Using Perceptual Inference Networks to Manage Vision Processes

Sudeep Sarkar* and Kim L. Boyer

Signal Analysis and Machine Perception Laboratory, Department of Electrical Engineering, Ohio State University, Columbus, Ohio 43210

Received August 28, 1992; accepted July 14, 1994

We provide a probabilistic framework, based on Perceptual Inference Networks, for the management of computational resources such as special purpose modules, feature detectors, and highly domain dependent algorithms. Since these resources tend to be computationally expensive and have limited applicability, judicious management is warranted. The resources are used to build a comprehensive description of the scene. Resources are selected in an information theoretic framework with the maximization of information gain per unit of computation as the optimality criterion. The viability of the algorithm is demonstrated in perceptual organization tasks. © 1995 Academic Press, Inc.

I. INTRODUCTION

The solution to the problem of computer vision is not possible using a single processing module, but instead involves the intelligent coordination and judicious management of a number of specific visual task algorithms. For example, any one shape from X algorithm cannot solve the 2D sketch problem nor can edge detection provide all the necessary information for recognition; texture and shading information are important in elucidating the underlying surface structure. Although more needs to be done, there has been significant progress in the design of efficient specific task modules solving specific visual tasks. Along with this success comes the realization that there is a trade-off among the accuracy of results, the knowledge of domain parameters, and the number of constraints one can impose. The generalization of algorithms to various domains is typically accompanied by inaccuracy and inefficiency. Thus the solution lies not in the generalization of different algorithms but in the efficient management and integration of specific visual task modules. Such paradigms are scarce in computer vision; the present work is a step in that direction. We propose a method based on probability theory to coordinate and manage various visual task modules.

The integration of visual modules that we seek differs from that usually considered in the Marr paradigm. In the Marr paradigm, integration occurs among various shapes from X modules on an algorithmic level by reformulating the constraints and the underlying theory. For a good review and a stimulating discussion of such approaches the reader is referred to [1]. The proposed mode of integration is not on the algorithmic level but involves the judicious management of existing algorithms. The approach relies on the opportunistic use of visual task modules which are not necessarily restricted to shape from X algorithms. The type of visual tasks we consider include various symmetry analyzers, region analyzers, corner detectors, shape from X, texture analyzers, road analyzers, and so on. This is not to imply that algorithmic integration is unnecessary; it can make the task easier but certainly cannot be the full solution. We believe that, although the algorithmic integration of a small number of modules is possible, such integration of all visual task modules is effectively impossible.

Specific visual task modules tend not to generalize very well to broader domains. For example, shape-from-shading algorithms require knowledge of the surface reflectance, light direction, or viewing direction. They invoke constraints such as constancy of surface albedo and illumination by a point source. This makes these algorithms inapplicable in real domains where the surface properties change from region to region. Besides, the execution time of these algorithms increases drastically with image size. Hence, we would like to apply such algorithms only at those select areas in the image where we are confident that the constraints hold and that useful new information will accrue. The paradigm presented in this paper engages probabilistic reasoning and information theory to manage such visual modules opportunistically.

Among the systems which rely on the opportunistic use of visual processes for general image interpretation tasks, some are rule-based. The CONDOR system of Strat [2, 3] has a set of computational processes which interact through a shared data structure. Each process is associated with a context set which, when satisfied, invokes the process. The final description of the image is determined using clique formation and selection. Similar knowledge-based ideas are used in the VISIONS Schema System [4]
which manages a distributed network of small special
purpose interpretation systems. Each schema is an expert
at recognizing one type of object and is built out of special-
ized knowledge sources. The communication among the
schemas is facilitated through a blackboard. McKeown
et al. [5] also use a production system knowledge base
to interpret aerial images of airports.

Another school, philosophically different from rule-
based expert systems, uses Bayesian networks. Levitt et
al. [6], in their INTACTS system, employ a control strat-
egy based on utility theory and Bayesian network infer-
ence. Candidate processing actions are selected by utility
computations based on the estimated value and the cost
of each action. The value of each action is calculated as
the expected increment in evidential value achieved at
the parent hypotheses due to the action. The choice is
biased to those actions that change the probabilities in
the network the most. Executable actions include providing
terrain support, the use of high-resolution sensors, and
actions to structure the network dynamically. Recently,
Rimey and Brown [7] used Bayesian nets and maximum
expected utility decision rules to select a sequence of
visual operations. The Bayesian nets were built on high-
level primitives such as utensils, cups, and plates. The
visual operations involved camera focus, cup detection,
napkin detection, and the like. They provide a solution
to "where to look next?", an important question in
the context of active vision. Jensen et al. [8] used context
modeling capabilities of Bayesian networks for image in-
terpretation tasks. Munck-Fairwood [9] uses a Bayesian
network for probabilistic 3D inference from 2D data.

We suggest a knowledge-based reasoning formalism
based on Bayesian networks called the Perceptual Inference
Network (PIN) in [10]. The PIN is a computational
structure used to integrate, to infer, and to manage spatial
information. In the present context, we aim to provide
the best description of a scene using the PIN which allows
us to generate hypotheses of different shapes from primi-
tive organizations. For example, in [10, 11] we used a
PIN in a perceptual organization context to generate par-
allelogram, circle, ellipse, and ribbon hypotheses from
primitive organizations such as parallels, closure, strands,
and segments.\footnote{See [12] for a comprehensive review of the work in perceptual organi-
zation.} The output set of hypotheses is large and
redundant. A set of lines is described as a parallelogram
and/or ellipse and/or circle. There is considerable ambigu-
ity in such a description. The strategy is to use special
purpose modules to resolve the ambiguous hypotheses
and generate a comprehensive description of the scene.
We probe the original gray level image for evidence to
resolve the ambiguities. Although the algorithm is demon-
strated using simple organizations, it is easily extensible
to complex shapes.

Mohan and Nevatia [13, 14] also use perceptual organi-

zation to generate scene descriptions in terms of primi-
tives like symmetric ribbons. They recognize the usefulness
of the structural relationships made explicit by perceptual organization in complex image understanding.
All reasonable feature groupings are first detected and
then the promising ones are selected by a constraint satis-
faction neural network.

The management scheme, depicted in Fig. 1, is built
around the concept of the PIN developed in [10, 11]. The
aim is to generate a hierarchical description of the scene.
This is done using two modules: the preattentive and
the attentive. The preattentive module, described in [15],
provides evidence in terms of primitive organizations like
parallelism, continuity, closure, and strands. The atten-
tive organization is done using a PIN which integrates
information and hypothesizes in the presence of noise and
occlusions. It receives information from the preattentive
module and a resource manager. The resource manager
provides evidence using a set of special purpose modules.
The management is done in an information theoretic
framework.

For the sake of completeness, we outline the theory
behind the Perceptual Inference Networks in Section II.
Section III presents the theory behind the management
scheme. Section IV briefly presents the visual modules
used in the results, which are presented in Section V. We
analyze the performance of the management scheme and
justify the use of the information theoretic principles in
Section VI.

II. AN OVERVIEW OF PINs

The PIN is a computational structure to integrate, to
infer, and to manage spatial information. The formalism
is derived from Bayesian networks and is based on proba-
bilistic arguments. The network provides a means to rep-
resent knowledge as well as acts as a computational struc-
ture for reasoning based on available evidence in light of
embedded knowledge. The formalism is developed in
greater detail in [10, 11]. In this paper we use the PIN
formalism to manage special purpose resources. Before
going into the details of the theory, we review the PIN
formalism.

A. Composite Nodes

The PIN, which is a modification of Bayesian networks,
is used to integrate information about various spatial fea-
tures to form composite hypotheses. Bayesian networks
[16] are directed acyclic graphs with nodes representing
propositions (or random variables) and arcs signifying
direct dependencies and quantified by conditional prob-
abilities. Suppose we have $m$ features $f_1, \ldots, f_m$ and $n$
locations $l_1, \ldots, l_n$. We form binary random variables,
$l_{ij}$. The event $\{l_{ij} = 1\}$ denotes that feature $f_i$ occurs at
location $l_j$. The aim is to formulate an efficient means of
updating the probabilities of these random variables based
on the evidence. Given a set of features at some locations, expectations for other features at different locations are formed; we want to devise an efficient means to do so.

The traditional Bayesian network approach would assign one Bayesian node to each random variable $l_j^i$, requiring $mn$ nodes. With $n = \mathcal{O}(N^2)$, $N$ being the image width, and with $m = \mathcal{O}(10)$ (or more), this is typically a very large number. We make the problem tractable by exploiting the conditional independencies inherent among the variables. Features which are dependent tend to be close together spatially. The dependency structure will make the connections among the random variables sparse. Besides being sparse, the network will also have a number of similar substructures because the same feature type at different locations will have the same neighborhood structure. We can group together some of the random variables representing the same type of feature at various locations. The group of random variables $l_j^i, \ldots, l_j^k$ forms a composite random variable representing the events that feature $f_j$ occurring at $l_i$ through $l_k$. The grouping will clearly reduce the number of nodes of the network, but the computational complexity of each node increases. However, this increase is far more than offset by the reduction in structural complexity. We refer to this modified form of the Bayesian net as a PIN.

Unlike a Bayesian network node which represents a single feature at a particular location, a PIN composite node (for example, Fig. 2) represents a particular feature type. Each composite node represents a collection of random variables denoting the presence of the corresponding feature at a set of locations. Thus, the common characteristic is the type of feature each random variable represents; they differ only in location. However, it may not always be possible to group all the random variables representing a given feature type into a single composite node if doing so introduces a cycle structure (ignoring the link directions) in the underlying graph of the PIN. A tree structured PIN is desirable because of the simplicity of its probability updating algorithm [16]. In cases which would introduce cycles we form multiple composite nodes for the feature type (e.g., the corner nodes $N11$, $N10$, and $N12$).
N14 in Fig. 2) or use dummy nodes (node N3 in Fig. 2) to break the cycles. (See [10, 17] for more detail.)

B. Conditional Probabilities

We must specify the conditional probability of a feature at a particular location, given the parents at compatible locations. The original formulation of message passing in Bayesian networks will suffice for this case. However, we need to check for compatible locations as the message arrives.

The specification of the conditional probability can be simplified if we assume that the conditional probability can be factored into two functions $g$ and $\text{Comp}$, one depending on the feature type, and the other being a locational compatibility function:

$$ P(l_x | l_{u_1}, \cdots, l_{u_m}) = g(x | u_1, \cdots, u_m) \text{Comp}(l_x, \{l_{u_1}, \cdots, l_{u_m}\}), \quad (1) $$

where $P$ is the conditional probability density function quantifying the link from the composite parent $U_1, \cdots, U_m$ to the composite node $X$. $l_x$ and $l_{u_i}$ denote the occurrence of the features $X$ and $U_i$ at locations $l_x$ and $l_{u_i}$, respectively. When used alone, $x$, $u_1, \cdots, u_m$ denote the values (typically binary) taken by the random variables representing the features $X$, $U_1$, ..., $U_m$, respectively.

Now $g$ is a conditional function which expresses the degree of belief in a feature, $X = x$, given the existence of other features, $U_1 = u_1, \cdots, U_m = u_m$. This, in general, will not be a valid probability function because it is multiplied by the Comp function to form the conditional probabilities. (Only if Comp is unity must $g$ be a true probability function.) For lack of a better name, we call $g$ the conditional belief function. This function is independent of the locational information and is affected only by the definitions of the features.

The form of the spatial compatibility function, $\text{Comp}$ in Eq. (1), depends on the feature sought. The range of the function should be chosen such that $P$ is a conditional probability function. We choose $\text{Comp}$ to be 1 when the locations of the features lie within a tolerance and 0 otherwise. It is effectively a spatial compatibility predicate. The belief in features at locations not compatible (i.e., Comp = 0) with the incoming message locations are unchanged. The choice of the tolerance is an open issue depending on the amount of positional error one can accept (or expects), and other forms of the function are certainly possible.

An interpretation of the chosen conditional probability form in Eq. (1) in terms of traditional probabilities is as follows. The event $l_x$, which represents the occurrence of feature $X$ at location $l_x$, can be considered to be the conjunction of two events: that a feature occurs at $l_x$ and the feature is $X$. Thus we can write $P(l_x)$ as $P(l_x | x) = P(x | l_x)P(l_x)$, where $l_x$ represent the occurrence of some feature at the location.

Using this the PIN conditional probability expression can be factored as

$$ P(l_x | l_{u_1}, \cdots, l_{u_m}) = P(x | l_x, l_{u_1}, \cdots, l_{u_m})P(l_x | l_u), \quad (2) $$

By comparing Eqs. (1) and (2), we can provide probabilistic interpretations to the factors $g$ and $\text{Comp}$. The first conditional probability $P(X = x | l_x, l_{u_1}, \cdots, l_{u_m})$ is estimated by $g$ and the second term $P(l_x | l_u)$ represents the locational compatibility function $\text{Comp}$. Our development is predicated on the assumption that this factoring of the total conditional probability is reasonable.

Factoring the conditional probability according to Eq. (1) means that instead of storing a large conditional matrix we need to store only the much smaller $g$ matrix and the Comp function definition. As for the other belief parameters ($\pi$, $\lambda$, and $\text{BEL}$), instead of storing one set, the composite node keeps lists of belief parameters for each location separately. Metaphorically, this forms a sort of "file drawer" of potential locations with each file containing the current belief in each (of those having nonzero belief). Since the beliefs in features at locations not compatible with a particular location are not changed, we can decompose the node into two modules. The first module computes spatial compatibility and chooses the belief parameters of the compatible features for updating. The updating is done in the second module by a single computational resource [16]. This computational resource is time shared by the features at different locations and is shielded from the network by the compatibility computation module, unlike previous formulations of Bayesian networks.

C. An Example PIN

An example PIN is shown pictorially in Fig. 2. This network has developed (manually) for perceptual organization and is designed to make inferences about salient structures such as parallelograms, circles, ellipses, and ribbons. The usefulness of the PIN is not restricted to such simple organizations and can be used with more complex shapes. However, for illustrative purposes, we will deal with simple shapes.

The network has 23 nodes, each representing a feature type, as depicted in the figure. Node N23 denotes the concept of a parallelogram, which is formed from the concept of a trapezoid and the attribute that the "nonpar-
allel" sides of the trapezoid are parallel. Trapezoid, N21, is a quadrilateral with a pair of sides parallel, represented by nodes N19 and N20 respectively. A quadrilateral is a polygon, node N15, having four sides. A polygon with three sides is a triangle, node N16. An equilateral triangle, node N18, is a triangle with appropriate symmetry, node N17. The features of closure, node N3, and corners at particular locations, node N14, help us to form polygonal hypotheses as it does to form closed geometrical features bounded by curves only, node N4, or by curves and straight lines, node N9. The features of ellipse, node N6, and circles, node N8, are formed from appropriate symmetry, nodes N5 and N7, respectively.

The detection of a closed token will generate expectations about other features according to their locational compatibility. Detecting a rectangle (with some probability) raises expectations for parallel lines and corners. The confirmation or rejection of the corner or parallel hypotheses will update the probabilities of the rectangle accordingly. Thus, the network integrates multiple sources of evidence.

1. Location Compatibility Functions, Comp. As we saw before, each node of the perceptual inference network has a location compatibility function, Comp, defined to calculate the compatibility of a message from a child or parent with its own location set (see Eq. (1)). Compatibility is ascertained based on a proximity tolerance. For the experiments this is 5 pixels. The choice of this value depends on the spatial resolution of the system (at least).

Computing compatible locations for some nodes is pretty obvious, like the triangle, the trapezoid, or the parallelogram nodes. We just match the endpoints. For nodes like the ellipse (N6) and the circle (N8) we consider the best fitting ellipse or circle, respectively. Nodes N4, N9, and N15 deserve special attention. For a message received at N4 from N6, we segment the ellipse at those points where the latus rectum intersects the boundary. This generates the (approximately) constant curvature segments for node N4. To generate compatible locations for spatial information from node N3 to N4, N9, and N15, we segment each closed chain of pixels into constant curvature parts to generate the appropriate set of compatible arcs or straight lines. This problem differs from that solved in the literature, where there is one continuous contour. We have a set of fragmented curves. The details of how this is performed can be found in [10]. To compute compatible locations for the quadrilateral node (N19), we consider the four largest lines of the polygon as constituting the quadrilateral boundary, for messages coming from node N15. Similarly, we consider the three largest sides of the polygon for the triangle node.

Once we have computed the compatible locations at a node in response to a message from its neighbors, we store pointer links both at the destination and at the source of the message. This avoids recomputing compatible locations for the same feature in subsequent updating. Storing pointer links also speeds up information calculations (discussed later) and subsequent updating.

2. Conditional Belief Functions, g. The conditional belief function, g (see Eq. (1)), captures the belief in the specified feature type, given the status of the corresponding parent nodes. The conditional probabilities of the underlying Bayesian network are partly characterized by g and partly by the locational compatibility function, Comp, discussed above. We consider the following general form for the conditional belief functions,

$$g(X = 1 \mid u_1, \ldots, u_n) = C_{(u_1, \ldots, u_n)}(1 - e^{-f_{\text{geom}}})$$

(3)

where $X$ is the composite node under consideration, $u_1, \ldots, u_n$ are the values taken by the binary random variables represented by $X$'s parent composite nodes, and $C_{(u_1, \ldots, u_n)}$ is a multiplier dependent on the state of the parents. We found from experimentation that the absolute value of the multiplier is less important than the relative values of the multiplier for different parent states. The second factor captures the geometric information in the primitive. The exponential is chosen to limit this factor to the interval $(0, 1)$. $f_{\text{geom}}$ is a factor based on the size of the features and some defining relations, like parallelism for a parallelogram. Qualitatively speaking, the larger the feature, or the more strictly it satisfies the defining relation of a feature type, the larger the value of the belief. Other functional forms with this qualitative behavior may also suffice.

In the present implementation $f_{\text{geom}}$ is calculated in the following manner. For the nodes representing some form of closed convex figure, that is N3, N4, N9, N13, N15, N16, N18, and N19, we use the form $f_{\text{geom}} = (\Pi_{i=1}^n l_i^m l_{aw}^n (\delta_{\text{convex}}), where the l_i's are the respective lengths of the n constituting segments of the closed figure, $l_{aw}$ is a normalizing constant and is chosen to be the average length of segments in the image, and $\delta_{\text{convex}}$ is 1 if the figure is convex and 0 if not. The geometric mean in the numerator is chosen to suppress the spurious generation of hypotheses like a quadrilateral with one side whose length is 0, which we noted were formed with an arithmetic average. The trapezoid node, N21, and the parallelogram node, N23, have the same expression for $f_{\text{geom}}$ as that above, with extra multiplicative terms of the form $\cos^2(\theta_{\text{diff}})$, where $\theta_{\text{diff}}$ is the angular difference between lines, to penalize for nonparallelism in the constituting sides. For the ellipse node, N6, we choose $f_{\text{geom}} = |b/a - 1| \sqrt{\text{Area}/l_{aw}}$, where b and a are the minor and major axes, respectively. The circle node, N8, has $f_{\text{geom}} = \sqrt{\text{Area}/l_{aw}}$.

D. Prior Probabilities

The prior probabilities, for this example, are the prior probabilities of occurrence of the root node features like
strands, constant curvature segments, corners, and symmetry. There may be domains where the occurrence of, say, corners is very low; this knowledge can be very concisely incorporated in the reasoning process through the assignment of estimates of prior probabilities. In the absence of any prior knowledge we assume equal priors, a solution which maximizes the entropy of the distribution. This is the best we can do in such a situation and is the assumption in all of the experiments thus far.

E. Evidence Instantiation

The pieces of evidence are the organizations detected in a preliminary, or preattentive, phase. These include strands, constant curvature segments, parallelograms, and ribbons. Each piece of evidence activates the nodes belonging to the corresponding feature type. This evidence is virtual in the sense that it conveys a graded degree of belief about the underlying features. This belief can be measured in terms of the photometric and geometric evidence we have for the feature. For a closed boundary we set the message equal to the ratio of the total length of the segments to the length of the perimeter of the hypothesized closed boundary. For parallels it is the fraction of overlap. Constant curvature segments are assigned confidence measures according to the straight line or arc fit error they exhibit.

F. PIN specification

While the structure of the PIN reflects the dependencies among the organization types, the conditional probabilities quantify these relationships. The structure and the conditional probabilities of the PIN used in the experiments of this paper were specified manually based on the following heuristics. The direct dependencies among groupings were represented by links going from groupings exhibiting less organization to groupings having more organization; the grouping with less organization is a part of the grouping with more organization. Also, the distance (in terms of links) between two nodes of the PIN representing two geometric figures was kept low if the groupings were similar. For example, the rectangle node and the trapezoid node are closer in the PIN than are the rectangle node and the circle node. This ensures that the size of the PIN remains small. The probability models were set using the approach outlined in Section II.C.2.

We take this opportunity to mention that in another work [17] we formalize these heuristics and offer an autonomous algorithm for complete PIN construction and specification. Given the Random Parametric Structural Descriptions (RPSDs) [18–20] of a set of target organizations of interest, and given the RPSDs of a set of elementary structures for which we can provide direct evidence, the algorithm not only automatically structures the PIN but also estimates the conditional probabilities. The expected size of the automatically designed PIN is linearly dependent on the number of target nodes and it is always guaranteed to be tree structured. We found the design process to be robust against minor perturbations of the input RPSD specifications.

III. THEORY

The PIN outlined in the previous section allows us to integrate and make inferences from multiple sources of information such as parallelism, corners, closure, and so on. The primary information is integrated to form composite organized hypotheses like parallelograms, circles, ellipses, or ribbons. The integration of multiple sources of information is achieved in a probabilistic framework which is also computationally efficient. In this section we investigate ways of refining the hypotheses in a PIN by the intelligent use of special purpose modules. The PIN framework is very amenable to controlling a battery of special purpose modules. We can associate each module with a node(s) of the PIN as providing evidence for it. For example, a corner analysis module can be associated with those nodes representing corners (N10, N11, N14), in Fig. 2.

The use of special purpose modules is not rare in computer vision. There are a number of algorithms which are designed with specific tasks in mind. For example, in domains where circles are the objects of interest, the use of a special purpose circle detector is warranted. Or one might have a special purpose road-marking detector. These special purpose tasks tend to be computationally expensive and have limited applicability. Therefore, it is highly desirable that we use them judiciously. We ideally want to apply them only when and where we expect the highest return.

The norm for “highest return” is a matter of choice. In keeping with the probabilistic approach we chose an entropy-based (or information theoretic) measure. We want to use that special purpose module which gives us the greatest degree of entropy reduction per unit computational resource. The measure of computational resource for a special purpose module involves the computational time, the use of special purpose hardware, amount of storage, and so on. Thus, the aim is to reduce the uncertainty in the final hypothesis set to the greatest extent while using the least possible amount of computational resources.

A. Control Algorithm

The goal is to produce the best description of the image using a set of salient organizations. For example, we might...
be interested in parallelograms, quadrilaterals, ellipses, circles, and ribbons. The set of Bayesian nodes representing these structures of interest forms the target set: N23, N19, N6, N8, and N13. The special purpose modules will be associated with the composite nodes representing the structures that they are designed to test for. For illustration, consider that we have special purpose modules for corners, contours, and regions associated with composite nodes N10, N11, N14, N2, and N3.

The algorithm is depicted in Fig. 3. Conflict arises when we have more than one organizational hypothesis for the same location in the image and the associated probabilities are nearly equal. Thus, there is an ambiguity in the description. For example, we might describe a set of edges as an ellipse, a rectangle, or a ribbon. If the probabilities of the respective hypotheses are very different then it is easy to resolve such a conflict; that is not true if they are close. Let \( p_1 \) and \( p_2 \) be the respective probabilities of two competing hypotheses. For these experiments, if \( p_1 < 2p_2 \) and \( p_2 < 2p_1 \) then they are declared to be close together. Other schemes are possible, but this seems to work pretty well.

We apply the computational resources to minimize the size of the conflict(s) as described below. The size of a conflict is determined as follows. First, we form a locational conflict graph whose nodes denote the target hypotheses and links denote conflicts among them. We detect such conflicts by looking for locational overlap and nearly equal probabilities, as above. If there is a locational conflict between two hypotheses and the probabilities are not close (decided as above) then the node representing the hypothesis having the lower probability is removed from further consideration.

The locational conflict graph is next broken into connected components. The size of a conflict is determined by the size of the associated component. We will apply the special purpose computational modules until the components of the conflict graph are below a specified size. The maximum tolerable size depends on the final conflict resolution scheme one chooses. For example, our final conflict resolution scheme involves clique detection, which is NP complete. Therefore, large components would make this method impractical; we set a maximum component size of 5.

### B. Information Theoretic Formalism

In this section we present the means by which the special purpose computational modules are used to reduce the uncertainty in the hypothesis set. Suppose we have a set of target hypotheses \( T_1, \ldots, T_n \), in locational conflict. This conflict will be resolved by using each appropriate module to provide additional evidence to the corresponding network sensory nodes. Let the Bayesian nodes in these sensory composite nodes be \( X_1, \ldots, X_m \).

The total uncertainty in hypothesis \( T_i \) can be measured by the entropy \( H(T_i) \). The uncertainty in \( T_i \), given knowledge of the sensory node \( X_j \), is \( H(T_i | X_j) \). The expected entropy reduction with knowledge of \( X_j \), or the mutual information between \( T_i \) and \( X_j \), is

\[
I(T_i, X_j) = H(T_i) - H(T_i | X_j).
\]  

(4)

Substituting standard expressions for the entropy in terms of the discrete probabilities of the events, we have

\[
I(T_i, X_j) = \sum_{t_i=x_i} \sum_{x_j} P(t_i, x_j) \log \frac{P(t_i, x_j)}{P(t_i) P(x_j)}.
\]  

(5)

Note that the mutual information is itself the relative entropy between the joint distribution and the product of the marginals.

#### 1. Computation

The mutual information can be calculated in a PIN by observing that \( P(t_i, x_j) \) is actually the notation \( \text{BEL}(t_i, x_j) \) in the context of a Bayesian network and that \( \text{BEL}(t_i, x_j) = \text{BEL}(x_j | t_i) \text{BEL}(t_i) \). Thus to calculate the mutual information between \( T_i \) and \( X_j \) we need \( \text{BEL}(t_i), \text{BEL}(x_j), \) and \( \text{BEL}(x_j | t_i) \). The values \( \text{BEL}(t_i) \)

\footnote{The BEL notation is used to be consistent with Pearl [16]. BEL\((t_i, x_j) = P(t_i, x_j)\), the Bayesian probability.}
and $\text{BEL}(x_i)$ are present at the nodes of the PIN in equilibrium. The quantity $\text{BEL}(x_i|t_j)$ can be calculated by temporary message passing in the same PIN as follows. We temporarily instantiate node $T_i$ through its range and for each value, $t_j$, note the value of the final $\text{BEL}(X_i = x_i)$. Although the temporary message passing scheme seems to involve a lot of overhead by instantiating $T_i$ to each of its states, this is not so. We need not perform locational matching at each composite node to determine the relevant Bayesian node because we use the pointer links between nodes, discussed earlier, for this purpose. As the reader might recall, we store the links between the nodes involved in message passing whenever we find a locational match. These links are just traced to perform the temporary updating instead of doing a full locational match.

2. Management. Using the method outlined above we calculate the mutual information between each pair, $T_i$ and $X_j$. This forms the $n \times m$ information matrix $\text{IMAT}$, where $\text{IMAT}(i,j) = I(T_i, X_j)$. We form the column sums $I_{\text{EXP}}(X_j) = \sum_{i=1}^n I(T_i, X_j)$, each of which is the total sensor mutual information for a given sensor with respect to the complete set of hypotheses in conflict. The sensor having the largest value for this is expected to provide the largest amount of information toward resolving the ambiguity. Associated with each sensory node is a computational module. Each module has a characteristic computational expense, which may include the time of execution or the amount of hardware needed. Let the computational expense to provide evidence for $X_j$ be $\tau_j$.

We have considered two resource management strategies. The first strategy assumes that we have a time constraint; we must complete the computation within an interval $T$. The optimal choice in this case can be decided by solving the problem

$$\begin{align*}
\text{maximize} & \quad \sum_{j=1}^m I_{\text{EXP}}(X_j) \delta_j \\
\text{subject to} & \quad \sum_{j=1}^m \tau_j \delta_j < T,
\end{align*}$$

where $\delta_j = 1$ if the resource is applied to investigate $X_j$ and 0 otherwise. The idea is to maximize the total information gain within the time constraint. This is the strategy used by Levitt et al. [6] but with a different utility function. Our aim is to reduce the ambiguity as far as possible. Thus, we adopt a different strategy based on conflict resolution which does not impose a maximum computation time constraint.

The second (adopted) strategy applies resources until all conflicts have been resolved, or we have exhausted all available resources. The sensory nodes $X_1, \ldots, X_m$ are ranked based on the expected information gain per unit computational resource, $I_{\text{EXP}}(X_j)/\tau_j$. For example, suppose we have three special purpose modules: corner, edge segment, and region analysis. Further suppose that the region analysis module takes three times as long to execute as does the corner module, and that the segment analysis module takes twice the time of the corner module. In this case, we divide the expected information $I_{\text{EXP}}$ of the sensory nodes corresponding to the corner nodes by 1, those of the segment modules by 2, and those of the region module by 3. Thus, the final ranking of the sensory nodes will be based on the expected information gain of the PIN per unit computational time. We then apply the special purpose modules, in order, only at those locations dictated by the nodes and update the PIN with the corresponding information. This updating is again computationally efficient because we use the pointer links at the nodes, as mentioned before.

C. Return Condition

Figure 3 offers two return options. The first option is to apply the special purpose modules to find evidence for all the sensory nodes $X_1, \ldots, X_m$ to resolve conflict in a particular component. The second option applies only a top few of the ranked special purpose modules and then recomputes the locational conflict graph. We use the first option; the second is computationally expensive in a sequential implementation. However, in a parallel implementation the second option might be more appropriate.

D. Final Hypothesis Resolution

After applying the special purpose modules to reduce the component sizes we expect to be left with a sparse locational conflict graph. The actual effect on the locational conflict graph depends on the information provided by the special purpose modules and depends on the degree of "surprise" they provide. At any rate, this will represent the best we can do with the available computational resources. Other sensor modalities might be warranted if the locational conflict graph is still dense.

For each component of the final locational conflict graph we form the locational consistency graph, which is its complement. In the locational consistency graph we look for the most significant clique. The significance of the clique is not determined by the number of nodes, but the sum of the individual hypothesis probabilities. In case of a tie, the size of the clique is the determining factor; we choose the one which has the larger number of nodes. This constitutes the consistent set of hypotheses for each component. The total set of largest cliques (one from each component) forms the description of the scene. Note that this automatically includes those features not in conflict to begin with.

Using the same algorithm, we can construct a set of descriptions of decreasing importance from the locational

---

6 Note that this is not Fisher's information matrix from detection and estimation theory.
conflict graph. To form such a set we rank the cliques in the locational consistency graph according to the sum of probabilities of the individual nodes. The importance of a ranked set of descriptions is enormous in the context of object recognition. Parts of objects missing in first description might be present in the next best description. This suggests a form of the celebrated "principle of least commitment."

IV. SPECIAL PURPOSE MODULES

Before presenting the results, we describe the special purpose modules used in these experiments. The call to a module provides it a location and its output is an updated location and a confidence measure. We consider three special purpose modules in this work: corner, edge segment, and region analysis. The algorithms represented here are rather simple and may readily be replaced by more sophisticated ones. These are used just to illustrate the viability of the overall scheme.

A. Corner Analysis Module

The corner analysis module investigates a square region around the location of interest. The size of the square region is determined by the size of edge operator used to detect the initial edge segments. The coarser the edge operator used, the less accurately the edge points will be located. For the optimal zero crossing operator (OZCO) edge detector [23] the expected edge pixel migration is bounded by 3γ [24], where γ is the scale of the operator. Thus, the region is a 6γ × 6γ square centered at the location of interest. At each point in the region we compute

\[
\text{CORNERITY}(i, j) = \|V(i, j)\| \|\nabla (\text{Grad}_{in}(I(i, j)))\|,
\]

where I(i, j) is the image gray level value at (i, j) and Grad_{in} is the gradient direction. This is the Kitchen and Rosenfeld cornerity measure [25]. The updated location is where the measure is maximized. The mean (μ), standard deviation (σ), and the maximum value (max) of the measure in the region are calculated. The updated confidence, \( \text{BEL}(1) \), is

\[
\text{BEL}(1) = \begin{cases} 
1 - \frac{\sigma^2}{(\text{max} - \mu)^2} & \text{max} \neq \mu \\
1 & \text{max} = \mu.
\end{cases}
\]

The confidence increases for low background noise and for strong (in terms of the gray level) corners. In this case, σ of the measure is low. Also, the measure would be high near the corner and low elsewhere and, consequently, \( \mu \) is low. Thus, we have a large confidence in the corner. The behavior is intuitively satisfactory.

![Segment and the area analyzed to provide evidence for it.](image)

Fig. 4. Segment and the area analyzed to provide evidence for it.

We consider the computational cost, \( \tau_j \), to be the size of the region analyzed, 36γ².

B. Segment Analysis Module

This module investigates the presence of an edge segment, straight or curved, and returns an updated location of the segment, together with a confidence measure. A strip centered at the queried location is investigated (see Fig. 4). The width of the strip is 6γ, based on the previous arguments (cf, corner analysis module discussion). The strip is computed as follows. The local tangent direction of the (parametrized) curve is estimated using

\[
\tan(\psi) = \frac{\dot{y}(s) \ast \hat{g}(s)}{\dot{x}(s) \ast \hat{g}(s)},
\]

where \( \ast \) denotes convolution, \( x(s) \) and \( y(s) \) are the coordinates of the curve, and \( \dot{g}(s) \) is the first derivative of a Gaussian.\(^7\) The Gaussian smoothing is needed to regularize the differentiation. We walk perpendicular to the local tangent, \( \theta \), to a distance of 3γ on either side, computing the gradient of the image function as we go. We then adjust the edge location to the maximum of the gradient magnitude profile. The mean (μ), standard deviation (σ), and the maximum (max) of all the investigated locations in the strip are calculated. The confidence in the presence of the edge segment is computed as

\[
\text{BEL}(1) = \begin{cases} 
1 - \frac{\sigma^2}{(\text{max} - \mu)^2} & \text{max} \neq \mu \\
1 & \text{max} = \mu.
\end{cases}
\]

The justification for this expression is similar to that of the corner detector. The area investigated is proportional to the length of the segment. Thus the complexity of the algorithm is \( \mathcal{O}(N) \). The computational cost, \( \tau_j \), is the area investigated, 6γL, with \( L \) being the length of the segment.

\(^7\) The scale of the Gaussian used for smoothing is 2.0. The choice of this parameter is not crucial.
C. Region Analysis Module

The region analysis module accepts as input a boundary description and analyzes it to determine the presence of a planar gray level enclosed region. The output is an updated region boundary and a confidence measure. The first step of the algorithm is to call the segment analysis module to update the boundary and give a confidence measure, \( \text{BEL}_\text{bound}(1) \). The region inside the updated boundary is then analyzed for a planar fit to the gray levels. To determine if a point is inside the boundary, we shift the boundary coordinates so that the origin is the point under consideration and walk around the boundary to identify crossings with the positive \( x' \) axis (see Fig. 5). A tangent to the \( x \) axis is not counted as a crossing. If the number of crossings is odd then the point under consideration is inside the region; otherwise it is outside. Note that we need to do this only for points inside the upright bounding rectangle of the region.

We compute the least-squares planar fit to the gray levels in the region as follows. Let a plane be characterized as

\[
g_p(x_i, y_i) = \theta_1 + \theta_2 x_i + \theta_3 y_i,\]

where \((x_i, y_i)\) are the coordinates of the point inside the region of interest, \(g_p\) is the height of the fitted surface, and the \(\theta_i\) are the parameters of the surface. Let the total number of points in the region be \(n\) and the image gray level at \((x_i, y_i)\) be denoted by \(g(x_i, y_i)\). The total squared error, \(E(\Theta)\), is

\[
E(\Theta) = \sum_{i}(g(x_i, y_i) - (\theta_1 + \theta_2 x_i + \theta_3 y_i))^2 = (G - Z\Theta)^T(G - Z\Theta), \tag{11}
\]

where \(G = [g(x_1, y_1), \ldots, g(x_i, y_i), \ldots, g(x_n, y_n)]^T\) is the vector of all the gray level values (observations), \(\Theta = [\theta_1, \theta_2, \theta_3]^T\) is the vector of parameters, and

\[
\hat{\Theta} = (Z^T Z)^{-1} Z^T G, \tag{12}
\]

where \(Z^T\) is the transpose of \(Z\), \(Z^T Z\) is a \(3 \times 3\) matrix given by

\[
Z^T Z = \begin{bmatrix} n & \sum_i x_i & \sum_i y_i \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i y_i \\ \sum_i y_i & \sum_i x_i y_i & \sum_i y_i^2 \end{bmatrix} \tag{13}
\]

and

\[
Z^T G = \begin{bmatrix} \sum_i g(x_i, y_i) \\ \sum_i x_i g(x_i, y_i) \\ \sum_i y_i g(x_i, y_i) \end{bmatrix} \tag{14}
\]

Substituting Eq. (12) into Eq. (11), we get the minimum planar fit error, \(E_{\min}\):

\[
E_{\min} = G^T G - (Z^T G)^T \hat{\Theta}. \tag{15}
\]

The smaller the fit error, the greater the confidence in a homogeneous region. We update the confidence as

\[
\text{BEL}(1) = \text{BEL}_\text{bound}(1)(e^{-E_{\min} / \sigma_0}), \tag{16}
\]

where \(\text{BEL}_\text{bound}(1)\) is the confidence in the boundary of the region (Eq. (10)) and \(\sigma_0\) is the value of the significant contrast parameter. For the experiments we consider contrasts of values greater than 25 (\(\sigma_0 = 25\)) to be significant. To estimate the computational cost of region analysis we consider the following. The work to calculate the fit error is constant for each interior point. Let the length of the boundary be \(N\). Then number of points inside the bounding box is \(\mathcal{O}(N^2)\). Thus, the total complexity is \(\mathcal{O}(N^3)\). We take the cube of the length of the boundary to be a good estimate of the computational cost of region analysis.
V. RESULTS

The visual task modules presented in the previous section were managed using the algorithm presented earlier. The perceptual inference network is implemented in C++ and runs on Sun Sparc-ELC machines. The voting and graph theoretic preattentive methods [15] provide the preliminary evidence to the PIN. The evidence is integrated by the PIN to construct composite hypotheses. These hypotheses are then analyzed and conflicts among them resolved using the special purpose modules in accordance with information theoretic criteria as described. The final hypotheses are selected using the algorithm based on clique detection, as above.

In this section we present results on two real images, one a simple blocks world scene and the other an aerial scene. In the next we discuss the performance of the management scheme and justify the use of the underlying information theoretic principles.

A. Blocks World

The gray level image\(^4\) is shown in Fig. 6a. The edges were detected using the OZCO presented in [23] at a scale, \(\gamma\), of 1.5. The edge contours were segmented into constant curvature segments using a modified form of Wuescher and Boyer’s algorithm [26]. The constant curvature segments are shown in Fig. 6b. Note the boundary of the top of the brighter block in the middle of the image is not segmented into four straight line segments, but into two straight lines and an arc. This is not a drawback of the segmentation algorithm, but is an example of the types of imperfections present in all low-level algorithms. We will see that the (deep) knowledge-based perceptual organization system is not misled by such faults.

Preattentive organization, implemented as a voting module followed by a graph theoretic module, detects organizations in a purely bottom up fashion from the edge segments. Some of the organizations detected are closed boundary hypotheses, parallels, strands of edge segments, and parallelograms. The closure groupings are shown in Fig. 6c, the open strands of edge segments in Fig. 6d, the parallels (including nearly concentric arcs) in Fig. 6e, and the parallelograms in Fig. 6f. These are introduced as evidence to the perceptual inference network. The preattentive evidence consists of six closed figures, 18 strands, eight ribs, 22 parallels, and 69 segments. Note that not all the parallelograms are detected, and the set of closed edge segments is not complete. The pieces of evidence are assigned probabilities based on photometric characteristics as discussed in [15].

After the network settles to equilibrium we have various organization hypotheses with associated probabilities. Since it is impractical to give a complete listing of the node probabilities, we show some of the organizations as images. Figure 7 depicts a set of such organizations. Note how the set of closed boundary hypotheses is more complete than in preattentive organization. Some parallelograms which were not detected before are now hypothesized, especially the top face of the bright block in the middle. The set of corners predicted is also shown overlaid on the original gray level image.

The target nodes are identified as N23 (parallelograms), N19 (quadrilaterals), N13 (ribbons), N6 (ellipses), and N8 (circles). We seek a description of the scene in terms of these organizations. After introduction of the preattentive evidence the PIN hypothesized 30 parallelograms, 30 quadrilaterals, seven ribbons, 36 ellipses, and 41 circles. We next construct the locational conflict graph which has 44 links forming 11 connected components. We compute the information matrix using the temporary message passing discussed before and use that to rank the needed sensory evidence that the special purpose modules can supply, namely, for corners (N10, N11, N14), segments (N2), and regions (N3). Based on the information matrix the PIN requested evidence on 47 (out of 197) corners, 12 (out of 56) regions, and 7 (out of 120) segments. Note that the special purpose modules were applied to provide evidence for about 15% of the hypothesized features. This represents a considerable savings, especially versus blindly applying the computational modules over the entire image. The requested locations are shown in Fig. 8. Note how the special purpose modules are used opportunistically and at selected places. For this example, we provide all the requested evidence. However, in a real situation the special purpose modules will provide evidence based on the ranking until the conflicts are resolved.

After the introduction of all the evidence the conflict graph had 43 links forming 11 components and its structure did not change greatly. However, there were some links added and some which were deleted. That the graph did not change much is not a drawback of the algorithm but a consequence of the fact that the special purpose modules did not provide much unexpected information in this particular case. The final resolved hypotheses, including those not in conflict to begin with, are shown in Fig. 9a. The overlay of the description on the initial edge image is shown in Fig. 9b. The final description obtained is very reasonable. The top horizontal bar is extracted as a parallelogram as are the dark areas on top. The second best description is shown in Fig. 9b. This represents some of the less probable descriptions of the scene and would be considered (as a part a complete vision system) if the best description does not serve the purpose. These descriptions provide a basis for constructing complex scene descriptions.

B. Aerial Image

The next result is presented on the aerial image shown in Fig. 10a. The edges detected are shown in Fig. 10b. As

\(^4\) Image courtesy of Dr. A. Etemadi of the University of Surrey.
FIG. 6. Organizations detected by the preattentive perceptual organization module: (a) Gray level image, (b) edge contours detected, (c) closed boundary hypotheses, (d) edge strand hypotheses, (e) parallel edge segments, (f) parallelogram hypotheses detected as intersections of parallel strips.

before, we seek a description in terms of parallelograms, quadrilaterals, ellipses, circles, and ribbons. The preattentive modules provided evidence for 47 closed figures, nine strands, 12 ribbons, 16 parallels, and 103 segments. On the introduction of this evidence into the PIN, we had 26 parallelograms, 26 quadrilaterals, 11 ribbons, 53 circles, and 58 ellipse hypotheses. The parallelograms, ellipses, circles, and ribbons are shown in Figs. 10c–10f, respectively. Based on the conflict graph (68 links forming 19 components) and information theoretic analysis, the
FIG. 7. Organizations detected using the attentive module on the blocks image: (a) Closed boundaries, (b) parallelograms, (c) ellipses, (d) circles, (e) ribbons (no change from the preattentive hypotheses), (f) corners.

system analyzed 18 out of 67 regions, 4 out of 147 segments, and 82 out of 249 corners, about 20% of the hypothesized features. This represents a significant reduction in computation time, again especially when compared to blindly applying the resources over the whole image. After the introduction of the new evidence we had 68 links in the locational conflict graph forming 19 components. The best and the second best sets of resolved hypotheses are
shown in Fig. 11. Note how the roads are described fairly well by parallelograms and ribbons. The descriptions are, again, very reasonable.

C. Timing Considerations

The entire algorithm is implemented in C++ and runs on Sun Sparc-ELC workstations. Before presenting the timing results we would like to add a disclaimer that the implementation is not optimal and there are ample opportunities for performance improvement by improved coding. However, to give the reader an idea of the computations involved we provide the performance we have achieved to date. The timings are depicted in Fig. 12. The algorithm has been divided into seven parts: the introduction of initial evidence, computation of the locational conflict graph, search for connected components, information matrix computation, resource application, introduction of new evidence, and final conflict resolution. From the table we see that a significant amount of the time is taken up by the special purpose modules. This is not surprising. As pointed out earlier, a special purpose modules tend to be computationally expensive and that is precisely what motivates much of this work. From the data above we
seek evidence for an average of about 15% of the initial hypotheses, representing a potential sevenfold time savings. Yet, again, when compared to blindly applying the resources over the whole image, the savings are even more substantial.

The entire conflict resolution scheme takes a few minutes on a sequential implementation. In a parallel implementation we would get a further speedup. To get a rough estimate we can divide the timing by 23, the number of nodes in the PIN. Thus, we can see that this approach is not computationally burdensome.

VI. PERFORMANCE EVALUATION

In the previous section we demonstrated the performance of the proposed resource management scheme on real images. Now we provide further experimental evidence to validate the underlying theory. First, we examine the use of the expected net gain in information (Section III. B.2) as a criterion to rank the computational resources. We want to see if the observed change in entropy (information) due to a particular process is commensurate with the expected change in entropy; we expect a high correlation between process rankings based on expected changes in entropy and those based on the observed changes. Second, we look for a correlation between the observed and expected changes in entropy of a given target hypothesis after applying the set of computational resources.

Before going into the evaluation methodology, we mention some caveats to consider while interpreting the results. Note that we are trying to compare expected values for the entropy change with the changes observed in a particular realization of the random experiment. Therefore, it is quite unlikely that the observed value will equal the expected value. However, they ought to be close. A more complete investigation would consider an ensemble average over a large number of images and perhaps even a variety of PIN designs looking for a range of features. It may also be noted that although the expected reduction in entropy $I(T, X)$ is always positive, the observed reduction in entropy can be negative. That is, the entropy might increase, instead of decreasing, after the application of a computational resource. This will happen if the new evidence contradicts the already accumulated knowledge, or if the new evidence from two or more sources is contradictory.

An increase in the entropy for a particular feature is not a failure in the theory or implementation, but simply one possible outcome of the experiment. It is entirely possible that introducing new evidence increases the ambiguity of a given situation. For example, suppose we detect four line segments arranged in a near rectangular configuration, but which do not intersect in corners. This collection of line segments seems to indicate the presence of a rectangle; thus the system would invoke a corner detector at the supposed corner locations in an attempt to confirm the rectangle hypothesis. If little photometric evidence for the corners is found, however, our confidence in the rectangle is diminished, and we are less certain as to the feature type (if any true organization
FIG. 10. Organizations detected using the attentive module on the aerial image: (a) Gray level, (b) edges detected in the aerial image, (c) set of parallelograms hypothesized, (d) ellipses predicted, (e) circles, (f) ribbons (no change from the preattentive hypotheses).

actually exists at all) than before we applied the corner detectors. If the corners are, in fact, not there, then we cannot reliably declare the presence of a rectangle. Knowing that the corners are missing is an important and useful piece of information. Further examination of the image (if other processes are available) is necessary to resolve the question. If no other processes are available, then our system has said all it can about the image in this case.
A. Methodology

As we saw in Section III. B.2, to rank the computational resources we construct the mutual information matrix \( IMAT(i, j) = I(T_i, X_j) \), where \( T_i \) is the \( i \)th target hypothesis in conflict and \( X_j \) is the \( j \)th sensory hypothesis. The ranking criterion is the column sum \( (I_{\text{Exp}}(X_j)) \) representing the total sensor mutual information for a given sensor with respect to the complete set of target hypotheses in conflict. \( I_{\text{Exp}}(X_j) \) is also the expected total reduction in entropy of all the target hypotheses given knowledge about the sensor \( X_j \). Applying a computational resource provides evidence about a sensory hypothesis which, in turn, changes the probabilities of the target hypotheses. We will compare the observed changes in entropy of the target hypotheses with the expectations given by \( I_{\text{Exp}}(X_j) \); the observed changes should be generally proportional to the expected values.

The row sums \( (I_{\text{Exp}}(T_i)) \) of IMAT represent the expected reduction in the entropy of each target hypothesis given knowledge about all the sensory hypotheses. We will also study the relationship of these expected entropy reductions with the observations.

B. Results

We present results on the two images considered in the previous section: the aerial and block world images.

1. Resource Ordering. Figure 13 presents the results on the aerial image. Figure 13a shows the plot of \( I_{\text{Exp}}(X_j) \), the net expected change in entropy over all the target hypotheses, for evidence about each sensory hypothesis. With three special processors, and about 33 locations to examine with each (on average), we generate a bit over 100 "sensors" whose entropy we wish to reduce. This establishes the horizontal axis of the plots. Figure 13b is the plot of the resultant change in entropy based on the observed evidence for a sensory hypothesis from a computational resource. As mentioned earlier, the observed changes in entropy are both positive and negative. This is also true for the entropy changes of the sensory hypotheses after applying the computational resources, as seen in Fig. 13c.

Of more importance than the precise degree of entropy reduction offered by a resource is the correctness of the process ordering. From Figs. 13a and 13b it is clear that the resource which was expected to provide the most information did reduce the entropies to the greatest extent. To quantify the overall similarity of the ordering in

<table>
<thead>
<tr>
<th>Processing Stage</th>
<th>Blocks Image</th>
<th>Aerial Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Evidence Introduction</td>
<td>3:30 min</td>
<td>2:54 min</td>
</tr>
<tr>
<td>Locational Graph Comp.</td>
<td>7 s</td>
<td>10 s</td>
</tr>
<tr>
<td>Connected Components</td>
<td>0.2 s</td>
<td>0.1 s</td>
</tr>
<tr>
<td>Information Matrix</td>
<td>15 s</td>
<td>16 s</td>
</tr>
<tr>
<td>Special Resources</td>
<td>4:32 min</td>
<td>3:01 min</td>
</tr>
<tr>
<td>New Evidence Introduction</td>
<td>40 s</td>
<td>53 s</td>
</tr>
<tr>
<td>Final Conflict Resolution</td>
<td>0.3 s</td>
<td>0.1 s</td>
</tr>
<tr>
<td>TOTAL</td>
<td>9:04.5 min</td>
<td>7:14.2 min</td>
</tr>
</tbody>
</table>

FIG. 12. Times of the computational resource management scheme on two images.
Figs. 13a and 13b, we use the function

\[ O_{\text{sim}}(k) = \sum_{i=1}^{k} \frac{I_{\text{EXP}}(X_i) \Delta H(X_i)}{||I_{\text{EXP}}(X_i)|| \cdot ||\Delta H(X_i)||}, \quad (17) \]

where \( \Delta H(X_i) \) is the sum of the absolute changes in the entropies of the target hypotheses after the application of the resource corresponding to the sensory hypothesis \( X_i \) and \( || \cdot || \) represents the norm. \( O_{\text{sim}}(k) \) is a measure of the similarity of the rankings of \( I_{\text{EXP}}(X_i) \) and \( \Delta H(X_i) \) for the first \( k \) sensory hypotheses in the ordered list. The maximum possible value is 1 and the minimum is 0. The plot of \( O_{\text{sim}}(k) \) for \( k = 1 \) to the total number of sensory hypotheses is shown in Fig. 13d. The plot suggests that the theoretical ordering of the resources correlates well with the observed change in entropy. Note that the similarity decreases as we consider the resources further down in the expected mutual information rankings. This suggests a lower limit for the expected net mutual information, below which it is not significant as a ranking statistic. (For our experiments, this limit was set to 0.0, the minimum possible.)

Figure 14 shows the corresponding plots for the blocks world image. The similarity of the expected and actual change in entropy is lower than that for the aerial image. However, the qualitative behavior is similar.

In conclusion we can say that the ranking of our resources based on the chosen information theoretic criterion is supported by, at least, the two experiments that we have presented. Clearly, additional work is warranted in this area, including exploring the lower limit of the expected net mutual information as a ranking statistic.
Other resource allocation strategies, especially for large numbers of resource options, should be investigated as well. Such topics lie beyond the scope of this paper.

2. Net Reduction in Entropy. Another interesting study is the observed reduction in the entropy of a target hypothesis \( \Delta H(T_i) \) on applying the set of computational resources, and its correlation with the expected change given by the row sum \( I_{exp}(T_i) \) of IMAT. As before, we found the observed change in entropy to be both positive and negative. Thus, for the comparison we considered the absolute value of the observed change. We quantified the similarity using the normalized correlation measure: 
\[
\Sigma_i \left( I_{exp}(T_i) \Delta H(T_i) \right) / \left( \|I_{exp}\| \|\Delta H\| \right).
\]

The value of this similarity index ranges from 0 to 1. For the aerial image this similarity is 0.7362 and for the blocks world it is higher at 0.9818.

\[\text{FIG. 14. For the blocks world scene: (a) Expected total mutual information (entropy reduction) for each sensor with respect to the set of target hypotheses in conflict. (b) Observed total entropy reduction over all the target hypotheses on the application of the resource corresponding to each sensory hypothesis. (c) Observed entropy reduction of the sensory hypotheses. (d) Plot of the similarity } (O_{out}(k)) \text{ between the ordering based on the expected information and that suggested by the observed entropy reductions.}\]

VII. CONCLUSION

In this paper we presented an information theoretic framework to manage special purpose visual task modules. The framework uses the perceptual inference network formalism to do probabilistic reasoning among hypotheses. The hypotheses were resolved using special purpose modules invoked according to the potential information content per unit of computation. The complete framework was demonstrated on simple perceptual organization tasks. The generalization to more complex shapes and more complex computational modules is currently under investigation. Also under study is the issue of a more flexible definition of spatial conflict, and a corresponding resolution strategy, to allow interleaved shapes and other such situations to handled more satisfactorily.
REFERENCES


