

NOTE

Uncertain Reasoning and Learning for Feature Grouping¹

ZuWhan Kim and Ramakant Nevatia

Institute for Robotics and Intelligent Systems, University of Southern California, 3737 USC-Watt way, PHE 204, Los Angeles, CA 90089
E-mail: zuwhan@iris.usc.edu, nevatia@iris.usc.edu

Received October 30, 1998; accepted September 8, 1999

Hierarchical perceptual organization is needed for 3-D object detection and description. The *hypothesize and verify* paradigm offers one approach to this task. Hypotheses are generated from simpler features satisfying some possibly task dependent properties. More global evidence is used to verify and assign a confidence measure to the hypotheses. This evidence may consist of several components which may be of very different natures. How to combine these components in an effective and efficient manner becomes of critical interest. We describe formal methods that use neural networks and Bayesian approaches for verification. These approaches also allow automatic learning of parameters from some examples. We illustrate these methods using a system for building detection and description from aerial images, but the techniques themselves are not specific to this domain. Experimental results indicate that substantial improvements in performance can be obtained by the use of these methods without extensive hand-tuning of parameters.

1. INTRODUCTION

Perceptual organization of low-level features that can be extracted immediately from images into object level descriptions is a key task in computer and human perception processes. Perceptual organization techniques have been a persistent topic of research in computer vision [1-6]. Many of these techniques are concerned about organizing points into curves or about connecting curve fragments. Such tasks can often be accomplished by applying a single organization rule. However, when object level organization is desired, it becomes clear that the process needs to be a hierarchical one with the simpler features being successively grouped into more global and complex features. The hierarchical nature of the organization does not imply that a feature cannot be grouped into more than one other level nor that the organization process itself must proceed in a strict bottom-up (or top-down) manner.

One approach to hierarchical perceptual organization is a hypothesize and verify paradigm (Figure 1). Grouping hypotheses at various levels are generated if some features satisfy a set of properties; these properties are intended to involve relatively small sets (say two) of elements and to be relatively simple to compute. This may cause features to be grouped into multiple, possibly overlapping hypotheses that are alternative explanations for the same part of the scene. Further evidence for the presence of the generated hypotheses is then collected; this evidence need not be confined to the 2-D properties in the image but may include 3-D inferences. The computed evidence is then combined to verify the presence of each hypothesis; the verification process assigns a confidence value to each hypothesis. The verification process can also be applied at other levels of the grouping hierarchy so that less expensive evidence is used to filter out unlikely hypotheses and more expensive analysis is applied to fewer hypotheses.

Key issues in a hierarchical hypothesize and verify approach are the representation levels and the properties and procedures to be used for generating and verifying the hypotheses. In this paper, we take the view that the representation levels and grouping criteria for perceptual organization are to be decided by a designer and may be domain-specific (design of general systems emulating human performance is another matter). A common problem for all such systems, however, is defining good strategies for combining evidence from the various available sources. Evidence for grouping, for example, may come from proximity, continuity, and symmetry. We need to combine such pieces of evidence in making a judgement of the quality of a hypotheses. The evidence to be combined can be of very different nature (not necessarily binary), for example, it may consist of 3-D information such as coplanarity, photometric information such as homogeneity, and feature regularity such as symmetry or completeness.

1. This research was supported by a MURI subgrant from Purdue University under Army Research Office grant No. DAAH04-96-1-0444.

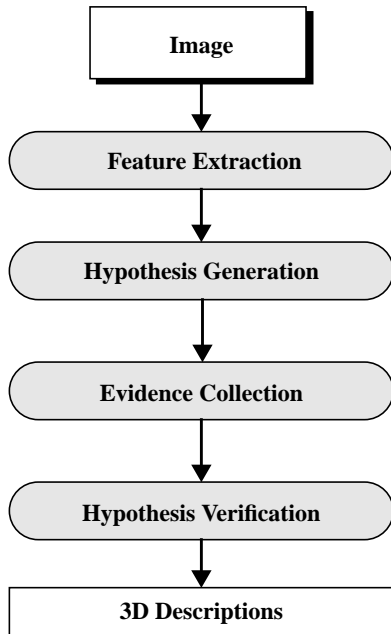


FIG. 1. Block diagram of a perceptual grouping process

Quantitative rules for combining such evidence can be difficult to infer directly from mathematical analysis. It is common for system designers to devise their own ad hoc methods for combining evidence based on their intuition and experience.

Good results can be obtained by this method, but the process of testing and modifying is a tedious one and the evidence is not necessarily used to best advantage. This paper explores more systematic approaches for the task of combining evidence; our approach includes a learning phase which helps reduce the amount of effort needed by a developer to achieve good performance. Use of formal methods of combining evidence and of using learning techniques is rare in perceptual organization methods; two previous methods may be found in [7, 8].

We illustrate our methodology in the context of perceptual grouping for a system for building detection and description from aerial images. We believe that the methodology we present should easily transfer to other systems that perform hierarchical grouping of similar complexity but in very different domains and using very different grouping criteria. We have experimented with methods using neural networks and Bayesian approaches. These methods, their application to perceptual grouping and comparative results are provided. Before introducing these methods, we give a brief description of the test system used to evaluate them.

2. PERCEPTUAL ORGANIZATION FOR BUILDING DETECTION

Detection and construction of 3-D models of buildings in an aerial image is of interest for many applications. The problem also provides a nice domain for study of perceptual organization techniques. Even though we restrict to one broad class of objects,



FIG. 2. An aerial image from Ft. Hood.

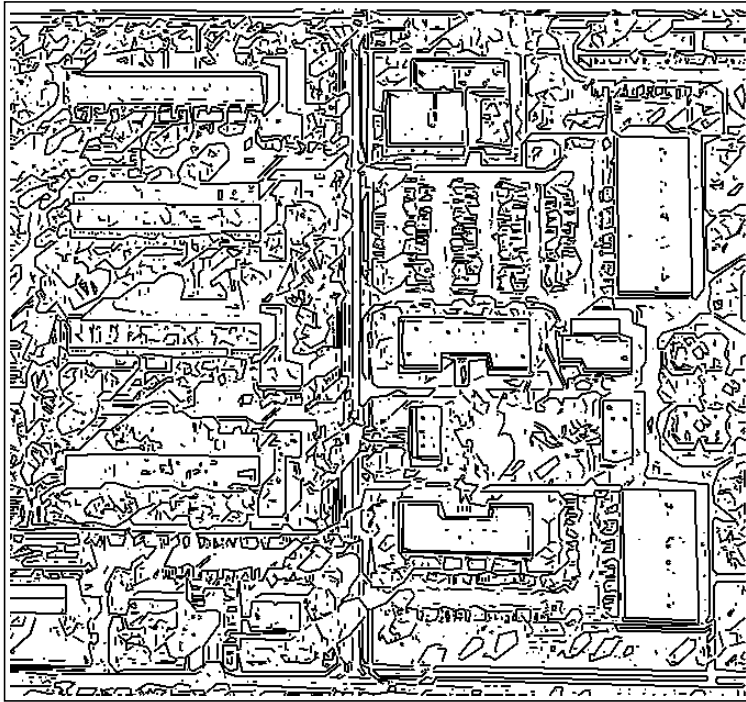


FIG. 3. Line segments detected in Figure 2.

a large number of instances are possible. Aerial images also are typically highly complex and contain large numbers of objects which makes the figure-ground separation problem particularly difficult. Consider for example the image shown in Figure 2. Figure 3 shows the line segments detected in this image using a Canny edge detector [9] followed by a line linking and approximation method [10]. This figure illustrates the difficulty of the grouping task. The desired objects do not have closed boundaries, and a large number of extraneous boundaries corresponding to other objects, architectural details, shadows, markings, and texture are present.

Many systems to detect buildings from aerial images have been developed [11]. We describe experiments with a system called BUDDS (BUilding Detection and Description System), developed by Lin and Nevatia. Details of this system are available elsewhere [12], here we only provide a brief overview needed to describe our methods for combining uncertain evidence. BUDDS uses only a single image and is limited to the case of flat rectilinear buildings which are modeled as composition of rectangular parts. The roof of a rectangular part projects to a parallelogram in an image (assuming orthographic projection which is accurate for most aerial images over the size of a building). Hypotheses for rooftops are generated by looking for parallelograms among the detected line segments. This is a hierarchical process, lines are grouped into parallel pairs, then into “U” shapes, and finally into parallelograms. The hypotheses are generated based on partial information only, parts of lines or even complete lines may be missing. The hypotheses are then

selected based on the strengths of underlying image evidence such as the coverage of the hypotheses by actual image lines, presence of corners and lack of crossing lines; details of this process may be found in [12]. Figure 4 shows the selected hypotheses computed by BUDDS from the lines of Figure 2. Note that many overlapping hypotheses remain consistent with the least commitment style of design indicated earlier.

The next step in the process is that of verification. For this purpose, each 2-D parallelogram hypothesis is given the interpretation of being a 3-D rooftop. The height of the hypothesis is inferred by finding the visible walls and the shadow on the ground. It is common for aerial images to have available the sun position and also a complete camera model relative to some ground coordinate system. Note that the wall and shadow evidence itself requires a search and is uncertain. Both wall and shadow evidence themselves consist of smaller components. The wall evidence consists of evidence for the vertical sides and for the horizontal line on the ground. The shadow evidence consists of the shadows of the lines of the rooftops and of the vertical lines. Knowledge of the camera model and the sun position allow us to compute the parts of the evidence that should be visible; the observed values are normalized by these. Table 1 summarizes the evidence used in BUDDS.

The main task addressed in this paper is how best to combine the various pieces of evidence available for the presence of a rooftop, the walls and the object shadow to decide if a building is actually present. As the information sources are fragmentary, it is also desirable to assign a confidence value to the decision which

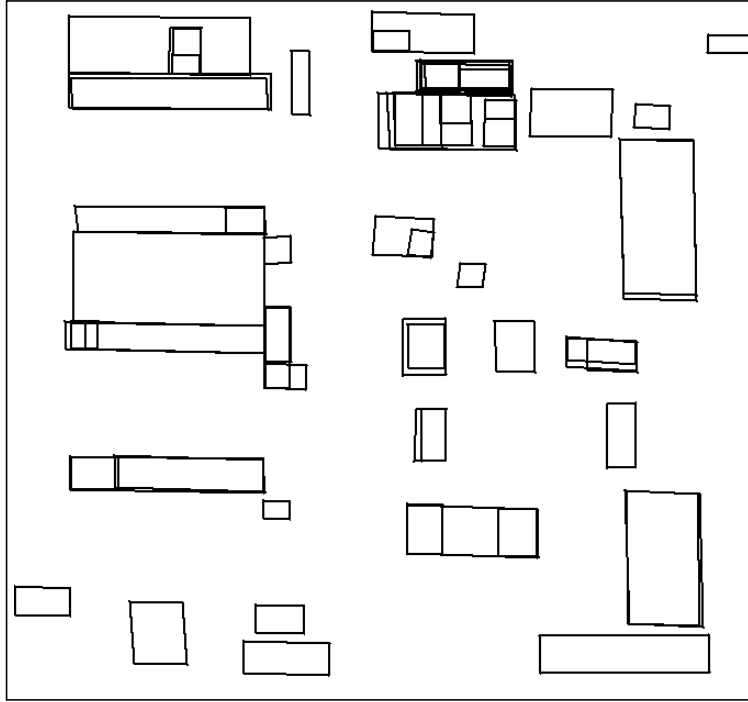


FIG. 4. Building hypotheses of Figure 2.

may be used in further selection depending on the utility of the decision for a particular task. We view this as the key step in completing the perceptual organization process, as it is the one that makes a decision of whether a figure (i.e., an object) actually exists with high enough confidence.

3. COMBINING EVIDENCE FOR HYPOTHESIS VERIFICATION

For this discussion, we consider the problem to be formulated as follows. For a hypothesis, say H , we are given a set of evidence, say E , which consists of various components say E_i . In this paper, for random variables, boldface letters will be used while the default font will be used for domain values (instances). In the same manner, $P(X)$ is a probability distribution on a random variable X while $P(X)$ is a probability of X being a particular instance X . H represents a given hypothesis being a

building while \bar{H} represents a false building hypothesis. The goal is to use these to compute the likelihood of hypothesizing H . We now describe some approaches to combining evidence to make the desired decision.

3.1 Linear Combinations

The simplest method of combining evidence is by taking a linear weighted sum of the components. The weights can be normalized so that the sum remains between 0 and 1. In this case,

$$H = \begin{cases} \text{Building,} & \text{if } \sum_i w_i E_i > \theta, \\ \text{Not Building,} & \text{otherwise,} \end{cases}$$

where w_i are the weights for the respective evidence E_i and θ is a threshold.

3.2 Certainty Factors

Another popular method is to treat each evidence as providing a certainty factor for the hypothesis and to combine the various pieces by the method proposed in the MYCIN system [13]. The combined evidence E_{ij} from the evidence E_i and E_j are obtained by the following rule:

TABLE 1
Evidence used in BUDDS.

Category	Evidence
Wall	Ground, Corner
Shadow	Roof-cast, Corner, Region
Roof	Standard-deviation

$$E_{ij} = \begin{cases} E_i + E_j - E_i \times E_j, & \text{when } E_i, E_j \geq 0, \\ E_i + E_j + E_i \times E_j, & \text{when } E_i, E_j < 0, \\ \frac{E_i + E_j}{1 - \min(|E_i|, |E_j|)}, & \text{otherwise.} \end{cases}$$

Since this rule is commutative, we can apply this rule to all the evidence repeatedly and combine them into one value which is to be thresholded to hypothesize H .

3.3 BUDDS Method

In the BUDDS, as described in [12], both linear combination and certainty factors are used. For each set of evidence (*e.g.*, roof, wall, and shadow), E_i are combined by linear weighting into R , W and S . The resulting combined evidence R , W and S are then combined by using certainty factor rules given above. A decision on verifying a hypothesis can be made by thresholding the combined certainty factor. This choice of combining evidence differently at the two stages was made on an *ad hoc* basis. The weights chosen for combining evidence in each set were determined by human experimentation with a number of test cases; these weights are same for all images, however.

3.4 Neural networks

Neural networks offer a popular method for decision making in presence of uncertainty, particularly when relationships between the output and input variables are not available from mathematical analysis. Each node (neuron) of such a network computes a weighted sum of its inputs and then applies a thresholding or a sigmoid function;

$$E = g\left(\sum_i w_i E_i\right),$$

where E is the combined evidence and g is a sigmoid function which approximates a step function (threshold function).

In fact, a *perceptron*, a neural network with a single node, is equivalent to a linear combination such as is explained in Section 3.1. In general, neural networks are more complicated networks with the outputs of several neurons feeding into the inputs of another neuron.

The programming of such a network can be considered to consist of two components: design of the network topology and assignment of the weighting parameters. It is common to apply a machine learning algorithm to determine the parameters for which several techniques exist, with *back propagation* being the most popular method for multilayer feedforward networks. The topology may also be learned in principle, but this is a much more difficult task. We think that it is more natural for a designer to fix the topology for the perceptual organization problems. We consider two cases.

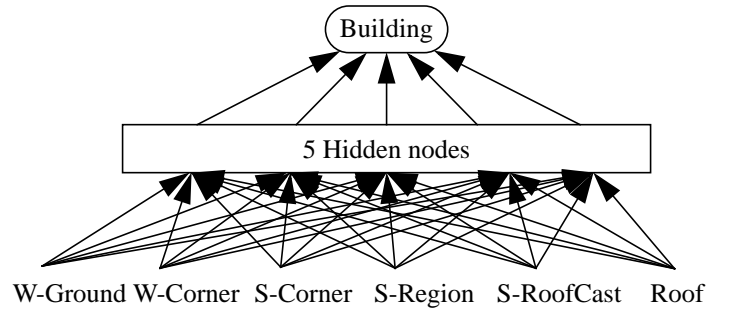


FIG. 5. A two-layer feed-forward neural network

Generic two-layer feedforward network: In this approach, the neural network topology consists of two layers. The input layers feed into a layer consisting of some number of hidden nodes whose outputs feed into a single output node (several output nodes may be used for multiple output values). The number of *hidden* nodes can have a significant influence on the performance of the system. However, the problem of choosing the right number of hidden units in advance is not well understood [14]. It could be determined by experimenting with different numbers of nodes (and learning optimal weights for each case). In our system, we used five hidden nodes for six input nodes, as shown in Figure 5.

A neural network with hidden layers is easy to design but has some major disadvantages. First is the lack of *transparency*. It is hard to understand the computations being carried out in the hidden layer and hence for a developer to intervene to improve performance. The second disadvantage is that knowledge of the internal structure of the evidence is not used efficiently. For example, the standard deviation of the roof is independent of the corner evidence of shadow but the shadow region is not. The neural networks with hidden layers combine all the evidence together into the hidden nodes and hence may suffer from serious *overfitting* problems.

Structured feed-forward neural network: For hierarchical perceptual organization problems, there is a natural structure to the available evidence. We can take advantage of this structure by having the neural network reflect it. Figure 6 shows a *structured* neural network that reflects the natural dependencies of the evidence for our system. In fact, this network is less powerful in class representation than the generic two-layer network of Figure 5 - for any network instantiation of the structured network, there exists a network instantiation of the generic two-layer network that is equivalent. Nevertheless, the structured network has an advantage over the generic two-layer network; it is *transparent*. The parameters of each node reflect the importance of the evidence, so that it is easy for the system developer to *understand* the meaning of the parameters. It is also good for learning, which will be discussed in Section 4.1.

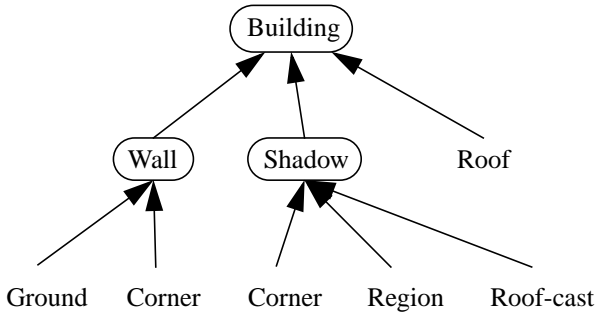


FIG. 6. A hand-structured neural network

3.5 Bayesian Classifier

Bayes' rule offers an optimal approach to making decision about a hypothesis, \mathbf{H} , given evidence E , by computing the conditional probability $P(\mathbf{H}|E)$. Bayes rule gives:

$$\begin{aligned} P(\mathbf{H}|E) &= \frac{P(E|\mathbf{H})P(\mathbf{H})}{P(E)} \\ &= \frac{P(E|\mathbf{H})P(\mathbf{H})}{P(E|\mathbf{H})P(\mathbf{H}) + P(E|\bar{\mathbf{H}})P(\bar{\mathbf{H}})} \end{aligned}$$

for a binary hypothesis \mathbf{H} .

The difficulty in applying Bayes' rule is in acquiring and representing the knowledge of the joint probability distribution $\mathbf{P}(\mathbf{E}|\mathbf{H})$. If E consists of n elements, $\mathbf{P}(\mathbf{E}|\mathbf{H})$ is an n -dimensional function. For example, if each element can be quantized into m values, we need to know a table of size m^n . It is not feasible to measure such a function even for problems with small number of evidence variables.

One simplification of this problem is to assume that all elements of E are *conditionally independent* given \mathbf{H} , the resulting classifier is known as a *naive* Bayes classifier [15]. In this case, $P(\mathbf{E}|\mathbf{H}) = \prod P(E_i|\mathbf{H})$; therefore, for the discrete case, we only need a table of size $m \times n$, which still shows good results for a number of problems [15].

In our problem domain, the probability distributions, $\mathbf{P}(E_i|\mathbf{H})$, are continuous ones. We need to represent such continuous distributions effectively. There are several considerations for an effective representation. It should be nonnegative and the total area of the distribution should sum up 1 since it is a probability distribution. The number of parameters should be small enough to avoid *overfitting* while allowing representations of a variety of distribution shapes. In general, there is a trade-off between the number of parameters and the representational power.

With these considerations, two different approaches can be applied. The first is to use parametric curves such as normal distribution curves. In fact, it is hard to find good parametric distributions, other than normal distributions, which satisfy the above

requirements. Unfortunately, the evidence distributions for BUDDS do not fit normal distributions well. Therefore, we applied the second approach, which is to quantize the distributions into several discrete levels. For BUDDS, we discretized evidence inputs for five different levels; the learning of the distributions and the results are shown in Section 5.

3.6 Bayesian Networks

The assumption of independence of evidence variables is not always justifiable. An attempt to keep the simplicity of the naive Bayes classifier but improve performance is described in [16]. An alternative is the concept of Bayesian networks, which have recently become popular for handling uncertainty in many applications [17, 18]. A Bayesian network is a graph representation with random variables as its node. Edges represent causal relationships where, given a parent node, all the child nodes are *conditionally independent*. Each node has a set of conditional probabilities as a function of its parent nodes (represented as a conditional probability table, CPT, for quantized variables). Probabilistic reasoning is done by applying Bayes' rule on the CPT's and the prior probabilities of the nodes and propagating it through the network. For example, the joint probability of nodes X_1, \dots, X_n is given by the following formula:

$$P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Parents}(X_i)).$$

The process of computing probability distributions of some nodes given probability distributions of other nodes is the process of Bayesian inference. Algorithms for such inference can be found in [17]; [14] contains a simple algorithm for *singly connected* networks.

Bayesian networks can accurately represent any joint probability distribution function but they exploit the conditional independence to reduce the dimensionality of the function or the number of parameters that define the network. They have the advantage of being transparent as the relations between the variables are made explicit. Generally, when the structure of a network reflects the causal relations between the variables, it is called a *causal* Bayesian network.

For BUDDS, we can find several causal relationships. For example, a strong building hypothesis causes strong wall evidence which causes strong ground and corner supports for the wall evidence. Figure 7 shows a causal Bayesian network reflecting those relationships. In this case, the wall and the shadow nodes are set to be *hidden* (unobservable) because it is tedious and difficult for the system developers to assess the strengths of wall and shadow evidence for an example.

As mentioned in Section 3.5, all the evidence variables of BUDDS are continuous. Thus, we need to represent the conditional probability distributions effectively. For Bayesian networks, there are more restrictions. First, the distributions are multidimensional since one node can have several continuous (or discrete) parent nodes. Therefore, the conditional probabilities

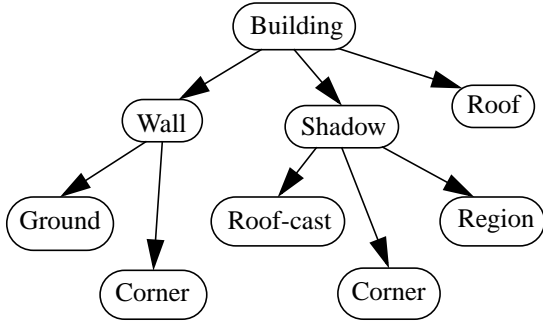


FIG. 7. A causal Bayesian network.

are tables of multidimensional functions. Also, for a discrete node with continuous parents, an *interfunctional* normalization rule is applied:

$$\sum_X f_{X|Y_1, Y_2, \dots, Y_n}(y_1, y_2, \dots, y_n) = \mathbf{1},$$

where X is a continuous node, Y_i are its parent nodes, y_i are the function parameters for Y_i , $f_{X|Y_1, \dots, Y_n}(y_1, \dots, y_n)$ is a conditional probability density function, and $\mathbf{1}$ is a constant function of value 1. It is almost impossible to find a parametric multidimensional distribution which satisfy all those conditions. In fact, the problem becomes easier for BUDDS if we regard hidden nodes (wall and shadow) as being discrete since the network has a restricted structure that all the continuous nodes have single discrete parents and no child. Still, this was not applied for BUDDS since we could not find good one-dimensional distributions as mentioned in Section 4.2. For BUDDS, once again, we quantized the evidence nodes into 5 different levels while the hidden nodes were quantized into 3 levels. For several approaches with parametric distributions, see [17, 18, 19].

Another approach to apply Bayesian methods to the perceptual organization problem was studied by Sarkar and Boyer [7]. They introduced PIN (Perceptual Inference Network), an augmented Bayesian network with spatial parameters in each node, for the hypothesis formation process. This spatial information needs to be incorporated in the reasoning process. In our approach, the spatial information is already included in the evidence collected for each hypothesis and is not explicitly included in the Bayesian inference process; therefore, a general causal network without augmentation is sufficient. Instead, we focus on dealing with hidden nodes to take advantage of the perceptual organization structures during the reasoning process. Section 4.3 describes learning of parameters associated with the Bayesian network in presence of hidden nodes.

4. LEARNING

The methods described in Section 3 require several parameters to be set for good performance. For the simpler methods,

such as the linear combinations and certainty factors, it is viable to set the parameters by a human designer experimenting with some number of examples. In fact, for BUDDS method of Section 3.3, the parameters were hand-tuned by the system developers. However, for other methods such as neural networks, Bayesian classifiers, and Bayesian networks, it is not practical to do so because of the complexity of the reasoning process; instead, we focus on learning of these parameters by applying machine learning techniques with several examples.

The output for BUDDS verification is binary; *BUILDING* or *NONBUILDING*. In this case, there is a trade-off between the number of *false negatives* (undetected buildings) and *false positives* (falsely detected buildings). It is common in machine learning to weigh the two kinds of errors equally; however, this is not always consistent with the application needs. For building detection, depending on the task, one or the other may be more troublesome. For example, if the results are to be edited by a user, removing false positives may not be so difficult, but if the results of the automated process are to be used directly, it may be important to weight the errors of false positives more than that of false negatives. The relation between false positives and false negatives is usually controlled by changing the threshold for the output, however, it can also be incorporated into the learning process (for neural networks and Bayesian networks) as explained later. We call the ratio between the cost of a false positive and cost of a false negative the *cost ratio* and the learning parameters which influence the cost ratio *cost ratio parameters*. By incorporating a cost ratio parameter into the learning process, we can optimize the system performance for a given cost ratio to obtain a better result than by merely changing the threshold value for the outputs.

4.1 Neural Networks

A back-propagation learning algorithm for neural networks minimizes the squared sum of the errors between the results and desired outputs;

$$Err = \sum_k (T_k - O_k)^2, \quad (1)$$

where Err is the total error to be reduced, T_k is the target value of the k th example, and O_k is the output of the current network. However, this does not support different costs for *false negatives* and *false positives*. Instead, the results depend on the ratio between the number of positive examples and negative examples in the learning set. To compensate for this we modify the Err function in (1) as follows:

$$Err = \sum_k \alpha_k (T_k - O_k)^2,$$

where $\alpha_k = (1 - \lambda)/n_p$ for positive examples, $\alpha_k = \lambda/n_N$ for negative examples, n_p and n_N are the numbers of positive and

negative examples, and λ is a *cost ratio parameter*. For our tests with BUDDS, λ was fixed to be 0.7.

The learning rule is to change weight w of each node to reduce Err by *gradient descent* (hill-climbing):

$$\Delta w = -\frac{\partial Err}{\partial w} = 2 \sum_k \alpha_k (T_k - O_k) \frac{\partial O_k}{\partial w} .$$

Computation of the derivative in the above equation follows the standard methods used in neural network back-propagation learning as described in, for example, [20] or [14].

It is very important to give a good *seed* (set of initial parameters) for the *gradient descent* approaches. For the generic network, it is hard to give a good seed for learning, so we repeated learning with several random seeds and chose the best result. For the structured network, since it is *transparent* and simple, we can manually set a intuitively good seed which in fact showed relatively good results (Section 5). Moreover, the learned parameters of the structured network reflect the importance of the evidence allowing the evaluation of the importance of evidence by observing the weights in the network. In fact, this process allowed us to remove several evidence nodes which made small contributions to the other nodes.

4.2 Naive Bayesian Classifiers

Learning for naive Bayesian classifiers is rather straightforward. In this case, we estimate the needed probability distributions $P(E_i|H)$ and the prior probabilities $P(H)$ by observing the occurrence of evidence and hypothesis and simply histogramming these quantities.

4.3 Bayesian Networks

Learning parameters for Bayesian networks with hidden nodes (i.e., nodes whose values we do not observe directly) is a difficult task. In [21], Kowh and Gillies developed a Bayesian network learning control system for an endoscope. They minimized the squared sum of the error as for neural networks in Section 4.1 above.

We use an approach following recent work of Binder *et al.* [18] which is based on *gradient ascent*. In their approach, the parameters are set to maximize the probability of observed variables, $P(H_1, E_1, H_2, E_2, \dots, H_n, E_n)$ in our case. However, for the verification task, we can make a better objective function;

$$\max \left[(1 - \beta) \frac{\sum g(O_P)}{n_P} + \beta \frac{\sum g(O_N)}{n_N} \right], \quad (2)$$

where β is a constant, $O_P = P(H|E_P)$ is an output of a positive example, $O_N = P(\bar{H}|E_N)$ is an inverted output of a negative example, n_P and n_N are the numbers of positive and negative examples, and g is a sigmoid function which approximates a step function. It maximizes the sum of *true positives* (detected

buildings) and *true negatives* (undetected nonbuildings). β is a *cost ratio parameter*. The higher values of β result in sacrificing some true detections to avoid a larger number of false alarms. The objective function is maximized by a hill-climbing search algorithm by taking the partial derivatives of (2)¹. The details of this derivation are given in the Appendix.

5. EXPERIMENTAL RESULTS

To compare the suggested inference and learning techniques to the original BUDDS method, we created several learning data sets from several aerial image windows. First, rectangular building component hypotheses were generated from each image window. From the hypotheses, we created a learning data set by displaying one hypothesis at a time to a human and asking for a decision on whether it represents the rooftop of a building or not. In most cases, there is no difficulty in making this decision. Those hypotheses which have significant overlap with the rooftops but are not accurately positioned were regarded as *don't care* and not used as learning examples. The main reason for not using these hypotheses in learning is that the evidence for them can be highly variable. Some of them have very good evidence because they include parts of buildings but some of them have poor evidence because of the incorrect positioning of the roof. It is not efficient to ask a human to look at the actual evidence values and to decide whether the hypothesis should be considered positive or negative. Also, the desired classifications for the *don't care* hypotheses are not clear. If such a hypothesis is selected, and no better hypothesis exists, it is an acceptable answer. If a better hypothesis does exist, the better one will dominate in a later overlap analysis phase. If a *don't care* hypothesis is rejected because of poor evidence, this decision too is acceptable. Hence, we do not use such hypotheses in either learning or evaluating the system.

An interactive tool was developed to collect learning data sets efficiently. As a result, 591 learning data sets were collected from 10 images of 3 different sites; 165 of them were determined as positive hypotheses, 303 were negative, and 123 were *don't care*. Figure 8 shows example hypotheses collected on an image window.

We use the results of the handcrafted BUDDS as a baseline to compare against other methods. We implemented a generic two layer neural network (Figure 5), a structured neural network (Figure 6), a naive Bayesian classifier, and a Bayesian network with discretized evidence (Figure 7). For the Bayesian network, continuous evidence values were discretized into 5 levels while the hidden nodes were set as 3-level discrete nodes. Initial sets of parameters for the structured neural network and the Bayesian

1. Besides (2), we have two more constraints: the sum of CPT entries with a given parent instance equals to 1 and all the parameters lie on [0,1]. Therefore, we first need to obtain the derivatives of (2) without applying these constraints, then project the resulting parameters onto the constraint space.



FIG. 8. Example hypotheses. Positive hypotheses are shown in red, negatives are blue, and the yellow ones are the *don't care* hypotheses.

network were given manually, while, for the generic neural network, several random sets of parameters were learned and the best result was chosen.

It is common to characterize the results in terms of detection rate and false alarm rate. In this paper, detection rate is the ratio of *true positives* to all the positive hypotheses and false alarm rate is the ratio of *false positives* to all the positive hypothesis. As already mentioned, a trade-off can be exercised between the detection rate and the false alarm rate. Rather than choose one value, we show the trade-off curves as in 8.. These are commonly called ROC (“Receiver Operating Characteristics”) curves. Figure 9 shows ROC curves for each method. In this paper, the curves were obtained by simply changing the threshold value for the continuous outputs of each method. For the lower threshold

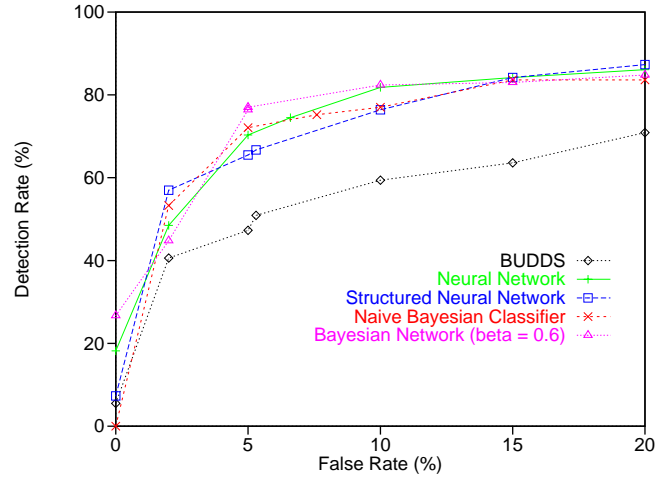


FIG. 9. ROC curves for various methods.

values, the more number of building components are detected but also more false alarms are detected, and for the higher threshold values, less false alarms but also less building components are detected.

The results clearly show that the automated learning methods show much better performance over most of the operating range than the original method. For example, keeping the false alarm rate fixed at 5%, we see at most 29% point improvement (Bayesian network) on the detection rate. Also, note that the structured neural network shows a competitive result with the generic neural network. In fact, the generic neural network did not show bet-



(a)



(b)

FIG. 10. Results on the image window in Figure 2; (a) BUDDS method, and (b) Bayesian network.

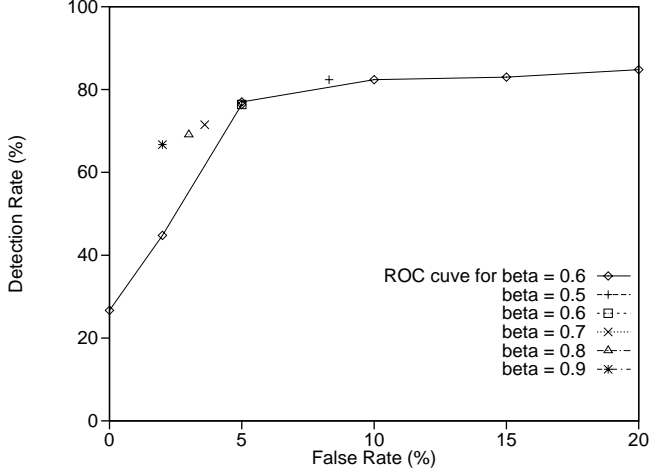


FIG. 11. Bayesian network results with various cost ratio parameters. The ROC curve is for $\beta = 0.6$. Threshold value 0.5 was applied for the other points.

ter performance than the structured neural network for the most of the random seeds applied, although only the best result is shown in the figure. This illustrates that applying the knowledge from structure of the perceptual organization problem can reduce a large amount of effort while it also provides a stable result.

Figure 10 shows the grouping results displayed in the image window in Figure 2. We only show the results of the BUDDS method and the Bayesian network method as the results from other learning methods are quite similar to those of the Bayesian network method for this example. We measure performance for each rectangular component of a rectilinear building. It is clear that the Bayesian network detects more building components with fewer false alarms. While the BUDDS method found two false alarms including one of a large size in the left side, the Bayesian network found only one in the top left corner. Also, the Bayesian network detected three more building components than the BUDDS method, including a small building in the bottom, but missed one component in the middle.

Figure 11 shows the results on Bayesian networks with various values of the cost ratio parameter β of (2). In this figure, the ROC curve is the result for $\beta = 0.6$. Other points show the detection and false alarm rates for various cost ratio parameters when the standard threshold value of 0.5 is applied. It shows that, with the proposed learning method for Bayesian networks, we can emphasize a different point on the ROC curve by simply changing the cost ratio parameter. By incorporating the cost ratio parameters into the learning procedure, we get better results than by controlling the cost ratio by merely changing the threshold values. However, the Bayesian network is more susceptible to the *overfitting* problem, which can cause the results on test data to be not as good as for the training data.

6. CONCLUSION AND FUTURE WORK

We have presented some algorithms for making verification decisions with uncertain evidence and shown an experimental evaluation for a building detection system. It is clear in this example that the results can be significantly improved by use of a systematic methodology such as the Bayesian networks. Furthermore, the development time and effort can be substantially reduced by using machine learning algorithms.

We believe that the presented methods are not specific to the particular system used in our tests or to the building detection task. These methods should apply to all perceptual grouping tasks that collect evidence from a number of cues to make a decision on which groupings are to be preferred. Also, we have only applied this methodology to the final stages of the grouping process. It can also be applied to earlier stages (as in [8] for example).

APPENDIX

To maximize the objective function of (2), the partial derivatives on the parameters (CPT entries) are used. For a discrete node X^1 ,

$$\begin{aligned} \Delta w_i &= \frac{\partial}{\partial w_i} \left[(1 - \beta) \frac{\sum g(O_P)}{n_P} + \beta \frac{\sum g(O_N)}{n_N} \right] \\ &= \frac{(1 - \beta)}{n_P} \sum g'(O_P) \frac{\partial O_P}{\partial w_i} + \frac{\beta}{n_N} \sum g'(O_N) \frac{\partial O_N}{\partial w_i}, \end{aligned}$$

where $w_i = P(X_i|U_i)$ is the i th CPT entry (instance combination) of X on its parent node(s) U . By averaging over the possible instances of X and U , we get

$$\begin{aligned} \frac{\partial}{\partial w_i} P(H|E) &= \frac{\partial}{\partial w_i} \sum_j P(H|X_j, U_j, E) P(X_j, U_j|E) \\ &= \frac{\partial}{\partial w_i} \sum_j P(H|X_j, U_j, E) \cdot \\ &\quad \frac{P(E|X_j, U_j) P(X_j|U_j) P(U_j)}{P(E)} \end{aligned}$$

Note that $P(H|X_j, U_j, E)$, $P(E|X_j, U_j)$, and $P(U_j)$ are independent on all CPT entries w_i of the node X . $P(E) = \sum_k P(E|X_k, U_k) P(X_k|U_k) P(U_k)$ can be represented as $P(E|X_i, U_i) P(X_i|U_i) P(U_i) + c$, where $c = \sum_{k \neq i} P(E|X_k, U_k) P(X_k|U_k) P(U_k)$, which is independent on w_i . For simplicity, λ_i will stand for $P(H|X_i, U_i, E)$ and $f(w_i)$ for $P(E|X_i, U_i) P(X_i|U_i) P(U_i)$. Then

1. A similar derivation can also be applied for the continuous nodes. We do not include this as it is not used in the system described here.

$$\frac{\partial}{\partial w_i} P(H|E) = \frac{\partial}{\partial w_i} \sum_j \lambda_j \frac{f(w_j)}{f(w_j) + c} = \sum_j \frac{\partial}{\partial w_i} \left(\lambda_j \frac{f(w_j)}{f(w_j) + c} \right).$$

For $i \neq j$, we get

$$\frac{\partial}{\partial w_i} \left(\lambda_j \frac{f(w_j)}{f(w_j) + c} \right) = \frac{-\lambda_j f(w_j) f'(w_i)}{(f(w_i) + c)^2},$$

and for $i = j$, we get

$$\frac{\partial}{\partial w_i} \left(\lambda_j \frac{f(w_j)}{f(w_j) + c} \right) = \frac{f'(w_i) \lambda_i (f(w_i) + c) - f'(w_i) \lambda_i f(w_i)}{(f(w_i) + c)^2},$$

where $f'(w_i) = \frac{\partial}{\partial w_i} (P(E|X_i, U_i)P(X_i|U_i)P(U_i)) = P(E|X_i, U_i)P(U_i)$.

Thus, by applying simple algebra to the above equations, we get

$$\begin{aligned} \frac{\partial}{\partial w_i} P(H|E) &= \frac{f'(w_i)(\lambda_i(f(w_i) + c) - \sum_j \lambda_j f(w_j))}{P(E)^2} \\ &= \frac{P(E|X_i, U_i)P(U_i)(P(H|X_i, U_i, E)P(E) - \sum_j P(H|X_j, U_j, E)P(E|X_j, U_j)P(U_j))}{P(E)^2} \end{aligned}$$

Here, $P(H|X_i, U_i, E)$ and $P(U_i)$ can easily be obtained by applying the Bayesian network inference rules. $P(E)$ can also be obtained by the same method. Once we know $P(E)$, $P(E|X_i, U_i)$ can be obtained with less computation by applying Bayes' rule:

$$P(E|X_j, U_j) = \frac{P(X_j, U_j|E)P(E)}{P(X_j, U_j)}.$$

Note that $P(U_i)$ and $P(X_j, U_j)$ is independent of the inputs, so they can be calculated only once for each iteration. Also, $P(E)$ is independent on X and U , so it can be calculated once for each data instance.

REFERENCES

1. R. Mohan and R. Nevatia, "Using Perceptual Organization to Extract 3-D Structures," *IEEE Trans. PAMI*, vol. 11, no. 11, pp. 1121-1139, 1989.
2. A. Shashua and S. Ullman, "Grouping Contours by Iterated Pairing Network," *Neural Info.*, vol. 3, pp. 335-341, 1991.

3. Mi-Suen Lee and Gerard Medioni, "Inferring Segmented Surface Description from Stereo Data," *Proceedings of CVPR*, Santa Barbara, pp. 346-352, 1998.
4. L. R. Williams and A. R. Hanson, "Perceptual Completion of Occluded Surfaces," *Computer Vision and Image Understanding*, vol. 64, no. 1, pp. 1-20, 1996.
5. D. W. Jacobs, "Robust and Efficient Detection of Salient Convex Groups," *IEEE Trans. PAMI*, vol. 18, no. 1, pp. 23-37, 1996.
6. D. G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355-395, 1987.
7. S. Sarkar and K. L. Boyer, "Using Perceptual Inference Networks to Manage Vision Process," *Computer Vision and Image Understanding*, vol. 62, no. 1, pp. 27-46, 1995.
8. M. A. Maloof, P. Langley, T. O. Binford, and R. Nevatia, "Generalizing over aspect and location for rooftop detection," *Proceedings of WACV*, Princeton, 1998.
9. J. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 679-698, 1986.
10. R. Nevatia and R. Babu, "Linear Feature Extraction and Description," *Computer Vision, Graphics and Image Processing*, vol. 13, pp. 257-269.
11. A. Gruen and R. Nevatia (editors), *Computer Vision and Image Understanding: Special Issue on Automatic Building Extraction from Aerial Images*, Vol. 72, No. 2, 1998.
12. C. Lin and R. Nevatia, "Building Detection and Description from a Single Intensity Image," *Computer Vision and Image Understanding*, Vol. 72, No. 2, pp. 101-121, 1998.
13. B. Buchanan and E. Shortliffe (editors), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison Wesley, Massachusetts, 1984.
14. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.
15. G. H. John, and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.
16. C. Elkan, "Boosting and Naive Bayesian Learning," TR No. CS97-557, University of California, San Diego, 1997.
17. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., 1988.
18. J. Binder, D. Koller, S. Russell, and K. Kanazawa, "Adaptive Probabilistic Networks with Hidden Variables," *Machine Learning*, vol. 29, pp. 213-244, 1997.
19. K. G. Olesen, "Causal Probabilistic Networks with Both Discrete and Continuous Variables," *IEEE Trans. PAMI*, vol.15, no.3, 1993.
20. M. A. Arbib, *The Metaphorical Brain 2*, pp. 384-386, John Wiley & Sons, 1989.
21. C. Kwok, and D. F. Gillies, "Using hidden nodes in Bayesian networks," *Artificial Intelligence*, 88, pp. 1-38, 1996.