion for a

region are weighted to reflect a confidence in the heuristic.

Semantic heuristics involve a region's object interpretation. They can be divided into special shape transforms and expected object relationships as expressed in the long-term object models. An example of an expected relationship is that a region labeled as a roof would be expected to touch a region labeled as a wall rather than occlude it or be occluded by it. Another example would be the long-term model for a window which would indicate that a window region lies in the same plane as a surrounding wall region.

Shape transforms are transforms in the recognition cone which match part of the outline of an object and imply that object's name. For example, certain transforms match roof outlines to detect houses and other transforms match the outline of trees. When these transforms succeed in matching, they indicate that the object they match lies in front of the bordering objects. This information on occlusion is passed along to region descriptions by the segmentation algorithm.

Shape transforms represent one solution to the figure-ground problem of perceptual psychology. A region is perceived as figure when the boundary between that region and its neighbors is perceived to belong to the contour of that region. When the contour of a region is perceived, it means that it must lie closer to the viewer than the neighboring occluded regions. Where a contour can be perceived as

belonging to either of two adjacent regions in an image then an ambiguous interpretation can result. Where other depth cues are absent or are also ambiguous, then figure-ground reversals occur. There are many examples of such reversing images. One of the best known is the faces-vase picture by Edgar Rubin. The presentation of a similar ambiguous figure to the vision model described here would result in conflicting hypotheses as to which regions were occluding each other and could lead to oscillation in the depth estimates assigned to the regions during the cycling of the adjustment routine.

The visual knowledge routines for inferring depth from expected object size and from the ground plane hypothesis are quite simple compared to the occlusion routine. The expected size routine uses the one heuristic: that the area of an object's image is proportional to its actual size and distance from the camera. By retrieving the expected size of an object from the long-term model of that object, the routine can estimate an object's depth. The size can be predicted with much more certainty for some objects than for others and this degree of certainty is expressed in a weight.

Using the hypothesis that the ground is a plane, the ground plane routine can calculate the depth of points in the image using an approximate model of the camera's position and orientation (see Duda and Hart, 24). The ground plane hypothesis has been used successfully and frequently in in-

door robot vision programs. The potential for hills and dips in near views of outdoor scenes makes the hypothesis much less reliable but still a useful cue.

The texture gradient routine estimates a region's orientation by using simple measurements of the size and coarseness of texture elements computed by the recognition cone. Horn (28) proposed the use of texture gradients, in a manner similar to shading gradients, to derive the precise shape of object surfaces. Texture gradients have been used, in a program by Bajcsy and Leiberman (29), to estimate object depth, in particular the surface of the ground. The routine for this present system uses texture gradients in a more heuristic and less precise way. The program measures changes in texture coarseness between different windows over a region. Although texture gradients could be used to determine point by point changes in surface curvature for a region, the present routine uses only average changes in texture coarseness over a whole region. These changes are interpreted in terms of a tilt and pan for the region with respect to the image plane. The long-term object models help resolve ambiguities in interpreting the average gradients. For example, the model for a brick wall indicates that it is usually perpendicular to the ground and therefore the gradient routine will tend to interpret a texture gradient for a region labeled as a brick wall only in terms of a pan angle for the region away from the image plane.

Shadows and highlights are detected by comparing the descriptions of neighboring regions. Two regions are rated for their similarity using the Zobrist-Thompson distance function on the region descriptions. The function is computed twice, once with average light intensity included as one of the terms of the function and once without it. If the two regions differ in light intensity but are similar in other properties, then they will be judged as being shadowed (or highlighted) and unshadowed (or not highlighted) portions of the same region. Special weight is given to the texture descriptors since saturation and hue values can change in complex ways when a region is shadowed.

The object interpretations of two regions are also compared for similarity. If the regions share a common object name, then changes in their characteristics are more likely to be interpreted as shadowed and unshadowed parts of the same object. Conversely, two regions with different object names may be reinterpreted if the shadow routine finds that they are very similar except in light intensity.

The shadow routine makes no attempt to infer what objects are casting shadows. Extending the shadow routine to make such inferences from the three-dimensional world model and hypotheses about the direction of the lighting source would greatly enchance the program's ability to detect shadows and would provide additional depth information.

Linear perspective is the final depth cue used by

the system. It has been used by artists in paintings since the Renaissance to convey a sense of space in two-dimensional pictures. It involves a set of assumptions about the way objects and surfaces are foreshortened in an image as they slant away from a viewer. In particular, lines that are parallel in a scene and recede from a viewer appear to meet at a distant vanishing point in an image.

Although the equations of linear perspective are well understood (24), it is still quite difficult for a computer vision system to detect it in complex outdoor scenes with partially occluded objects, irregular region boundaries, and segmentation errors. Furthermore, the interpretation of the slopes of lines in an image, even when they can be recognized as approaching a single point, requires object specific knowledge on the part of the viewer, human or machine. The painting, Les Promenades d'Euclide by René Magritte, provides an example of the power and pitfalls of linear perspective. The artist depicts a roof with a conical steeple next to a view of a Parisian boulevard receding into the distance. The two images are juxtaposed in such a way that the foreshortening of the street, as its sides slant toward a vanishing point on the horizon, produces exactly the same outline as the pointed sides of the steeple. The sloping sides of the street give a strong impression of the street lying on the ground and running off in the distance. For the steeple, exactly the same configuration of lines helps convey the feeling of a conical

object rising perpendicular to the ground.

The linear perspective routine copes with some ambiguities by consulting information in the world model and information about specific objects in the long-term object models. The routine starts by trying to measure the foreshortening of a region in two directions. It calculates the height and width across a region by scanning the region from top to bottom and left to right. Next, the average derivative or change in the height and in the width of the region is computed. The averages are used as heuristics for estimating the foreshortening in a region along two axes. When calculating the averages, the routine disregards points where the world model indicates that a region's boundary is occluded by another region. To express the foreshortening of a region as a tilt or pan of the region away from the camera, the routine must consult the long-term object model corresponding to the region's interpretation. In the case of a brick wall, the long-term model indicates with weights that the wall frequently has parallel sides and is usually perpendicualr to the ground. A narrowing in a wall region along the horizontal axis of the image plane would then be interpreted as a pan away from the image plane.

Long-Term Object Models

A long-term object model describes the general char-

acteristics and properties shared by many different instances of a particular object. The main purpose of long-term models is to aid in the construction of the short-term world model by providing information on the spatial organization of objects and providing contextual constraints which can reduce the possible interpretations for a region. The models contain the system's knowledge of specific objects, which is used by the visual knowledge routines, placement routine, and the object formation routine. The models are called "long-term" to distinguish them from the short-term object descriptions in the world model which represent specific instances of objects in view in a particular scene.

The object models are probably most similar in flavor to the models of Sakai et al. (7), with two important differences. One, these models place much more emphasis and detail on describing the spatial relationships between the parts of an object and other objects. Two, Sakai et al. match their models against region descriptions to recognize objects in the scene. In this system, identification of objects is done by the transforms in the recognition cone and not by matching the long-term models to region descriptions.

The present system's models are implemented as instances of SIMULA classes so that they all have a common form. The models contain average values for surface texture and hue, values for expected object size and variations in size, the name and relationship of any simpler objects that form the

object, and contexts that the object is likely to be found in, along with the orientation of the object with respect to other objects in that context. The models also indicate for each object any larger or more complex object that this object is a part of, and special descriptors which indicate, for example, whether or not the object sits on the ground, has parallel sides, or is likely to consist of several distinct regions. The model for an object also contains pointers to transforms in the recognition cone that imply the presence of that object in a scene.

The long-term model for a brick wall that has been given to the program will be described to illustrate a typical model. A wall has an expected size but a low confidence weight since brick walls vary widely in size. A typical texture and hue are specified for red bricks at a standard reference distance. The model indicates that a wall does consist of bricks as subparts and it can itself form a part of the object house. Expected contexts or associations for a wall include windows which are specified to be surrounded by the wall and to lie in the same plane as the wall. Other walls can also touch a wall, almost always at right angles. A wall is specified to be found near trees and grass and to touch the ground, usually forming the vertical leg of a right triangle. The wall model designates a wall as being recognizable at the region level, meaning that it can consist of a single region of fairly uniform properties, as opposed to an object like a house which consists

F I G U R E 14.

THE CONTENTS OF A LONG TERM MODEL

1.) Size, variation in size.

2.) Parts, example: walls and roof for a house.

3.) Relationships between parts.

Specified as angles of contact between parts, and orientation of each part with respect to ajoining parts.

4.) Expected contexts for an object or any larger objects that this object is a part of.

5.) Orientation of the object with respect to other objects in a context expressed as an approximate roll, tilt, and pan with respect to the object.

6.) Expected surface properties: color, texture, etc. (if applicable)

7.) Special predicates (see text): "on the ground", "vertical to ground", "parallel sides", "segment level", planar, etc.

8.) Backlinks to the transforms that imply this object.

of several different looking parts. Finally, the model specifies that a wall is a plane and usually has parallel sides.

Most of the information in any object model is weighted with a confidence value. The weight lets routines such as the visual knowledge routines rate the probability that their hypotheses are correct. Spatial orientations, like the relationship between a wall and a window, are given by a set of standard parameters which indicate probable angles of contact between objects and whether one object is usually above or below or beside another object.

The major parts of a long-term object model are summarized in figure 14.

Processing Scenes Over Time

This system is designed to merge successive views of a scene over time to enhance the accuracy of the world model and reduce processing time. For an industrial robot that must navigate through and manipulate objects in its environment, integrating a sequence of views over time is important for several reasons. It maintains a continuity from one view to the next while providing information to direct the analysis of each new input image in terms of the last. It increases the system's confidence in the object interpretations and the world model by providing new perspectives on

objects and permitting monocular motion parallax to be used for depth calculations. Analyzing scenes over time is crucial if any objects actually move in the scene. Finally, a vision system that can process several views of a scene provides a robot planner top down control over what is perceived by controlling what views of a scene the vision system gets.

This system uses the information from a previous view in several ways to assist in processing the current view. It can influence lower level processing by using the objects that were found in the previous view to select additional transforms to be used in the recognition cone. It can lower the threshold and reweight the terms of the Zobrist-Thompson operator to get a coarser or finer segmentation. It uses the region descriptions from the last view to match up corresponding areas of the current view. Paired regions from the two views are used to select an interpretation for the new region, assign it to an object, and to get an additional depth estimate from the parallax in the region's position in the two views. The pairing of regions and the motion parallax calculations are designed but not implemented at this time.

Region matching is performed by a visual inference routine for motion parallax. The routine assumes that different views of a scene are relatively close together in time and viewing position. Even so, most regions can be expected to change in position and size from one image to the next.

This characteristic makes the problem of motion parallax more similar to the problem of registering stereoscopic images made at the same moment of time than to the problem of detecting a small number of moving objects from several views of a scene taken from the same camera position. Nevertheless, this system's parallax routine matches region boundaries in a manner more reminiscent of the movement detection work of Potter (30), Aggarwald and Duda (31), and Nagel (32) than of the point for point cross correlation approach of such stereovision programs as Hannah's (33).

The motion parallax routine is called after the placement routine has made an initial placement of a region into the short-term world model. The parallax routine attempts to match the boundary of the region with the boundary of the corresponding region in the previous scene. It reduces matching expense by first selecting a small set of candidate regions from the previous scene, matdhing a region against these candidates with a simple comparison function based on the regions' descriptions, and using the depth information in the world model generated from the previous view.

Before any matching occurs, the world model built up from the previous views is transformed as follows to correspond approximately to its appearance in the next input image. The transformation is made by moving segments according to their depth and orientation and the parameters des-

cribing how the camera will be moved before the next view. New region descriptions are computed to complete the update of the world model. The model is then saved and a new image is input and segmented. The regions from the new segmentation can now be matched against the old model to find their corresponding regions in the previous image. Any discrepancies between the predicted location of a region and its actual position in the new segmentation can be attributed either to inaccuracy in the model's estimate of depth and orientation for the region or to actual movement of the object that contains the region.

Each region from the current view of the scene is matched in turn against a set of candidate regions from the previous image. A region from the current view of the scene that is being matched is referred to here as the target region. The set of candidates contains regions in the same general area of the image as the target region and also regions of similar size and intensity from elsewhere in the previous image. A comparison function measures the similarity between the target region and the candidate regions and the most similar match is selected. The comparison function is a Zobrist-Thompson operator which forms the sum of the weighted differences between the two regions' properties. The properties used are textual measures, hue, saturation, intensity, order of connectivity or the number of different neighboring regions,

object interpretation, and region size. Target regions which are not rated by the Zobrist-Thompson operator to be similar to any of the candidates are left unmatched. Conflicts between two target regions matching on the same candidate region are resolved by selecting the closest match.

Once regions from the old and new views have been matched, the motion parallax routine attempts to match the boundaries of the pairs of regions to get a new estimate for a region's depth and orientation. Normally, the depth estimate from the parallax routine is treated like the hypothesis of any other visual knowledge routine. These hypotheses are combined with other estimates by the adjustment routine. But a large discrepancy between the depth of a region indicated by motion parallax and that predicted from the other visual knowledge routines in the old world model may be interpreted as object motion. If the long-term model associated with the object interpretation of the region indicates that it is a movable object, then the discrepancy is attributed to object motion. If the long-term model indicates that a region is unlikely to be moving, then the discrepancy is resolved by either discounting the match as a bad match and disregarding it as a depth cue or by accepting the depth estimate from the old world model, depending on the similarity rating for the match and the confidence weight for the region in the old world model.

Results

The vision model described in the preceding section, with the exception of the motion parallax routine, has been implemented in SIMULA and is running on a UNIVAC 1110 computer. The program was given long-term object models and transforms for a number of objects including tree, house, grass, sky, car, street, window, brick wall, concrete wall, roof, and ground. Figures 3 and 4 and figures 6 through 13 show the program's output at various stages of processing for figure 2 as the input scene. The performance of the recognition cone on figure 2 and on other data is reported elsewhere (18).

Figure 15 shows a picture of the distinct objects found in the scene, where each individual instance of an object is represented by a different uniform shade of gray. Figures 16 and 17 give the major group of trees fringing the house and the major region of grass respectively. All of the regions of the house which were correctly identified as belonging to a house were also correctly merged into one house as shown in figure 3. Using the world model and its occlusion heuristics, the program correctly inferred that the wall which extends in front of the house was part of the house even though the wall is totally isolated from the rest of the house by trees. A few small portions of the roof of the house were incorrectly identified and therefore were not incorporated into

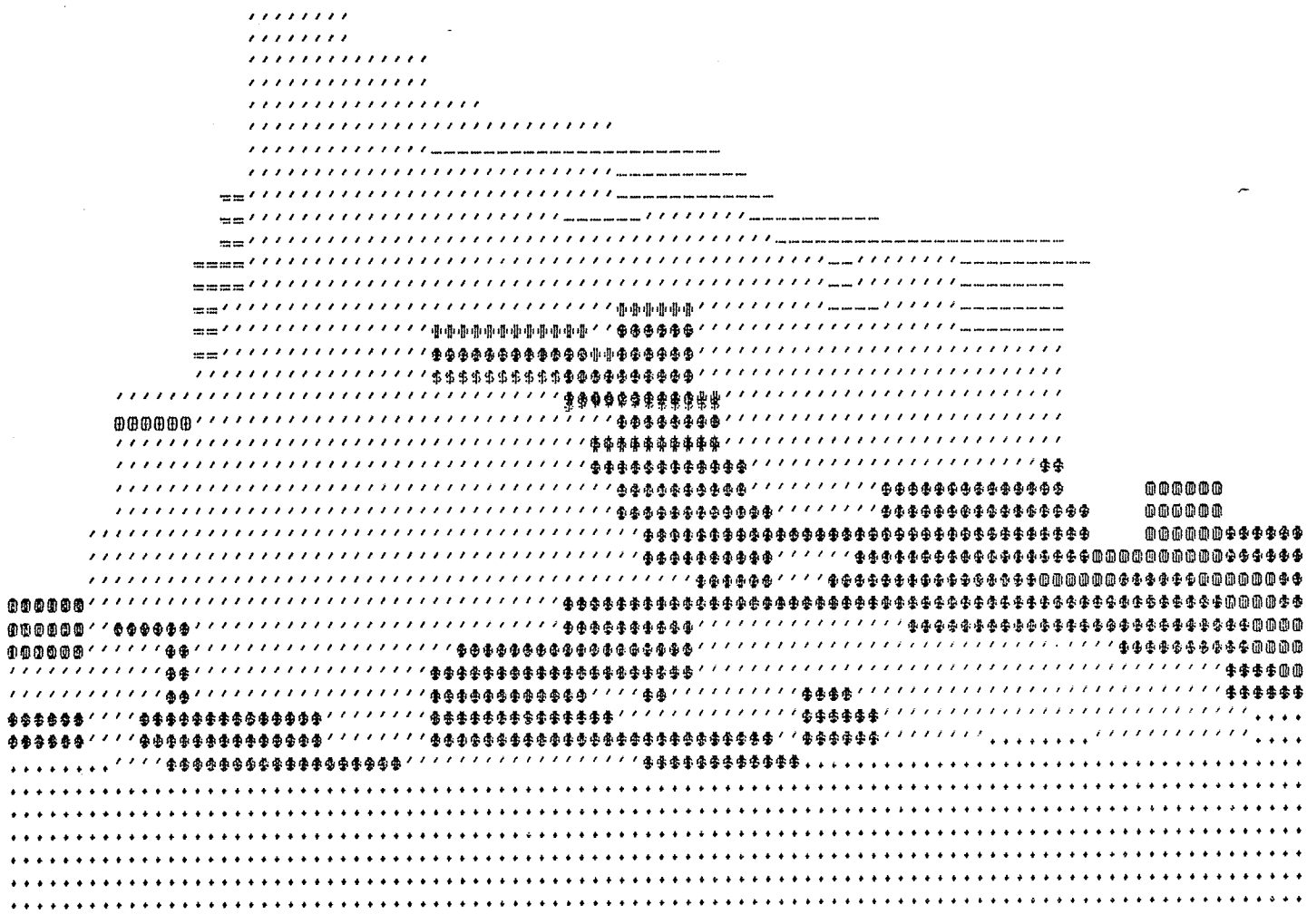
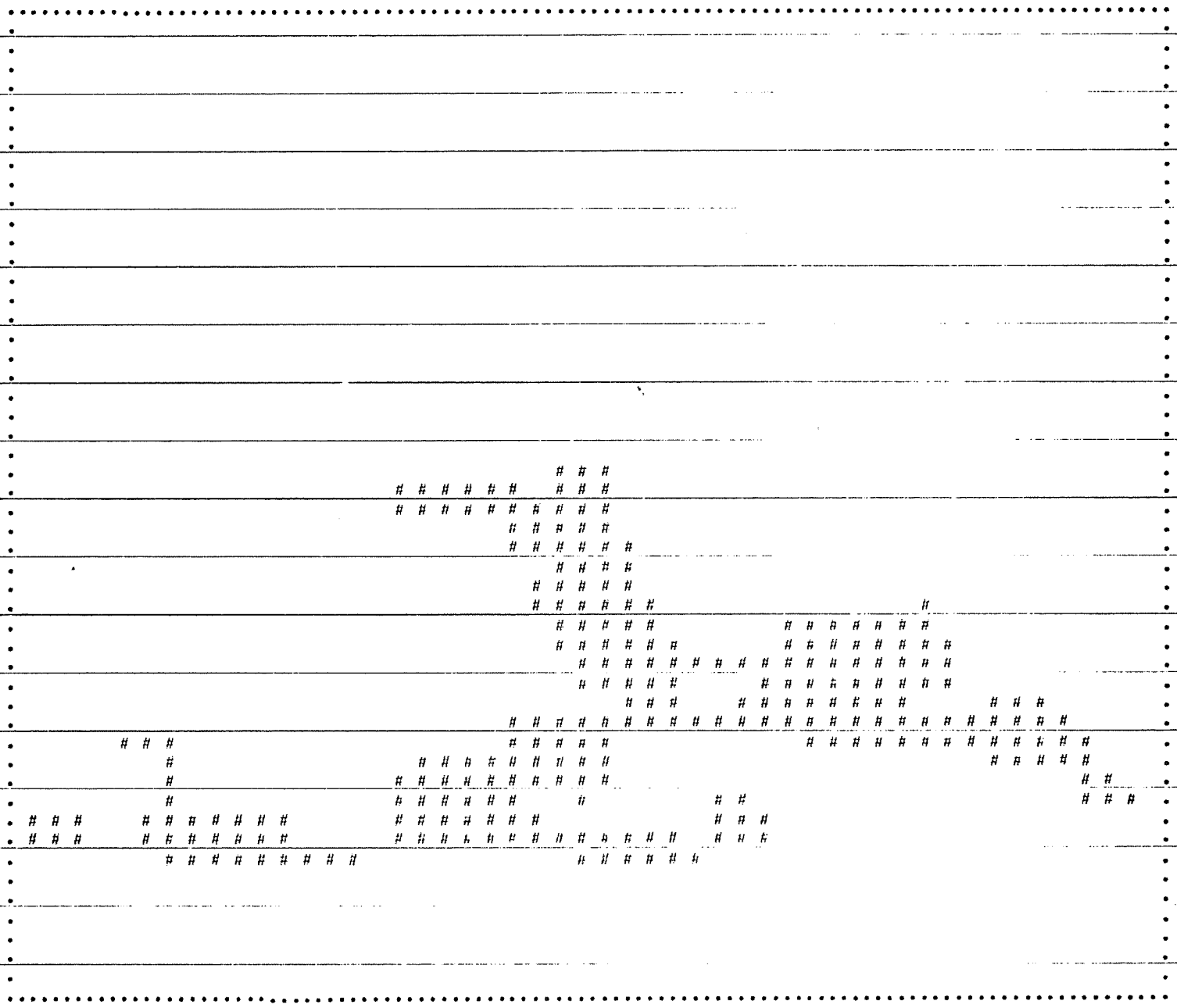


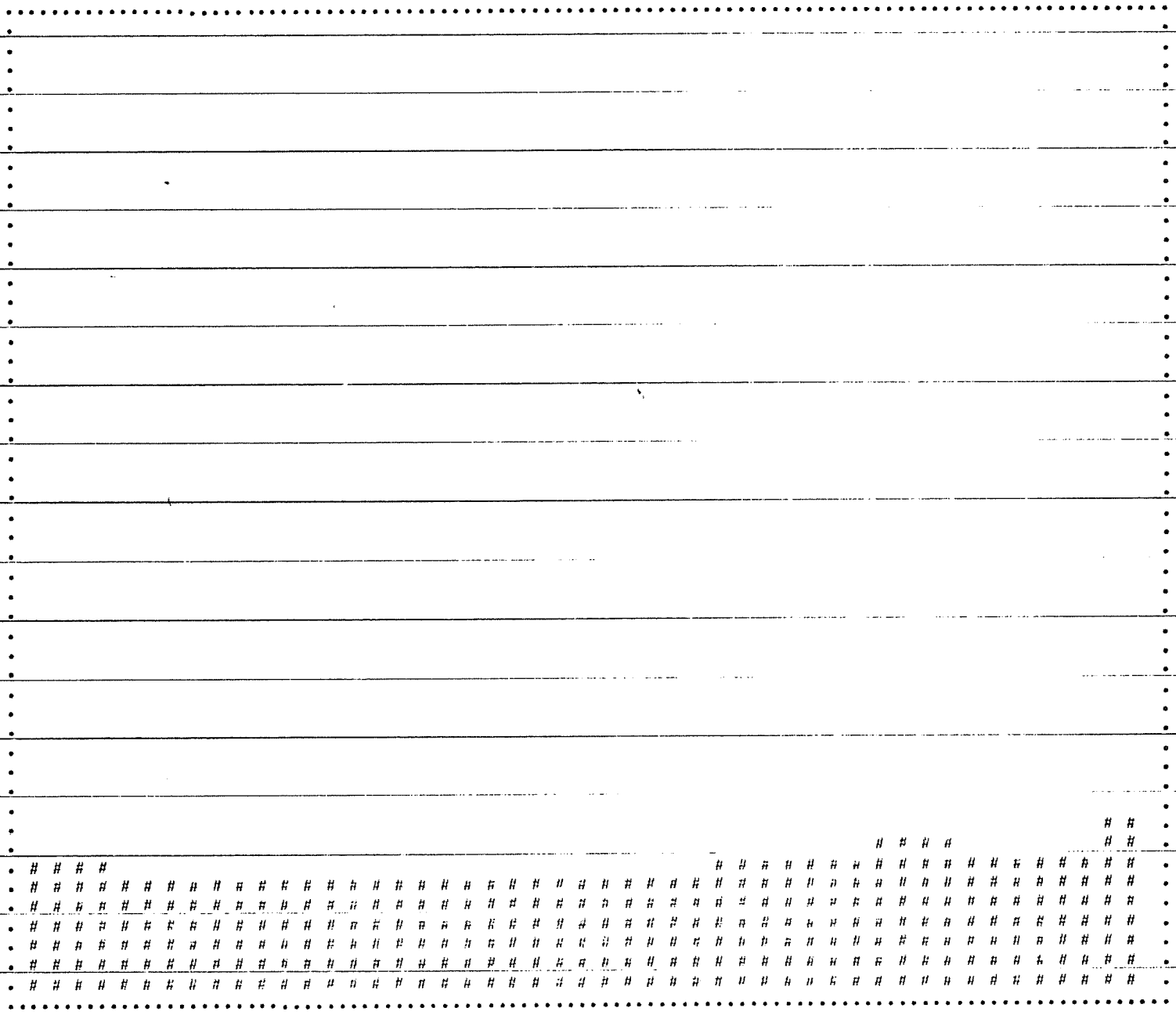
FIGURE 15. The distinct objects found in the house scene shown in figure 2. All objects share a common shade of gray.

THE SEGMENTS BELONGING TO TREE 25 FOR OBJECT 12

103 97 74 95 96 94 92 87 86 83 81 78 76 52 54 60 59 51 38 31 32 29 62 67 72 73 69



THE SEGMENTS BELONGING TO GRASS 28 FOR OBJECT 3
101



the house in the short-term world model.

The depth plot generated from the world model shown in figure 3 represents the distance of each point in the scene from the camera by a grayscale value. The lighter the point the farther it is from the camera. The boundaries of the objects found by the program are outlined on the depth plot. Figure 18 is the same as figure 3 but it gives a scaled value representing the depth of each point in the scene.

Figures 3 and 18 indicate that the program successfully determined the general spatial organization of the scene. The world model that was formed indicates that the grass recedes from the camera in the foreground and that the trees and house rise vertically from the ground. The wall in front of the house does extend in the model out away from the house and the large tree near to where the wall meets the house is represented as standing in front of the house. Most of the complex structure of the house was missed by the program. The house was treated as if it were viewed front on. The program did not find the front first floor extension on the house and did not detect the boundary between the end of the house which slants away to the left and the front of the house which slants away to the right. These errors are due primarily to the failure of the texture operators to detect changes in the brick or tree texture. After averaging in the cone, the resolution was too coarse to detect texture gradients of that scale.

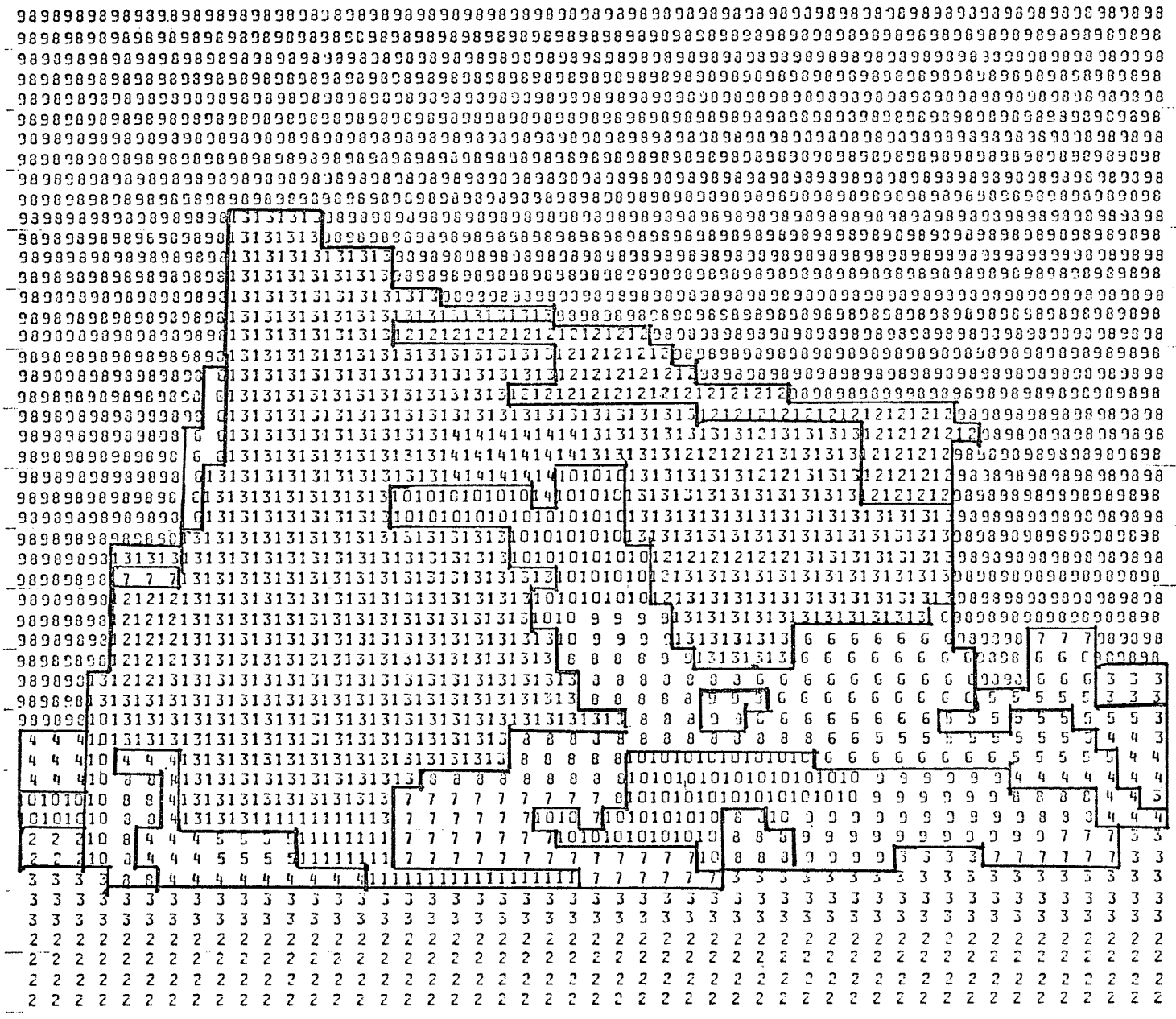


Figure 18. A depth plot produced from the world model with the depth of each point represented with two digits.

Figure 19 gives a list of CPU execution times for various parts of the program on the house scene of figure 2. The total execution time for an entire run of the program from a 600X800 image to the output of the world model is approximately 4 minutes, 13 seconds with an additional 1 minute, 45 seconds spent by SIMULA in garbage collection for a total CPU time of about 6 minutes for an entire run. The total pool of storage available to the program is 56k of 36 bit words.

Discussion

Three aspects of the computer vision model presented here should be emphasized: the use of multiple depth cues, the breakdown of visual information into several distinct types, and the spatial organization of a scene. The predictions of all depth cues are represented in a common form and combined by a single procedure. The cycling of the adjustment routine lets the higher weighted and more accurate depth information be spread over the scene. Combining cues which indicate an absolute distance, such as expected size or motion parallax, with cues that indicate relative positioning, such as occlusion or shadow, lets the program resolve the depth of regions that would be ambiguous using only one cue. The house in figure 2 is an example. Since the base of the house is largely screened by trees, it would be difficult to use

F I G U R E 19 .

EXECUTION TIMES

	Net Time:	Time Spent in Garbage Collection:	Total Time:
1.) Averaging 600X800 in red, green, blue to 200X200...	25sec.	-	25sec.
2.) Low Level Processing the Recognition Cone Layers 2 to 4	34.7sec.	-	34.7sec.
3.) Object Recognition: the Cone Layers 4 to 8	2min. 54sec.	1min. 43sec.	4min. 37sec.
4.) Total Recognition Cone...	3min. 54sec.	1min. 43sec.	5min. 37sec.
5.) Segmentation:			
House scene.....	1.7sec.	-	1.7sec.
Car scene.....	2.5sec.	-	2.5sec.
Room scene.....	1.9sec.	-	1.9sec.
6.) Segmentation and World Model Construction.....	19.3sec.	2.18sec.	21.5sec.
7.) World Model Construction with no Adjustment Cycling.....	11.7sec.	1.6sec.	13.3sec.
8.) Total Execution Time.....	4min. 13sec.	1min. 45sec.	6min.

the ground plane hypothesis directly on the house. But the ground plane hypothesis can be used on the grass and occlusion knowledge can indicate that the trees are touching the grass and that the house is behind the trees. In this way, the ground plane hypothesis can indirectly contribute to the depth determination of the house.

It was useful in the design of this vision model to structure the system's knowledge into four distinct types. The first type consists of operators which characterize the image in terms of a set of features or descriptors. This type includes such operators as edge detectors and measures for hue and texture. Object-specific knowledge is the second category of visual information used. It specifies what specific objects look like in terms of size, shape, and surface characteristics and in terms of expected contexts for the object. Object appearance is described in terms of the features of the first kind of visual knowledge. In this program, object-specific knowledge is represented by recognition cone transforms and by long-term object models. In contrast to object-specific knowledge, general visual knowledge represents information, largely independent of what particular objects might be in the scene, about the mapping of a three-dimensional scene onto a two-dimensional image. The final type of knowledge is the short-term world model. It is both the end result of processing a scene and the basis for processing the

next view. The world model is different from both object specific knowledge and general knowledge in that it is a description of the specific characteristics of the actual objects in view.

The appropriate output for a computer vision system depends upon what it is being used for. An optical character recognition machine need only supply names of letters in a left to right and top to bottom scan. A flexible, mobile industrial automaton needs to know not only what objects are in front of it but also fairly precisely where they are located. The vision system described here strives for the latter type of scene understanding. Some sort of spatial organization is necessary if a vision system is to know not only what kind of objects it is observing but how many and were they are. The use of multiple depth cues working on a common three-dimensional world model enables the present vision system to approach that type of scene description.

References

(1) Baumgart, B. G., "Geometric Modeling for Computer Vision." Stanford Art. Intel. Laboratory Memo AIM-249, 1974.

(2) Lewis, R. A. and A. K. Bejczy, "Planning Considerations for a Roving Robot with Arm," Proc. Third Int. Joint Conf. on Art. Intel., pp. 308-316, 1973.

(3) Finkel, R., R. Taylor, R. Bolles, R. Paul, and J. Feldman, "An Overview of AL, A Programming System for Automation," Proc. Fourth Int. Joint Conf. on Art. Intel., pp 758-765, 1975.

(4) Uhr, L., "Layered 'Recognition Cone' Networks that Preprocess, Classify and Describe," IEEE Trans. Comp., 21, pp. 758-768, 1972.

(5) Levine, M. D., "A Knowledge-Based Computer Vision System," Advanced Papers for the Workshop on Computer Vision System, Eds. E. Riseman and A. Hanson, 1977.

(6) Marr, D., "Representing Visual Information," Adv. Papers Workshop on Comp. Vision., Eds., E. Riseman and A. Hanson, 1977.

(7) Sakai, T., T. Kanade, and Y. Ohta, "Model-Based Interpretation of Outdoor Scenes," Proc. Third Int. Joint Conf. on Pattern Recog., pp. 581-585, 1976.

(8) Shirai, Y., "Recognition of Real-World Objects Using Edge Cues," Adv. Papers Workshop on Comp. Vision Systems., Eds., E. Riseman and A. Hanson, 1977.

(9) Hanson, A., E. Riseman, and T. Williams, "Constructing Semantic Models in the Visual Analysis of Scenes," Milwaukee Symposium on Automatic Computation and Control., pp. 97-102, 1976.

(10) Sloan, K., "World Model Driven Recognition of Natural Scenes," Ph.D. dissertation, unpublished, Univ. of Pa., Philadelphia, 1977.

(11) Uhr, L., Pattern Recognition, Learning and Thought. Englewood Cliffs, N.J.: Prentice-Hall, 1973.

(12) Ohlander, R., "Analysis of Natural Scenes," Ph.D. dissertation, Carnegie-Mellon Univ., Pittsburgh, 1975.

(13) Schacter, B., L. Davis, and A. Rosenfeld, "Scene Segmentation by Cluster Detection in Color Spaces," SIGART, No. 58, June, 1976.

(14) Zobrist, A. and W. Thompson, "Building a Distance Function for Gestalt Grouping," IEEE Trans. Comp., 24, 1975.

(15) Yakimovsky, Y., "Boundary and Object Detection in Real World Images," Journal Assc. Comp. Mach., 23, pp. 599-618, 1976.

(16) Nagel, H., "Experiences with Yakimovsky's Algorithm for Boundary and Object Detection," Proc. Third Int. Joint Conf. on Pattern Recogn., pp. 753-758, 1976.

(17) Barrow, H. and J. Tenenbaum, "Representation and Use of Knowledge in Vision," Stanford Research Inst. Tech. Note 108, 1975.

(18) Uhr, L. and R. Douglass, "Parallel-Serial Recognition Cones for Perception: Some Test Results," Proc. Fifth Int. Joint Conf. on Art. Intel., 1977.

(19) Ito, T., "Color Picture Processing by Computer," Proc. Fourth Int. Joint Conf. on Art. Intel., pp. 635-642, 1975.

(20) Rosenfeld, A., "Visual Texture Analysis: An Overview," Comp. Science Tech. Rept. Series TR-406, Univ. of Maryland, 1975.

(21) Tenenbaum, J. and H. Barrow, "IGS: A Paradigm for Integrating Image Segmentation and Interpretation," Proc. Third Int. Joint Conf. Pattern Recogn., pp. 504-513, 1976.

(22) Riseman, E. and M. Arbib, "Computational Techniques in the Visual Segmentation of Static Scenes," Comp. Graphics and Image Processing, 6, pp. 221-276, 1977.

(23) Tanimoto, S. and T. Pavlidis, "The Editing of Picture Segmentations Using Local Analysis of Graphs," CACM, 20, pp. 223-229, April 1977.

(24) Duda, R. and P. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

(25) Rosenfeld, A., R. Hummel, and S. Zucker, "Scene Labelling by Relaxation Operations," IEEE Trans. on Systems Man and Cybernetics, SMC-6, pp. 420-433, 1976.