UNCERTAINTY REASONING IN HIERARCHICAL VISUAL EVIDENCE SPACE

by

Jianzhong Qian

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

APPROVED:

Roger W. Ehrich

Roger W. Ehrich, Chairman

Richard W. Conners

Richard W. Conners

James B Campbell

James B. Campbell

John W. Roach

John W. Roach

Ronald J. Pieper

Ronald J. Pieper

April 1990

Blacksburg, Virginia

UNCERTAINTY REASONING IN HIERARCHICAL VISUAL EVIDENCE SPACE

by

Jianzhong Qian
Roger W. Ehrich, Chairman
Electrical Engineering

(ABSTRACT)

One of the major problems in computer vision involves dealing with uncertain information. Occlusion, dissimilar views, insufficient illumination, insufficient resolution, and degradation give rise to imprecise data. At the same time, incomplete or local knowledge of the scene gives rise to imprecise interpretation rules. Uncertainty arises at different processing levels of computer vision either because of the imprecise data or because of the imprecise interpretation rules. It is natural to build computer vision systems that incorporate uncertainty reasoning. The Dempster-Shafer (D-S) theory of evidence is appealing for coping with uncertainty hierarchically. However, very little work has been done to apply D-S theory to practical vision systems because some important problems are yet to be resolved.

In this dissertation, a computational framework is presented to show how Dempster-Shafer evidence theory can be applied to a hierarchically structured hypothesis space in a computer vision application. Based upon *a priori* knowledge, uncertain visual information is transformed from one level to another through stages of evidence collection and mapping, hypothesis generation, evidence interaction, and hypothesis verification. The system reasons about significant perceptual features through both top-down and bottom-up active processes. It is shown how to make use of partial and locally ambiguous information at different levels of abstraction to achieve reliable interpretation. It is also shown how the reasoning process can make use of spatial relationships among pieces of visual evidence to strengthen the reasoning results. Some theoretical problems, which arise from the adoption of the Dempster-Shafer model as the paradigm for a computer vision system, have been resolved. Methods for implementing this algorithm are presented for applications in object recognition and image understanding. Experiments on some applications are given to demonstrate the merit of the framework.

ACKNOWLEDGEMENTS

The author sincerely wishes to thank Dr. Roger W. Ehrich for his guidance and suggestions, his time and patience, and his encouragement and continuous financial support throughout the author's whole Ph.D. program. The author gratefully appreciates the interdisciplinary study and research environment provided by Dr. Ehrich, in which the author benefited by professional training in both Electrical Engineering and Computer Science. This work is the result of four years of hard work which includes countless discussions and interactive work sessions with Dr. Ehrich.

The author would like to thank Dr. Campbell for his support, especially for his suggestions and help in the development of the DNESYS system which is one of important applications in this research work. The author also would like to thank Dr. Conners for his advice and help in this research work at the Spatial Data Analysis Laboratory. The author gives thanks to Dr. Roach and Dr. Pieper for their inputs and comments on this work. The valuable help from all my committee members shall always be appreciated.

Also, the author would like to thank Dr. Robert M. Haralick who helped the author to begin his research and graduate study in this university. The work experience with Dr. Haralick was very valuable to the author.

The author deeply thanks his wife Xingming and his daughter Xiaolan to whom this work is dedicated. This work would not have been possible without their continuous encouragement, understanding, and moral support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter I
INTRODUCTION

The task of computer vision researchers is to endow computer with human-like visual capabilities. This task involves understanding complex visual processes and building intelligent computer vision systems. Object recognition is a critical part of this task. Rosenfeld (1986) claims that the goal of a computer vision system is to detect and locate objects of a given type (either 2-D or 3-D) in images.

The scientific problems posed by computer vision are very important since vision is the most powerful sensory modality for any intelligent system which interacts with the outside physical world. Much research work has been done throughout the past thirty years for industrial, military, office, laboratory, and social applications. These problems are still extremely challenging, although almost no one realized that computer vision was so difficult in early 1960's. This misunderstanding arises because we humans are ourselves so good at vision (Marr, 1982). As a branch of science, computer vision is young and immature (Haralick, 1985).

Since one of the earliest papers explored a solution to the task of recognizing 3-D objects (Roberts, 1965), many computer systems have been proposed to solve this task. They can be classified into two main categories: object recognition from intensity images and object recognition from depth maps. The former can be further divided into two groups: the viewpoint-dependent system which aimed at the reconstruction of depth information from the 2-D image first, and the viewpoint-independent system which performs its vision task without bottom-up depth reconstruction under the assumption that viewpoints are invariant over wide ranges. The successes of expert systems (rule-based systems, knowledge-based systems), which are characterized by the use of significant amounts of non-procedural knowledge to assist in data interpretation in some domain, have motivated computer researchers to develop interesting vision systems in that domain. In some systems, a static collection of rules and a dynamic database of facts are maintained, and the inference engine utilizes a set of production rules. Researchers are trying to improve this traditional basic paradigm by organizing rules into a hierarchy and generating the inference engine in various ways. However, the approaches often used are to "think up something which might work and try it out" (Besl and Jain, 1984). Although some of them have achieved limited success, we are far away from achieving success in general computer vision. We feel that much further research work needs to be done in this field, especially in the "intelligence" aspects.

The reason for emphasizing the intelligence aspects of computer vision is based upon the following facts: the eye is just a sensor; the visual cortex of human brain is our primary organ of vision. It turns out that perception is not the simple result of analyzing a set of stimulus patterns, but rather a best interpretation of sensory data based upon prior knowledge. By using knowledge, humans may infer many properties of a visual scene that may not be directly supported by the visual data. The senses play only a role that provides evidence for checking hypotheses about the nature of our surrounding environment. Thus, vision involves mobilizing knowledge and expectations about the environment and the objects in it. There are many micro-decision making processes at the different levels in perception. That relates research in computer vision to some research areas of AI in which large amounts of problem-specific knowledge are used to obtain constrained solutions. We may say that increasing "intelligence" of a computer vision system is one of the hopeful directions in computer vision research field. The goal may be achieved both by the use of knowledge and by reasoning with uncertainty.

One of the major differences between a computer vision system and an ordinary AI system is that the former must analyze raw image data. It is a formidable challenge to apply AI and its logical, symbolic world to computer vision and to force it to confront a noisy, uncertain, real world. Because of missing data, occlusion, and many forms of image degradation, the amount of available information in the raw data may be limited. The viewpoints may be unknown. The knowledge about the raw image may be incomplete. Uncertainty is caused either by the imprecise data or by the imprecise interpretation rules (which is a set of ordered and organized knowledge). Due to distortion and dissimilar views, some objects do not always present look the same. Since none of the currently existing low-level image processing operators is perfect, some important features are not extracted and erroneous features are detected. As a result, evidence from the observed data are often incomplete or may conflict, and the rules used in the vision system are often just intuitively accepted. In addition, uncertainty also arises from statistical information available from an inadequate training set. There are two cases. One is that the feature distribution in such a training set may only coarsely characterize feature space. The other is that the *a priori* probability of seeing a particular object at random in the training set is a highly unreliable estimate of seeing that object in the real world (Lehrer, Reynolds, and Griffith, 1987). The lack of capability to deal with the uncertainty and vagaries may be the main problem which limits the capability of computer vision systems.

Researchers believe that one of important abilities of human vision is handling uncertain information through a process of perceptual grouping, evidence gathering and reasoning based upon prior knowledge. It can make use of partial and locally ambiguous information to achieve reliable identifications. This is done by allowing interpolations through data gaps and extrapolations to be made to new situations for which data are not available. Based upon this observation, perceptions can be treated as hypotheses, specifically, as "perceptual hypotheses" (Gregory, 1980). According to Gregory, hypotheses are effective in having powers to predict future events, unsensed characteristics, and further hypotheses (the structure of the perceptual hypotheses can be hierarchical). They can be generated by observation and induction and can be confirmed or disconfirmed, though not with logical or probabilistic certainty.

It seems that the evidence and hypotheses in a visual process are hierarchical in nature. There are two possible paths from the lowest level to the highest level in the hierarchy. One path is significant points ... significant lines...shapes...objects...scene. Another path is significant points...significant patches... regions (or surfaces)...objects...scene. The perceptual grouping process in this hierarchy is so fast for humans that we are not aware of its complexity. The hypotheses generated at each level are supported by the evidence gathered from that level and from the levels below. The hypotheses verified at the lower levels can be the new evidence for the hypothesis at the higher level. In computer vision, evidence begins with the raw image from which ambiguous features are extracted. The multiple hypotheses about a visual event in turn accumulate partial evidence from the raw data up to the final decision level and support different interpretations of the scene. According to Marr's Principle of Least Commitment (Marr, 1982), evidence from different sources at different abstract levels should be accumulated to sufficient degree before making binding decisions.

Also, contextual information from one processing level can be treated as evidence for hypotheses formed at the same level. Context plays an important role in general vision. Since most scenes contain some easily identified components, these then provide a great deal of information that can greatly ease the recognition of more difficult components of the scene. The use of contextual information means the use of multiple evidence for a specific hypothesis. Evidential reasoning provides an opportunity to incorporate the most helpful contextual information to speed up the recognition process.

The evidence theory proposed by Dempster and Shafer (D-S model) (Shafer, 1976) suggests how evidence concerning hierarchically–related hypotheses might be combined consistently to allow uncertain reasoning at any level of abstraction. It provides a way for handling the aggregation of evidence gathered at varying levels of detail or specificity. Because the D-S approach allows one to attribute belief to subsets, as well as to individual elements of the hypothesis set, it may be similar to the perceptual grouping and aggregation process at varying levels of abstraction in human vision. Although the probability judgments made in the D-S model can usually be adapted to a Bayesian argument, such argument models both the hypothesis and the evidence for that hypothesis by requiring prior probabilities for the hypothesis and conditional probabilities for the evidence given the hypothesis. The problem is that such information is not usually available. Shafer argues that the D-S model is just a generalization of the Bayesian model (Shafer, 1986). The advantage of using the D-S model over the Bayesian model is the ability to use less complete probabilistic models. Almost all real applications of uncertainty reasoning are implemented either by modifying the pure theory or by restricting application cases. Certainly, a new method for adapting the D-S model to the reasoning process in a hierarchical evidence space for computer vision is needed.

Therefore, instead of using deterministic methods such as those that have been used in many proposed vision systems, a reasoning system that can handle uncertain information and represent and manipulate incomplete and imperfect knowledge, should be built into a computer vision system. That will allow systems to behave in a more "intelligent" manner. Although applications of uncertainty reasoning in expert systems have became popular, the uses of uncertainty reasoning are just beginning (Wesley, 1986; Andress and Kak, 1988.) In most cases, uncertainty reasoning is used only as an aid at a particular processing level. There has been no significant success in using it as a unified method throughout the entire computer vision system.

We believe that if a unified framework for uncertainty reasoning algorithms could be developed for hierarchical visual recognition at multiple levels of abstraction, it would provide bidirectional active vision through both bottom-up and top-down processes. If such a framework were adopted, then low level processing would be more intelligent and more reliable, and high level processing would be more powerful and more flexible. Consequently, many improvements and new methods could be developed at each processing level. In this dissertation, such a framework for uncertainty reasoning in computer vision systems is proposed.

The second chapter of this dissertation briefly reviews some typical vision systems which are relevant to this research work and comparatively examines the mathematical tools of uncertainty reasoning. Other researchers' efforts to incorporate uncertainty reasoning into computer vision systems are evaluated in that chapter. Chapter III details the ways of efficient implementation of D-S model in a hierarchical evidence space. Some controversial issues about the D-S model are discussed and corresponding solutions in the context of computer vision are given here.

As a base of the reasoning system, Chapter IV defines a hierarchical visual event space for the uncertain reasoning framework. Associated with the visual event space, a hierarchical visual evidence space and a hierarchical hypothesis space is also introduced. The relationships between these spaces are defined. The correspondences between these spaces and the D-S model are developed.

A reasoning system to adapt the D-S model to computer vision is proposed in Chapter V. We direct the particular attention to reasoning in the hierarchical visual evidence space while keeping the computational complexity under control. It begins with methods of visual evidence collection, evidence mapping, and initial *b*pa assignment. Then, the interactions of hierarchical visual evidence are described according to the D-S model. Finally, an algorithm using the D-S model for handling spatial relations is presented. These three processes are all based on the visual event hierarchy, the visual evidence hierarchy, the visual hypothesis hierarchy, and the reasoning tree defined in Chapter IV.

The system implementation is presented in Chapter VI. A paradigm, which consists hypothesis generation and selection, filling in missing information (or information-gap filling), and hypothesis verification, is given. The database, the knowledge base, and the focus of attention for the hierarchical uncertainty reasoning are described. Finally, some applications of this uncertainty reasoning system for computer vision are presented in Chapter VII. These experimental results show the merits and promises of this framework.

Chapter II

LITERATURE REVIEW

This chapter will review some of the literature in the areas of computer vision systems, uncertainty reasoning theories, and uncertainty reasoning in computer vision. This review will focus only on those topics that are relevant to this proposed research. It will provide some insights into the following questions: where we are, what kind of challenges we have to meet, and the scope of this work.

2.1 Vision Systems Related to This Research

This section is not intended as an overview of all general purpose computer vision systems, but rather a view of a selection of those relevant to this research work. A survey of 3-D object recognition using gray level images can be found in (Haralick, 1985; Rosenfeld, 1984; Brady, 1982; and Barrow and Tenenbaum, 1981). A survey of model-based gray level image analysis systems can be found in (Binford, 1982). A survey of vision systems, which emphasizes the recognition from depth maps or range images, is found in (Besl and Jain, 1984). There has been a great deal of activity in this area in recent years. A survey of vision systems in which AI played an important role can be found in (Shapiro, 1985). A review of more recent developments in computer vision systems can be found in (Brady, 1987; Grimson, 1987). In this section we will briefly review several vision systems that involve issues related to this research work.

Roberts (1965) explored a solution to problem of recognizing polyhedral objects by solving exactly for viewpoint and object parameters and by matching topologically equivalent points. Despite its lack of robustness, this early algorithm has been considered as fundamental work in the field.

Since 1974, Hanson and Riseman *et al.* have been designing a complex computer vision system, VISIONS (Visual Integrating by Semantic Interpretation of Natural Scenes) (Riseman and Hanson 1978, 1984, 1989) that is an ongoing long-term project. The system operates on multi-spectral images of outdoor scenes and interprets the scenes based upon low-level segmentations. The most attractive aspect of this system is the use of hierarchical structures in the model building process. This process consists of four main components:

1. the process that construct the multi-level knowledge structure in forms of Short Term Memory (STM) for the model and Long Term Memory (LTM) for the world,

2. the processes that transform image data between different levels of representation,

3. the hierarchical control strategies, and

4. the tree search mechanism that deals with the hierarchical structures of knowledge and data.

Nagao *et al.* proposed a system to analyze complex aerial photographs (Nagao *et al.*, 1980). This is basically a "segment-extract-decide" system. In this system, an aerial photograph is first segmented into regions by several low-level processes. Using dominant features of each extracted region, procedures to recognize that region are applied based upon the extraction of specialized features. One of the main features of this system is the way in which contextual information is used to resolve the ambiguities in some regions according to specific structural relations which are preorganized as rules in the system. The structural relations applied by the system, however, are fixed. As a result, the interpretations depend upon assumptions that are image-oriented and are not broadly useful.

The use of symbolic reasoning to aid in analyzing scenes is the main feature of the ACRONYM system (Brooks, 1981). ACRONYM was the first computer vision system to employ a general symbolic constraint manipulation system to determine if a set of constraints is inconsistent. The basic paradigm in this system is top-down and model-driven. For reasoning efficiency, the system constructs five graphs.

1. Object Graph. All object models are stored as nodes in this graph. Each node is a frame that contains the object description.

2. Prediction Graph. This graph is produced by the knowledge from the object models and the feedback from the low-level processes. The nodes of this graph represent the predictions of specific image features.

3. Restriction Graph. The nodes of this graph are attached to the nodes and arcs of Prediction Graph. Each node in this graph contains the constraint parameters which are provided by the measurements on the image features.

4. Observation Graph. This graph is constructed under the direction of Prediction Graph. All observables (ribbons and ellipses) and their relationships are stored in this graph.

5. Interpretation Graph. This graph is for matching the Prediction Graph against the Object Graph based upon the Observation Graph and Restriction Graph.

Invoking these graphs, ACRONYM performs symbolic reasoning in a top-down manner. The final version of the Interpretation Graph represents a match of a maximal subgraph of the Prediction Graph and gives the global interpretation of the image.

The top-down paradigm is only one part of the ACRONYM system, and since the top-down paradigm is inadequate for complex scenes, it limits ACRONYM's utility as a general vision system. In fact, interpretations are limited in this system. It has been tried only on a single viewpoint and with only a few objects such as Boeing-747 aircraft. However, the significance of ACRONYM is that it provides a general methodology for intelligent use of models, constraints, and feedback in a computer vision system.

Few vision systems whose expertise is used to reason about shape completion and network connection from an edge map and its corresponding intensity image have been published. However, some systems whose functions are partially related to this task have been proposed in recent years. Nazif and Levine (1984) proposed an expert system which uses rules to segment an image into uniform regions and connected lines using general knowledge about low-level properties of processes. This system can both split and merge regions and either add, delete, or join lines.

Their algorithm is to segment an intensity image rather than to reason about an edge or a line map. Line analysis is combined with region analysis, since they depend upon each other. Therefore, the system does not take care of shape completion which is mostly based upon an edge map under object oriented guidance. As a result, the linking criteria for line analysis are independent of the class of scene under analysis. Usually there are not enough rules to handle all data configurations. In the system, rules have the following format:

Condition 1 AND Condition 2 AND ... AND Condition N, ACTIONS.

If one of these conditions fails, then the rule fails. This format makes it difficult to handle the partial failure condition and the conflict condition which often occur in real images.

In contrast to Nazif's system, McKeown and Pane (1984) proposed a system named ALIGN to align fragmented linear features (region like) to a connected composite region using domain specific knowledge.

ALIGN does not reason about edges and lines. The criteria in ALIGN are used only for aligning roads or road-like region segments. ALIGN is difficult to adapt to other domains, and as the authors point out, several problems have not been solved. One is that neither intersections nor forks are considered in that system. Consequently, it can not reason about complex object contours or complex networks. There are no alternatives which may satisfy the criteria at the same time. It has only a weak ability to cope with the multicriterion optimization problem.

There are many other vision systems, including Shirai's system (Shirai, 1978) which uses obvious edges as cues for interpreting the entire scene; Ballard's system (Ballard *et al.*, 1978) which uses an image model and geometric constraints for locating ships at docks and in locating ribs in chest x-rays; the SIGMA system (Davis and Hwang, 1985) which attempts to integrate both bottom-up and top-down processes into a single flexible reasoning process; Faugeras' system (Faugeras and

Hebert, 1986) which uses a hierarchical representation of visual features to recognize objects while locating them; Ben-Arie's system (Ben-Arie and Meiri, 1987) which applies the optimal matching search of a multinary relations graph to recognize 3-D aircraft from a 2-D image; and Lowe's system (Lowe, 1987) which can recognize 3-D objects from unknown viewpoints in single gray-tone images.

In spite of the use of different algorithms and different applications, the common problem in the systems referenced above is that none of them has the ability to cope with uncertainty information and none of them does uncertainty reasoning. This may be a main deficiency that limits their use and applicability. In fact, most of these systems only use a few models and only "know" a few objects. They were successful in their restricted scene domains with few object classes.

2.2 Uncertainty Reasoning

Uncertainty reasoning is a collection of inference techniques for reasoning with uncertain information. It involves basically three primary aspects: the representation of uncertainty information, the combination of uncertain evidence, and the propagation of uncertainty through the reasoning process. This is a problem of broad interest and considerable difficulty. To cope with real world problems, one must deal with uncertainty which appears in the following forms:

1. the information is partial
2. the information is not fully reliable
3. the representation is imprecise
4. the information from different sources conflicts.

Recently some theories of uncertainty reasoning have been proposed. Among them, the Bayesian update theory (Duda, Hart, and Nilsson, 1976; Pearl, 1986), Dempster-Shafer evidence theory (Shafer, 1976, 1986; Gordon and Shortliffe, 1985), Shortliffe's Certainty Factor (Shortliffe and Buchanan, 1975; Shortliffe, 1976; Buchanan and Shortliffe, 1984), and Zadeh's fuzzy logic theory (Zadeh, 1965, 1975, 1986) are four of the most prominent theories. A survey paper on this topic can be found in (Bhatnagov and Kanal, 1986; Henkind and Harrison, 1988; Dubois and Prade, 1985, 1988). We will review these issues briefly and compare them.

2.2.1 The Bayesian Update Model

The traditional numerical approach to uncertainty reasoning is the Bayesian probability model (Duda *et al.*, 1976, 1979; Pednault, 1981; Charniak, 1983; and Pearl, 1986). Historically, many interpretations of probability have existed, including

1. probability as relative frequency in the long run
2. probability as a measure of subjective belief. Different applications require different definitions of the probability (Carnap, 1950). In the probability theory for reasoning under uncertainty, the probability is usually defined as the degree of confirmation or the measure of subjective belief.

Let H=$\{h_1, h_2, ...\}$ be a set of hypotheses known to be mutually exclusive and exhaustive. Let E=$\{e_1, e_2, ...\}$ be a set of evidence. For a given hypothesis h_i, the prior probability $P(h_i)$ is defined as

$$P(h_i) \longrightarrow [0,1] \qquad \forall h_i \in H$$

$$\sum_{h_i \subseteq H} P(h_i) = 1$$

The conditional probability $P(h|e)$ is the probability of hypothesis h supported by evidence e. Duda, Hart, and Nilsson (1976) consider the problem of updating the probability of a hypothesis h with a prior probability $P(h_i)$ when new evidence e_j is obtained for which the conditional probabilities $P(e_j|h_i)$ and $P(e_j|\bar{h}_i)$ are known. By the conditional independence assumptions

$$P(e_1, e_2, ...e_m|h_i) = \prod_{j=1}^{m} P(e_j|h_i) \qquad (2.1.a)$$

$$P(e_1, e_2, ...e_m|\bar{h}_i) = \prod_{j=1}^{m} P(e_j|\bar{h}_i), \qquad (2.1.b)$$

they use the odds-likelihood form of Bayes's theorem and give an updating formula for the odds on h_i in terms of a product of likelihood ratios

$$\frac{P(h_i|e_1, e_2, ..., e_m)}{P(\bar{h}_i|e_1, e_2, ..., e_m)} = \frac{P(h_i)}{P(\bar{h}_i)} \prod_{j=1}^{m} \frac{P(e_j|h_i)}{P(e_j|\bar{h}_i)}. \qquad (2.2)$$

Charniak (1983) proposed another approximation method to estimate the hypothesis h_i. For given a set of evidence $E = \{e_1, e_2, ...\}$, the conditional probabilities

$$P(h_i|e_1, e_2, ..., e_n)$$

can be calculated by Bayes Theorem

$$
\begin{aligned}
P(h_i|e_1,e_2,...,e_n) &= \frac{P(h_i)P(e_1,e_2,...,e_n|h_i)}{P(e_1,e_2,...,e_n)} \\
&= \frac{P(h_i)P(e_1,e_2,...,e_n|h_i)}{\sum_{i=1}^{K} P(h_i)P(e_1,e_2,...,e_n|h_i)}
\end{aligned}
\tag{2.3}
$$

.

To use this formula, two independence assumptions must be made. The independence assumptions allow us to break the joint probabilities into combinations of individual terms. That is

$$
\begin{aligned}
P(h_i|e_1,e_2,...,e_n) &= \frac{P(h_i)P(e_1|h_i)...P(e_n|h_i)}{P(e_1)P(e_2)...P(e_n)} \\
&= \frac{P(h_i)P(e_1|h_i)}{P(e_1)}\frac{P(e_2|h_i)}{P(e_2)}...\frac{P(e_n|h_i)}{P(e_n)}.
\end{aligned}
\tag{2.4}
$$

This formula tells us that with initial probability estimate $P(h_i)$ for a hypothesis, we can calculate the degree of belief for the hypothesis with the support from new evidence e_j by multiplying this estimate by

$$\frac{P(e_j|h_i)}{P(e_j).}$$

However, the assumption that $P(e_i,e_j) = P(e_i)P(e_j)$ is very restrictive and is hard to satisfy in the real applications. It often occurs that two pieces of evidence tend to arise together for a single hypothesis. Charniak analyzed this and determined that violations of this assumption would affect all evidence probability estimates by a same factor since it is used only to determine the denominator of the combination formula. It turns out that the estimated value becomes greater than 1 and gives only relative rankings.

Pearl (1986) shows how uncertainty reasoning can be conducted in a hierarchical hypothesis space using a Bayesian model. Suppose $H = \{h_1, h_2, \ldots, h_n\}$ is a set of hypotheses which are mutually exclusive and exhaustive. Each subset of H which has semantic interest forms a new hypothesis. Pearl's algorithm begins with an initial probability assignment for each singleton hypothesis h_i, such that

$$\sum_{i=1}^{n} P(h_i) = 1 \tag{2.5}$$

where $P(h_i)$ denotes the probability that h_i is true given previous evidence. If S is an event that is a conjunction of some other hypotheses, then the probability of this event is the sum of probabilities of its constituents:

$$P(S) = \sum_{h_1 \in S} P(h_i). \tag{2.6}$$

Pearl uses the likelihood ratio λ_s to represent the degree to which new evidence e confirms or disconfirms S. That is,

$$\lambda_s = \frac{P(e|S)}{P(e|S^c)}. \tag{2.7}$$

Confirmation is expressed by $\lambda_s > 1$; disconfirmation by $\lambda_s < 1$. A 3-step belief update process is given by his algorithm (see Figure 1).

Pearl's algorithm provides a convenient framework for updating belief values in a hierarchical hypothesis space. However, it has been pointed out (Lee, 1988) that his algorithm still can not distinguish between uncertainty or lack of evidence and

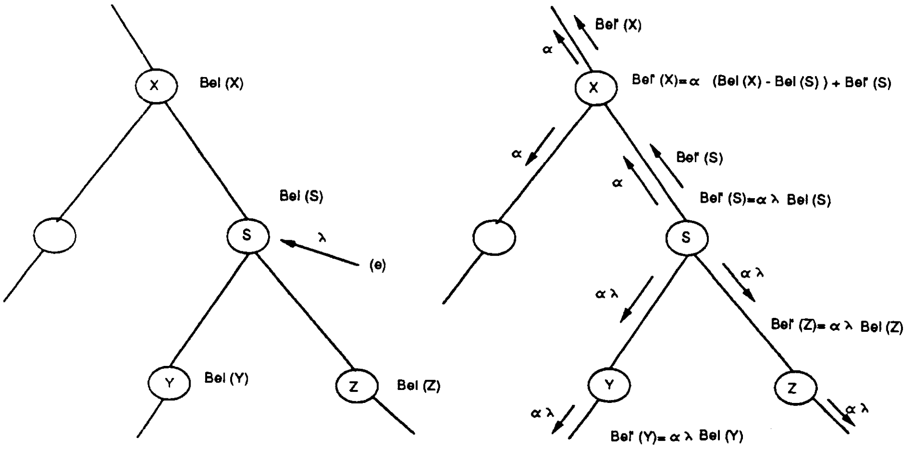


Figure 1. An illustraction of Pearl's algorithm.

equal certainty. Also, in Pearl's formulation there is no way to express ignorance about a hypothesis.

Because of the restrictions of the Bayesian model, there are several problems in applying this model to uncertainty reasoning. First, the assumptions of conditional independence are usually difficult to satisfy in real applications. As a result, bias is introduced into the system. Second, the Bayesian model fails to distinguish between uncertainty, or lack of knowledge, and equal certainty. It usually treats two unknown events by assuming equal prior probabilities even though there may be no basis for doing so. Third, the Bayesian model assumes that belief in H is equivalent to $P(H)$ so that the belief in $\bar{H}$ is $1 - P(H)$. In fact, belief functions are not additive in general. Evidence partially in favor of H does not necessarily give information about its negation. Finally, the subjective probability assignments in uncertainty reasoning cannot maintain consistency in the Bayesian model.

2.2.2 Certainty Factor

Shortliffe proposed the certainty factor in MYCIN's intuitive model of uncertainty reasoning (Shortliffe, 1976). MYCIN used *Belief MB* and *Disbelief MD* as the units of measurement. They are defined as

$$MB(h,e) = \begin{cases} 1 & \text{if } P(h) = 1, \\ \frac{max(P(h|e),P(h))-P(h)}{max(1,0)-P(h)} & \text{otherwise} \end{cases} \qquad (2.8.a)$$

$$MD(h,e) = \begin{cases} 1 & \text{if } P(h) = 0, \\ \frac{min(P(h|e),P(h))-P(h)}{min(1,0)-P(h)} & \text{otherwise} \end{cases} \qquad (2.8.b)$$

where $P(h|e)$ is the conditional probability of h given e and $P(h)$ is the *a priori* belief in h. The ideas behind these formulations is that for a piece of new evidence e

$$\begin{cases} Belief\ increased & \text{if } P(h|e) > P(h) \\ Belief\ decreased & \text{if } P(h|e) < P(h) \\ Belief\ not\ effected & \text{if } P(h|e) = P(h) \end{cases}$$

The certainty factor CF combines MB and MD into a single number. That is

$$CF(h,e) = MB(h,e) - MD(h,e) \tag{2.9}$$

Generally speaking, suppose we want to accumulate two partial supports corresponding to evidence e_1 and e_2 in the presence of other evidence e_p. Let $CF(h,e,e_p)$ be a certainty factor that represents the hypothesis h given the current evidence e and previous evidence e_p. Let f be a combination scheme. Then

$$CF(h, e_1e_2, e_p) = f(CF(h, e_1, e_p)CF(h, e_2, e_1e_p)). \tag{2.10.a}$$

For the case of more than two pieces of evidence, CF can be iteratively extended in the same way:

$$\begin{aligned} CF(h,(e_1e_2)e_3,e_p) &= f(CF(h,e_1e_2,e_p), CF(h,e_3,e_1e_2e_p)) \\ &= f(f(CF(h,e_1,e_p), CF(h,e_2,e_1e_p)), (CF(h,e_3,e_2e_1e_p))) \end{aligned} \tag{2.10.b}$$

and so on until e_n (Cheng *et al.*, 1986).

There are four schemes in MYCIN to combine multiple evidence. Here only a typical one is described. Let $\{e_1, e_2, \dots e_N\}$ be a set of evidence. Let $E_1 = e_1$, let

$E_2 = e_1e_2$ be the presence of e_2 given the previous evidence E_1, let $E_3 = e_1e_2e_3 = E_2e_3$ be the presence of e_3 given the previous evidence E_2, and so on. Finally, we have

$$E_N = e_1e_2 \dots e_N = E_{N-1}e_N. \tag{2.11}$$

Thus, the combination scheme for MB and MD based upon the set of evidence $\{e_1e_2e_3 \dots e_N\}$ is

$$\begin{aligned} MB(h, E_1) &= MB(h, e_1e_2) \\ &= MB(h, e_1) + MB(h, e_2) \times (1 - MB(h, e_1)) \end{aligned}$$

$$\begin{aligned} MB(h, E_2) &= MB(h, E_1e_3) \\ &= MB(h, E_1) + MB(h, e_3) \times (1 - MB(h, E_1)) \end{aligned}$$

and so on, until finally we have

$$\begin{aligned} MB(h, E_N) &= MB(h, E_{N-1}e_N) \\ &= MB(h, E_{N-1}) + MB(h, e_N) \times (1 - MB(h, E_{N-1})). \end{aligned} \tag{2.12}$$

Thus, all evidence is evaluated, and so are the MD. The total certainty factor, which represents the degree to which all evidence E_t supports the hypothesis h, is

$$CF(h, E_t) = MB(h, E_t) - MD(h, E_t) \tag{2.13}$$

where $0 \leq MB \leq 1$, $0 \leq MD \leq 1$, and $-1 \leq CF \leq 1$. A positive CF denotes that the hypothesis h is confirmed to a certain degree. A negative CF denotes that the hypothesis h is disconfirmed to a certain degree. A zero valued CF suggests that the hypothesis h is independent of the evidence.

The certainty factor is simple and easy to use in uncertainty reasoning. It was successful in diagnosing diseases in MYCIN. The basic weak points in the CF model are

1. it is an *ad hoc* technique and therefore lacks mathematical foundation.
2. it is good only for a singleton hypothesis space. It is incapable of consistent management of evidence bearing upon hierarchical hypotheses.

2.2.3 Dempster-Shafer Theory of Evidence

Based upon Dempster's upper and lower probabilities (Dempster, 1967), Shafer proposed a mathematical theory of evidence (the D-S model) (Shafer, 1976). The theory suggests a coherent approach for the representation and manipulation of incomplete and imperfect knowledge in a hierarchically structured hypothesis space. As a developing theory, the D-S model has attracted much interest and has sparked considerable debate among statisticians and knowledge engineers (Hummel and Landy, 1988). Recently, attempts have been made to apply D-S model to expert systems (Garvey, 1987). Before examining the D-S model, some relevant notation and definitions should be reviewed first.

Definition 2.1 **Frame of Discernment :** A frame of discernment Θ in the D-S model is defined as the set of all different possibilities, which are mutually exclusive and exhaustive, under consideration.

Each such possibility is an element of Θ. The set of all subsets of Θ is denoted by 2^{θ}. Each subset of Θ corresponds to a hypothesis. For example, if $\Theta = \{a, b, c\}$, then $\{a, b\}$ is the hypothesis that either event a or event b has occurred. The evidence is represented as a basic probability assignment (*bpa*) over the hypotheses discerned by the frame Θ.

Definition 2.2 **Basic Probability Assignment :** A basic probability assignment, denoted *bpa*, is a generalization of a probability mass distribution (or m-function) which represents the impact of each distinct piece of evidence on the subsets of a frame Θ.

A *bpa* is denoted by m and must satisfy the following conditions:

$$(1)\, 0 \leq m(A) \leq 1\; for\ all\ A\ \subseteq \Theta$$

$$(2)\, m(\phi) = 0$$

$$(3)\ \textstyle\sum_{A \subseteq \theta} m(A) = 1$$

For a subset A of the frame of discernment Θ, we can calculate the belief function, the plausibility function, uncertainty, and the combination of belief functions. These quantitatively characterize the hypothesis A.

Definition 2.3 **Belief Function :** A belief function for A, denoted $Bel(A)$, is the sum of the beliefs committed exactly to every subset of A by m.

That is

$$Bel(A) = \sum_{B \subseteq A} m(B), \tag{2.14}$$

where A is any subset of Θ and $Bel(A)$ measures the degree of belief in A.

Definition 2.4 **Focal element :** A subset A of the frame Θ is called a focal element of a belief function Bel over Θ if $m(A) > 0$.

Definition 2.5 **Simple Support Function :** A belief function is called a simple support function if it has at most one focal element not equal to the entire frame Θ.

Definition 2.6 **Plausibility Function :** A plausibility function, denoted Pl, is the maximum extent to which one may believe hypothesis A.

That is

$$Pl(A) = 1 - Bel(\bar{A}) = \sum_{B \cap A \neq \phi} m(B), \tag{2.15}$$

where $Bel(\bar{A})$ is the support for negation of A or the doubt on A. Note that $1 - Bel(\bar{A})$ is not the same as $Bel(A)$ since evidence partially in favor of a event A does not necessarily mean that the evidence is partially in favor of its negation. In the Dempster-Shafer model, $Bel(A) \leq 1 - Bel(\bar{A})$. This avoids the Bayesian restriction that requires $P(A) = 1 - P(\bar{A})$.

Definition 2.7 **Commonality Function :** A commonality function, denoted $Q(A)$, is defined as

$$Q(A) = \sum_{B \subset \theta, A \subset B} m(B). \tag{2.16}$$

The relationship between the commonality function and plausibility function is as follows:

$$Q(A) = \sum \{(-1)^{|B|+1} Pl(B) | \phi \neq B \subseteq A\} \tag{2.17a}$$

$$Pl(A) = \sum \{(-1)^{|B|+1} Q(B) | \phi \neq B \subseteq A\} \tag{2.17b}$$

where $|B|$ denotes the number of elements in the set B. Note that by definition, $Q(\phi) = 1$ and $Pl(\phi) = 0$ for any belief function.

One of the advantages of Dempster-Shafer model is that it allows the representation of uncertainty as well as the representation of ignorance. The amount of uncertainty with respect to a hypothesis A given the evidence can be represented as the interval of the $Bel(A)$ and $Pl(A)$, ie, $[Bel(A), 1 - Bel(\bar{A})]$. The degree of ignorance about a hypothesis A is the difference $Pl(A) - Bel(A)$.

Dempster's rule provides a way to combine the support for a hypothesis A based on multiple bodies of evidence. Let Bel_1, Bel_2, m_1, and m_2 be two belief functions and their corresponding *bpas*, respectively. Then the *bpa* m of the new belief function Bel can be calculated as the orthogonal sum of m_1 and m_2.

Definition 2.8 **Orthogonal Sum :** The orthogonal sum of m_1 and m_2 is defined as

$$m(A) = \begin{cases} 0 & \text{if } A = \phi \\ K \sum_{X \cap Y = A} m_1(X) m_2(Y) & \text{if } A \neq \phi \end{cases} \tag{2.18a}$$

where K^{-1} is the normalization factor

$$K^{-1} = 1 - \sum_{X \cap Y = \phi} m_1(X) m_2(Y) \tag{2.18b}$$

and X and Y are individually varied over all subsets of Θ.

The orthogonal sum and normalization factor can be also obtained from the commonality functions

$$K^{-1} = \sum \{(-1)^{|B|+1} Q_1(B) Q_2(B) | \phi \neq B \subseteq \theta\} \tag{2.18c}$$

and

$$Pl(A) = K \sum \{(-1)^{|B|+1} Q_1(B) Q_2(B) | \phi \neq B \subseteq A\} \tag{2.18d}$$

where Q_1 and Q_2 are the commonality functions for Bel_1 and Bel_2, respectively.

Among others, the computational complexity is a main problem in applying the D-S model. It is necessary to find a method that permits the application of the D-S model in a hierarchical hypothesis space while avoiding exponential time requirements. We will discuss this issue in Chapter III.

The advantages of the D-S model over previous approaches include the following:

1. It provides a coherent approach to the management of uncertainty among hierarchically related hypotheses.

2. It does not require the use of an *a priori* probability distribution, which is usually extremely difficult to determine and is forced to artificial precision by the mathematical requirements of traditional probabilistic approaches.

3. It is good for combining evidence from multiple sources. The combination order does not effect the result.

However, there are some problems in the D-S model that are yet to be solved. The main problems are:

1. There exist different interpretations of the D-S theory which can lead to different results. The organization of the hierarchical evidence space also affects the reasoning results.

2. In the case of a long chain of inferences which are required by handling uncertainty in a hierarchically organized network, the computational complexity is considerably increased due to iteratively using the Dempster's combination rule. It is necessary to find a method which allows the possibility of applying the D-S model in a hierarchical hypothesis space while avoiding the exponential time requirements.

2.2.4 Zadeh's Fuzzy Set Theory

Begun in 1920's, some multi-valued logic systems including model logic (Carnap, 1960), intuitionistic logic (Heyting, 1971) and temporal logic (Rescher *et al.*, 1969) were developed because of the apparent inability of classical logic to model degrees of truth and uncertainty (Rescher four et al., 1969,1971). However, the comprehensive models of a set based upon multi-valued logic were not established until Zadeh's fuzzy set theory (Zadeh, 1965) was proposed in 1965. Zadeh's fuzzy set theory is a generalization of the ordinary mathematical concept of set. It is well suited for representing the imprecision of vague linguistic predicates like "young," "old," "medium," etc., and the imprecise quantification related to linguistic concepts like "more," "some," "few," and "majority" etc.

Let F be a fuzzy subset of a universe of discourse U and X be a variable taking values in U. Then a membership function $\mu_F(X)$ expresses the degree of membership of each element X in the fuzzy subset F and $\mu_F(X) \longrightarrow [0,1]$. The larger the value of $\mu_F(X)$, the higher the possibility that $X \in F$. The definition of the membership function is usually subjective and context sensitive.

Fuzzy reasoning can be implemented under the set operations on fuzzy set of their membership functions (Zadeh, 1975). Among others, three most fundamental operations are

- Set complement: $\mu_{\bar{F}}(X) = 1 - \mu_F(X)$.
- Set intersection: $\mu_{F \cap G}(X) = min(\mu_F(X), \mu_G(X))$.
- Set union: $\mu_{F \cup G}(X) = max(\mu_F(X), \mu_G(X))$.
- Cartesian product of F and A: $\mu_{F \times A}(X,Y) = min(\mu_F(X), \mu_A(Y))$.

where F and G are fuzzy subsets of universe U and $X \in U$; A is a fuzzy subset of universe V, and $Y \in V$. A typical example for logical reasoning using a fuzzy set might be

$$IF \quad A \quad THEN \quad B \quad ELSE \quad C \quad = \quad (A \times B) \cup (\bar{A} \times C)$$

The design of the membership function plays an important role in fuzzy set theory. The membership function, however, is context sensitive, and a generic membership function for all situations is impossible. Therefore, different membership functions will result in different computational complexity and different reasoning performance. Furthermore, fuzzy set theory allows us to formalize the vague concept "tall," but it is not useful for representing the approximate height of a person. Thus the theory seems more appropriate for representing roughly defined concepts rather than uncertainty information about the world.

2.3 Uncertainty Reasoning in Vision Systems

In recent years, the application of artificial intelligence in computer vision forced AI to confront a noisy, uncertain, real world. Uncertainty reasoning has been used as a promising tool for solving these difficulties in computer vision systems. Unfortunately there has been no significant success in using it as a unified method throughout the entire system.

Lowe has tried the Bayesian model in his SCERPO (an acronym for Spatial Correspondence, Evidence Reasoning, and Perceptual Organization) (Lowe, 1985). Lowe realized that the attractive feature of using evidential reasoning for computer vision is that it allows us to combine information of varying reliability from many sources. The evidential reasoning was to be used to reduce the search space and to match the object model. Unfortunately, the evidential reasoning component of SCERPO was less developed than other parts of the system. The system was tested by only a single object under consideration. No details of the reasoning component have been reported.

Wesley (Wesley, 1986) found that there were various features of the DS theory that were not readily available in pure Bayesian-based approaches. He discussed some work on integrating the DS theory into a knowledge-based high-level computer vision system through a set of outdoor image interpretations. The results suggested that the performance of a knowledge-based system was improved by using DS theory. Difficulties were raised from the theory for generating mass functions and from the independence requirement of Dempster's rule. The DS theory that was used in the reasoning system involved single level reasoning.

Li and Uhr use the evidential reasoning mechanism based on the D-S model to reason about components in an outdoor scene (Li and Uhr, 1987). After the initial labeling for each extracted region, contextual geometrical relations were used as evidence to improve the labeling results. It was reported that building windows were recognized based on belief functions in the D-S model. Apparently the reasoning processes in this system are only for singleton hypotheses.

Lehrer and Reynolds proposed a method for initial hypothesis formation in image understanding (Lehrer, Reynolds, and Griffith, 1987). An alternative computational approach to the D-S model is used for the evidence combination. The hypotheses formed by this method are intended to provide an initial focus of attention which is used in an image understanding system to interpret outdoor scenes. This method, as the authors claim, is capable of automatically generating a knowledge base for the formation of initial object hypotheses using statistical information provided from a set of training objects. This is a successful application of uncertainty reasoning in a particular part of a vision system. However, they do not show how to use uncertainty reasoning throughout the entire recognition process.

An evidence-based 3-D vision system for range images was proposed by Hoffman and Jain (Hoffman and Jain, 1987). Different surface patches are extracted by range image segmentation. As evidence features, these patches are associated with the object models in the database. A measure of similarity between the observed features and a set of salient features for a given object in the database is used to interpret the unknowns in the image. Instead of the conventional uncertainty methods we reviewed above, this system uses an evidence matrix and a similarity function to combine evidence and to make decisions. The evidence similarity measure has a value between -1 and 1. A value of this measure close to 1 indicates a good match between the observed evidence features and the positive evidence features corresponding to a specific object in the system knowledge base, whereas a value of -1 indicates the opposite. Object recognition is based upon detecting a large value of the similarity measure. This idea behind the method is somewhat similar to that of ACRONYM (Brooks, 1981).

A description of work-in-progress on PSEIKI (a Production System Environment for Integrating Knowledge with Images) was presented by Andress and Kak (1987). The proposed system uses domain-independent information and performs inexact reasoning based upon the Dempster-Shafer formalism in a hierarchical space. There is no detailed description of how the inexact reasoning is applied or how the D-S model is used. The basic idea from the description is parallel to our work on the DNESYS system (Qian, Ehrich, and Campbell, 1990). A paper recently published by Andress and Kak (Andress and Kak, 1988) discusses in detail the evidence accumulation mechanisms and the control structure in PSEIKI. Although only a set of simple-looking images, such as a sidewalk scene with a few edges, were tested, their research work represents a new effort in the right direction.

From the above review, we see that although applications of uncertainty reasoning in expert systems have become popular, the uses of uncertainty reasoning in computer vision systems are in their infancy. Uncertainty reasoning in the systems described above is used only as an aid at a particular processing level. It is not used as a consistent method throughout all processing levels in the visual evidence hierarchy. Fortunately, more and more people tend to agree that uncertainty reasoning is a necessary component of a computer vision system. It is likely that some vision systems in which uncertainty reasoning plays an important role will appear in the next few years.

Chapter III
THE ADOPTING THE D-S MODEL FOR COMPUTER VISION SYSTEMS

The adoption of the uncertainty reasoning theory of the D-S model as the paradigm of our vision system model gives rise to numerous problems. Among them, two are crucial. One is how to represent uncertain information based on hierarchical visual evidence. The other is how to reason about visual events in a hierarchical visual evidence space while keeping the computation complexity under control. We will discuss these two problems theoretically in this chapter and will give our detailed solutions in Chapter V. In addition, there is a problem with Dempster's rule when some information is conflicting. This is discussed and a new algorithm to solve this problem is proposed in this chapter.

3.1 Uncertain Information Representations in Computer Vision Systems

3.1.1 Uncertainty in Visual Knowledge

The reasoning process in the proposed computer vision system requires the description of various levels of visual information and the use of knowledge to transform visual information from one level to another. The performance of a computer vision system depends upon the efficiency of use of visual knowledge. There is no efficient way to use real world knowledge without proper representations. Therefore, knowledge representation in a computer vision system is the same key issue as in a purely symbolic AI system.

According to Binford (1982), there are three levels of knowledge in computer vision: physical knowledge, perceptual knowledge, and semantic knowledge. What he did not explain is that associated with each level of knowledge there are different uncertainty sources.

1. Physical knowledge. This is the knowledge about the physical world where images are taken. Since images of concern to us are taken in the 3-D physical world, the physical laws dominating the process can be used as part of the knowledge source. Uncertainty in this type of knowledge source is caused by uneven or insufficient illumination, occlusion, distortion or dissimilar views, unknown camera parameters, and noise, etc.

2. Perceptual knowledge. This is the knowledge that is used to group primitive pictorial entities into more global ones. Properties of an image such as proximity, similarity, continuity, smoothness, symmetry, and so on, belong to this level of knowledge. Uncertainty in this type of knowledge source is caused by imperfect low-level processing operators such as over- or under segmentation, imprecise grouping rules, or inadequate training sets.

3. Semantic knowledge. This knowledge concerns the properties of and relations among real-world objects. It is especially important for recognizing membership in abstract object classes. Uncertainty in this type of knowledge is caused by inflexible relational models, imprecise interpretation rules, and mismatching because of missing data or information gaps produced by the raw data or by lower-level processing. Recognition at high levels is performed in terms of generic objects, while measurement (observation) is in terms of a specific object instance. As a result, there is a large information gap between abstract

concepts and an observed image. Therefore, we have to fill in the gap through computation and reasoning so that a correspondence is established between concepts and image data.

Human visual knowledge accumulation is usually from specific to general or from concrete to abstract. There is no doubt that the abstraction process is a bottom-up process. We can not see an "abstract dog" in the real world. After seeing lots of individual dogs such as big dog, small dog, white dog, and yellow dog, then we may build up an "abstract dog" concept with the common features of different individual dogs in mind. Afterward, we have the second step. When we are told that "X is a dog", and then a semantic link from the "abstract dog" concept is made to X. Consequently, we know that X is an animal with all properties or features of the "dog" concept. Also, perception is a process that involves the use of reasoning to reduce uncertainty about visual events. However, when knowledge is transferred inappropriately upward, the distorted knowledge will cause misinterpretion of visual events. This causes further uncertainty.

Therefore, ambiguity or uncertainty is an important aspect of visual knowledge. Then the problem is how to express uncertainty in computer vision. Rather than "all-or-none", there are always other terms such as "sometimes", "somehow", "maybe", 'probably', etc. for describing uncertain knowledge in vision. For example, "sometimes" is used to describe objects whose appearances change as a function of time: most trees appear green in the summer but dark gray in the winter. The recognition of a house requires the knowledge about abstract "house". There are many type of "house" with different appearances. The use of belief (uncertainty) values for expressing this kind of diversity is necessary for the visual knowledge representation. Belief value propagation, which will be described later, can be used for several purposes in computer vision:

(1) It provides a way to express the uncertainty facts, concepts, and ambiguity knowledge quantitatively.

(2) It may guide the reasoning or searching process.

(3) Instead of giving a list of possible instances, it can be used to make an optimal choice between competing instances.

(4) It may handle exceptions when visual events include some unexpected features.

There are many ways to represent visual knowledge. Among them, the *semantic network* (Quillian, 1968) and *frames (or schemata)* (Minsky, 1975) are the most popular methods.

A semantic net represents information as a set of nodes linked by labeled arcs that represent relationships among the nodes. In computer vision, most relational models are represented by using semantic networks. However, relations (or links) between nodes in a semantic network are essentially *YES-NO* or *IS-A* relations. An *IS-A* relation is a common representation method for organizing the domain knowledge. However, when a concept can be translated in different ways, a problem arises with using a simple "IS-A" relation. An example of this is determining the front vertex in a NECKER CUBE VIEW. A semantic network does not give us a criterion to make an optimal choice between the competing instances. This is because the semantic network has not any attached belief value for its links. Furthermore, a semantic network is unable to handle exceptions without any belief value propagation. This is especially crucial in computer vision because objects do not look the same due to occlusion, dissimilar views, and distortion. For example, a semantic network can be used to describe a table with a smooth face and four legs. But if only two legs can be seen in a picture, it is difficult to drive a match with that semantic network.

On the other hand, a frame is a data structure for hierarchical structured information in which the top levels represent things that are always true and the lower levels have slots to be filled by specific instances or attributes of events. In computer vision, frames can be used to handle visual information hierarchically and can hold attributes needed to generate hypothesis about visual events. It also may be used to describe one visual event from different viewpoints. Without belief value propagation, however, the same problem may occur as in the semantic network, and uncertainty reasoning can not be carried out.

To represent visual knowledge with uncertainty information, a belief value propagation is used by attaching belief values to frames which is embedded in a semantic network (Figure 2). First, a visual event hierarchy is constructed in our reasoning system. More general and more abstract events are put into the higher level. The most specific entities are treated as individual nodes which at the bottom. All the hierarchical nodes are connected by relational links such as *IS-A* and *PART-OF* and by geometric links such as *NEXT-TO, ABOVE, RIGHT-OF, SURROUNDED-BY* etc. which describe the geometric relations among visual events. All of these functional links are associated with belief values. These belief values represent uncertainty such as "sometimes", "somehow", "maybe", "probably", etc. In the reasoning and matching process, the belief value attached to each functional link provides good guidance. It makes the system choose the most confident alternative from the set of uncertain choices. An example of this might be selecting among multiple viewpoints for a 3-D scene viewed in a simple 2-D images.

In the above representation, each node in the semantic network corresponds to a visual event which is defined and described by a frame. The links in the semantic network represent interrelationships between these frames. A belief value is attached to each link, which represents the uncertainty in their interrelationship. Each frame contains the following information:

Visual Event Identification

Detailed Description About the Event

Prior Belief Value

Updated Belief Value

Property Measurements

Belief Mapping Function

Links to Higher Level Nodes

Subsemantic Network ID in the Next Lower Level.

Some information in this frame is fixed while other information must be filled in by the reasoning process. Note that the belief values attached to each frame are used to characterize the uncertainty about that visual event itself. The belief values for describing the relationships between this event and other events are attached to the functional links of the semantic network in which the frame embedded. The

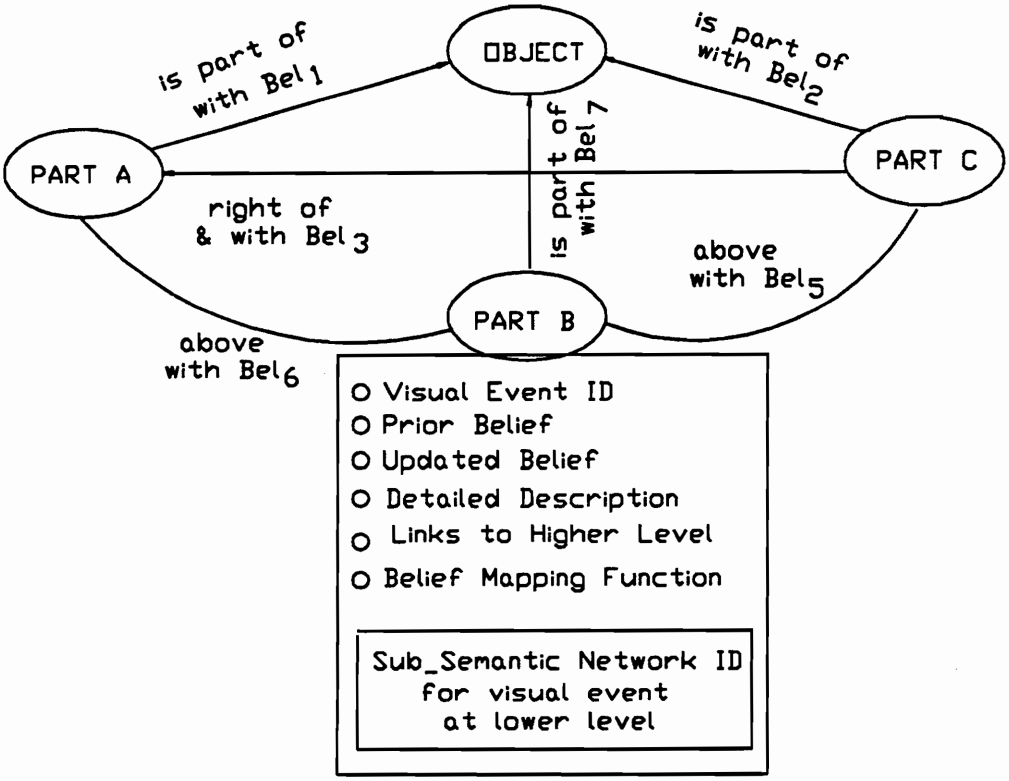


Figure 2. Frames embedded in semantic networks.

subsemantic network ID in Figure 2 gives links for accessing the related visual events in the next lower level which have the same representation forms as in the current level. We will give further discussion on this issue in the following sections and will present a solution in later chapters.

3.1.2 On Belief Functions

Generally speaking, the uncertainty in a piece of information can be represented by means of belief functions. A belief function, denoted Bel, conveys the degree of uncertainty about a logical proposition. Propositions can be represented as a subsets of a given set. Then, the logical notations of conjunction, disjunction, implication, and negation can be translated into set-theoretic notions of intersection, union, inclusion, and complements:

If

$$A \subseteq \Theta \Leftrightarrow H_1(proposition)$$

$$B \subseteq \Theta \Leftrightarrow H_2(proposition)$$

Then

$A \cap B \Leftrightarrow$ conjunction of H_1 and H_2.

$A \cup B \Leftrightarrow$ disjunction of H_1 and H_2.

$A \subset B \Leftrightarrow H_1$ implies H_2.

$A = \bar{B} \Leftrightarrow H_1$ is the negation of H_2.

$\{\phi\} \Leftrightarrow$ a proposition that is known to be false

$\{\Theta\} \Leftrightarrow$ a proposition that is known to be true

A state of belief (or mass distribution) m is defined as

$$m : P(\Theta) \longrightarrow [0,1]$$

$$\sum_{A \subseteq \Theta} m(A) = 1$$

$$m(\phi) = 0$$

and a belief function Bel is defined as

$$Bel(X) = \begin{cases} m(X), & \text{if } X = singleton; \\ 1 & \text{if } X = \Theta; \\ \sum_{Y \subseteq X} m(Y) & \text{otherwise,} \end{cases} \tag{3.1}$$

where Θ is the set of all plausible subsets describing a valid configuration, and $P(\Theta)$ is a probability distribution over the power set of Θ. The term $m(\phi) = 0$ expresses the fact that no confidence is committed to ϕ.

Definition 3.1 **Committed :** A belief value is said to be committed to a subset if that value gives the confidence that some hypotheses in the subset is true.

Definition 3.2 $Bel^{\downarrow}_{\Theta}(A)$ **:** The belief value for A under the consideration of all evidence for A in the subset of A and A itself.

Definition 3.3 $Bel^{\downarrow}_{A}$ **:** The belief value for A under the consideration of all evidence for A in the subset of A except A itself.

Definition 3.4 $Bel_A(A)$: The belief value that is exactly committed to A not to the subset of A.

A portion of belief committed to one subset is also committed to any subset containing it. This corresponds to the case in which a portion of the belief committed to a proposition A, denoted $Bel_{\Theta}^{\downarrow}(A)$, is also committed to any other proposition it implies, denoted $Bel_{A}^{\downarrow}(A)$. A portion of belief that is committed exactly to a given subset A (not to the subsets of A), denoted $Bel_A(A)$, is the quantity $m(A)$ which is called a basic probability assignment (bpa). Therefore, to obtain the measure of total belief committed to A, the $m(B)$ for all proper subsets B of A should be added to $m(A)$. That is

$$Bel(A) = Bel_{\Theta}^{\downarrow}(A) = Bel_A(A) \oplus Bel_{A}^{\downarrow}(A) = m(A) + \sum_{\substack{B \subset A \\ B \neq A}} m(B) = \sum_{B \subseteq A} m(B). \qquad (3.2)$$

In the case of the presence of imprecise evidence, the probability of a proposition A is only imprecisely known. In this situation, the amount of uncertainty with respect to A and the degree of ignorance about A can be modeled by the interval $[Bel(A),\ Pl(A)]$. The lower bound $Bel(A)$ represents the degree of certainty of A, the upper bound $Pl(A)$ represents the extent to which A cannot be denied, and $Pl(A) - Bel(A)$ represents the ignorance. From the Equation 2.15, we have the following observations:

(1) $Bel(A) = 1 \rightarrow Bel(\bar{A}) = 0$ and $Pl(A) = 0$. This represents the proposition that A is certainly true.

(2) $Bel(\bar{A}) = 1 \rightarrow Bel(A) = 0$ and $Pl(A) = 0$. This represents the proposition that A is certainly false.

(3) $Pl(A) = 1 \rightarrow Bel(A) = Bel(\bar{A}) = 0$. This represents total ignorance about A.

Example 3.1

Let $\Theta = \{a, b, c\}$. Then $2^\Theta = \{\phi, \Theta, \{a, b\}, \{a, c\}, \{b, c\}, \{a\}, \{b\}, \{c\}\}$ is the power set of Θ. If an event $A = \{a, b\}$, then $\{a, b\}, \{a\}, \{b\}$ are A's subsets. Suppose the initial *bpa* over the Θ are

$$m(\phi) = 0$$

$$m(\Theta) = 0.2$$

$$m(\{a, b\}) = 0.3$$

$$m(\{a, c\}) = 0.05$$

$$m(\{b, c\}) = 0.05$$

$$m(\{a\}) = 0.15$$

$$m(\{b\}) = 0.15$$

$$m(\{c\}) = 0.1$$

where $m(X)$ measures the belief that one commits exactly to X, not the total belief that one commits to X. Thus,

$$\sum_{X \subseteq \Theta} m(X) = 1;$$

$$Bel(A) = \sum_{B \subseteq A} m(B) = m(\{a, b\}) + m(\{a\}) + m(\{b\}) = 0.3 + 0.15 + 0.15 = 0.6.$$

The $Pl(A)$ can be obtained in three different ways.

By Equation 2.15 we have

$$\begin{aligned} Pl(A) &= 1 - Bel(\bar{A}) \\ &= 1 - \sum_{B \subseteq \bar{A}} m(B) \\ &= 1 - m(\{c\}) = 1 - 0.1 = 0.9. \end{aligned}$$

By Equation 2.15 we also have

$$\begin{aligned} Pl(A) = 1 - Bel(\bar{A}) &= \sum_{B \cap A \neq \phi} m(B) \\ &= m(\{a,b,c\}) + m(\{a,b\}) + m(\{a,c\}) + m(\{b,c\}) + m(\{a\}) + m(\{b\}) \\ &= 0.2 + 0.3 + 0.05 + 0.05 + 0.15 + 0.15 = 0.9. \end{aligned}$$

By Equation 2.17b

$$\begin{aligned} Pl(A) &= \sum \{(-1)^{|B|+1} Q(B) \mid \phi \neq B \subseteq A\} \\ &= (-1)^2 Q(\{a\}) + (-1)^2 Q(\{b\}) + (1-)^3 Q(\{a,b\}) \\ &= \sum_{B \supseteq \{a\}} m(B) + \sum_{B \supseteq \{b\}} m(B) - \sum_{B \supseteq \{a,b\}} m(B) \\ &= 0.7 + 0.7 - 0.5 = 0.9 \end{aligned}$$

where $Q(X)$ is the commonality function which is defined by Equation 2.16.

This computation leads to the following conclusions:

(1) The degree of one's belief in the proposition A is 0.6.

(2) The degree to which one cannot believe the proposition that A is false is 0.9.

(3) The degree of ignorance for A is $0.9 - 0.6 = 0.3$.

(4) The degree to which one believes that A is false is 0.1.

In computer vision, we often need to make decision about a certain visual event at different processing levels based upon incomplete information or imprecise data. For each hypothesis corresponding to a visual event, the belief function and the interval $[Bel, Pl]$ can be used to represent the uncertainty in the reasoning process. $[1, 1]$ is used to completely confirm the hypothesis; $[0, 0]$ is used to completely deny the hypothesis; $[0, 1]$ is used to totally ignore the hypothesis. If $0 < Bel < 1$, then $[Bel, 1]$ is used to represent the degree of support for the hypothesis. If $0 < Pl < 1$, then $[0, Pl]$ is used to represent the degree of rejection of the hypothesis. If $0 < Bel$ and $Pl < 1$, then $[Bel, Pl]$ is used to represent the degree of belief in the hypothesis with the amount of uncertainty $Pl - Bel$.

3.1.3 Uncertainty Transfer and Evidence Discount

In computer vision, one confirmed hypothesis may be treated as a piece of evidence for other hypotheses at a higher level. In other words, the asserted proposition is available for making further inferences at the next processing level. However, if the confirmed hypothesis has a considerable amount of uncertainty (i.e. the interval $[Bel, Pl]$ is quite large), then the evidence itself is unreliable.

Definition 3.5 **Information Gap :** The information gap is the difference between the completely true evidence and the unreliable evidence.

There are two possible ways to treat unreliable evidence when the evidence is mapped to the evidence strength for supporting hypotheses at the higher level. One is that as soon as a hypothesis is confirmed, it is treated as completely true evidence. The information gaps are filled immediately by prior knowledge associated with that hypothesis. The uncertainty is cut at the current reasoning level. The other is that the degree of uncertainty is transferred to the next higher inference level by means of evidence discounting. The information gaps are not filled until the final decision making level. The first method is simple and fast, but when the number of pieces of unreliable evidence is large and the amount of uncertainty in each piece of unreliable evidence at the current reasoning level is significant, this method may mislead subsequent interpretation at the higher level. The second method allows the higher level reasoning to take the influence of the lower level uncertainty into account. We will discuss the second method in more detail.

The original idea of "discounting beliefs" was proposed by Shafer (Shafer, 1976) to deal with unreliable sources of information. It was not intended to solve the uncertainty transfer problem in an evidence hierarchy like in computer vision. Let $Bel(A)$ be a belief function and α be the probability that the source is unreliable. To express doubts on the unreliable source, $m(A)$ is reduced and $m(\Theta)$ is increased by the discount rate α. That is

$$m'(A) = (1-\alpha)m(A) \tag{3.3.a}$$

$$m'(\Theta) = \alpha + (1-\alpha)m(\Theta). \tag{3.3.b}$$

Note that the quantity $m(\Theta)$ corresponds to the amount of ignorance. Discounting beliefs in one piece of evidence corresponds to increasing the amount of ignorance.

The problem now is how to determine the discount rate α for our task. Suppose for a confirmed hypothesis B that $Bel(B)$ is 0.5, $Pl(B)$ is 0.8, and $Bel(\bar{B})$ is 0.2. The doubt on B is expressed in $Bel(\bar{B})$ and $Pl(B) - Bel(B)$. $Bel(\bar{B})$ expresses the disbelief in B and $PL(B) - Bel(B)$ expresses the amount of ignorance for B. To transfer uncertainty between levels of hierarchy, these two factors have to be considered together. The discount rate should be a function of both. That is

$$\alpha = f(Bel(\bar{B}), PL(B) - Bel(B)) = W_1 Bel(\bar{B}) + W_2(Bl(B) - Bel(B)) \tag{3.4}$$

where W_1 and W_2 are the weights, $W_1 > W_2$, and $W_1 + W_2$=1. The reason for setting $W_1 > W_2$ is that the disbelief weighs more than the ignorance does for the unreliability of evidence. Since the mapping function, which maps the body of evidence to evidence strength $m(X)$ for hypotheses at the higher level, treats the B as a piece of completely true evidence, the discounting rate α makes the uncertainty transfer possible. For above example, if $W_1 = 0.66$ and $W_2 = 0.34$, then the discount rate is

$$\alpha = 0.66 \times 0.2 + 0.34 \times 0.3 = 0.234.$$

3.1.4 Dependent Measurements

To adopt the D-S model for our computer vision system, the evidence from different sources at each level of hierarchy is supposed to be independent (Dempster, 1967). This assumption is not always valid in real applications.

The definition of statistical independence can be described in the following way:

Let $A_1, A_2, \ldots, A_n$ be n events, $1 < k \leq n$, and $1 \leq i_1 < i_2 < \ldots < i_k \leq n$. The events $A_1, A_2, \ldots, A_n$ are called mutually independent events if and only if

$$P(A_{i_1}, A_{i_2}, \ldots A_{i_k}) = P(A_{i_1})P(A_{i_2}) \ldots P(A_{i_k}).$$

where $P(X)$ is the probability of event X.

The concept of independence in D-S model is slightly different from that in probabilistic model since these models treat chance in different ways (Wesley, 1986). Dempster (1967) pointed out that different measurements by different observers on different equipment would often be regarded as independent. The concept of independence of evidence sources used in our system is in the sense that evidence sources are not influenced by each other. To recognize an object, for example, the visual measurements obtained from three different viewpoints may regarded as three independent sources.

The visual evidence in a computer vision system is basically derived from different measurements which are outputs of different processing modules. Some of these measurements are clearly independent, and some may be not. At the region level, for example, some texture measurements are independent of the mean greytone measurement at the region level. The length of a region boundary is independent of the mean greytone measurement of that region. The geometric measurements of a region, such as circularity, area, mean radius, and orientation of the region, are independent of the greytone level measurements within that region. Also measurements of spatial relations between regions are independent of all measurements of each region itself. The pieces of evidence used at a particular level may be made independent by carefully choosing the types of measurements made at that level.

Dependent measurements in computer vision often take forms in which one measurement can be expressed as a function of other measurements. As pieces of evidence, if the both measurements enhance the belief for a hypothesis in the same direction, then they can be used by the D-S model with certain inaccuracy. Because inaccuracy of measurements are allowed in uncertainty reasoning and the measurements chosen in computer vision usually support the belief in the same direction, this kind of measurement dependency should not be forbidden when applying the combination rule.

In this sense, different measurements by different selected processing modules at each level of hierarchy may be considered as relatively independent evidence sources in a computer vision system.

3.2 The Reduction of Computational Complexity

Computational complexity is a main problem in adapting the D-S model to computer vision systems. Reasoning about the visual events in a hierarchical visual evidence space usually requires a long chain of inferences in a hierarchically organized network. In general, the amount of computational complexity for exact implementation of Dempster's combination rule increases exponentially with the number of subsets in a frame of discernment. It is necessary to find a method that permits the application of the D-S model in a hierarchical hypothesis space while avoiding exponential time requirements. Shafer and Logan (Shafer and Logan, 1987) proposed an algorithm which applied Barnett's technique (Barnett, 1981) to a partitioned frame of discernment Θ for the above problem. We will examine this algorithm and adapt it for computer vision systems.

3.2.1 Partitions of a Frame of Discernment

Decomposing a complex problem into several simpler parts is a standard scientific problem solving methodology. Partitioning a large hypothesis space for applying the D-S model is such a strategy.

Definition 3.6 **Partition :** A partition of a frame of discernment Θ is a set of disjoint non-empty subsets of Θ whose union equals Θ.

Such a partition $\mathcal{P}$ can itself be regarded as a frame. Then $\mathcal{P}$ has fewer elements than Θ. The evidence from each partitioning is combined to constrain the final decision.

Definition 3.7 **Refinement :** A partition $\mathcal{P}_1$ of Θ is called a refinement of another partition $\mathcal{P}_2$ of Θ if for every element P_1 in $\mathcal{P}_1$ there is an element P_2 in $\mathcal{P}_2$ such that $P_1 \subseteq P_2$.

Suppose there is a frame of discernment Θ which contains a set of exhaustive and mutually exclusive visual events $\{a, b, c, d, e\}$. Let a partition of Θ

$$\mathcal{P} = \{P_1, P_2, P_3\}$$

and another partition of Θ

$$\mathcal{P}_2 = \{P_{21}, P_{22}\}$$

be such that $P_1 = \{a, b\}$, $P_2 = \{c, d\}$, $P_3 = \{e\}$, $P_1 \subseteq P_{21}$, $P_2 \subseteq P_{22}$, and $P_3 \subseteq p_{22}$ (See Figure 3). The partition $\mathcal{P}$ is called a refinement of $\mathcal{P}_2$.

Let $\mathcal{P}^*$ denote the set consisting of all unions of elements of $\mathcal{P}$. That is

$$\begin{aligned} \mathcal{P}^* &= \Big\{\{P_1 \cup P_2 \cup P_3\}, \{P_1 \cup P_2\}, \{P_1 \cup P_3\}, \{P_2 \cup P_3\}, \{P_1\}, \{P_2\}, \{P_3\}, \phi\Big\} \\ &= \Big\{\{a, b, c, d, e\}, \{a, b, c, d\}, \{a, b, e\}, \{c, d, e\}, \{a, b\}, \{c, d\}, \{e\}, \phi\Big\} \end{aligned} \tag{3.5}$$

which is a field of subsets of Θ.

Definition 3.8 **Carried :** A belief function Bel over Θ is carried by $\mathcal{P}$ if the random subset S corresponding to Bel satisfies

$$Pr[S \in \mathcal{P}^*] = 1.$$

Given a subset $A = \{a, b, c, d\}$ of Θ and given a partition $\mathcal{P}$ of Θ, there are several useful properties of the partition.

(1.) If Bel is carried by $\mathcal{P}$, then

$$Bel(A) = Bel(A_{\mathcal{P}}) = max\{Bel(B) \mid B \subseteq A, B \in \mathcal{P}^*\} \tag{3.6.a}$$

where $A_{\mathcal{P}}$ is the largest element of $\mathcal{P}^*$ contained in A:

$$A_{\mathcal{P}} = \cup\{P \mid P \in \mathcal{P}, P \subseteq A\}. \tag{3.6.b}$$

(2.) If Bel is carried by $\mathcal{P}$, then

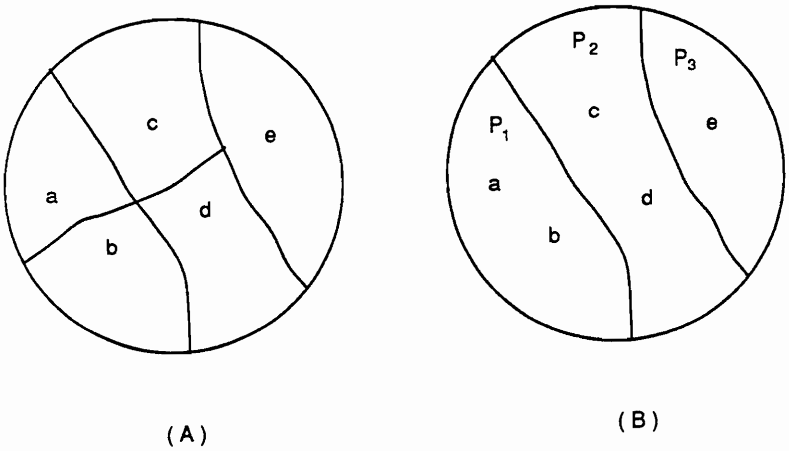


Figure 3. The partition 1 of Θ.

$$Pl(A) = Pl(A^{\mathcal{P}}) = min\{Bel(B) \mid B \supseteq A, B \in \mathcal{P}^*\} \tag{3.7.a}$$

where $A^{\mathcal{P}}$ is the smallest element of $\mathcal{P}^*$ containing A:

$$A^{\mathcal{P}} = \cup\{P \mid P \in \mathcal{P}, P \cap A \neq \phi\}. \tag{3.7.b}$$

Here the example in Figure 3 is used to illustrate the above properties. Since $A = \{a, b, c, d\}$, $P_1 = \{a, b\}$, and $P_2 = \{c, d\}$, it is evident that $P_1 \in \mathcal{P}$, $P_1 \subseteq A$, $P_2 \in \mathcal{P}$, and $P_2 \subseteq A$. Then,

$$A_{\mathcal{P}} = \cup\{P \mid P \in \mathcal{P}, P \subseteq A\} = \{P_1 \cup P_2\}.$$

Similarly,

$$A_{\mathcal{P}} = \cup\{P \mid P \in \mathcal{P}, P \cap A \neq \phi\} = \{P_1 \cup P_2\}.$$

Thus,

$$\begin{aligned}
Bel(A) = Bel(A_{\mathcal{P}}) &= max\{Bel(B) \mid B \subset A, B \in \mathcal{P}^*\} \\
&= max\Big\{\{Bel(P_1) \mid P_1 \subseteq A, P_1 \in \mathcal{P}^*\}, \\
&\quad \{Bel(P_2) \mid P_2 \subseteq A, P_2 \in \mathcal{P}^*\}, \{Bel(P_1 \cup P_2) \mid \{P_1 \cup P_2\} \subseteq A, \{P_1 \cup P_2\} \in \mathcal{P}^*\Big\} \\
&= Bel(P_1 \cup P_2) = Bel(\{a, b, c, d\})
\end{aligned}$$

$$
\begin{aligned}
Pl(A) = Pl(A^{\mathcal{P}}) &= min\{Pl(B) \mid B \supseteq A, B \in \mathcal{P}^*\} \\
&= min\Big\{\{Pl(P_1 \cup P_2 \cup P_3) \mid \{P_1 \cup P_2 \cup P_3\} \supseteq A, \{P_1 \cup P_2 \cup P_3\} \in \mathcal{P}^*\}, \\
&\quad \{Pl(P_1 \cup P_2) \mid \{P_1 \cup P_2\} \supseteq A, \{P_1 \cup P_2\} \in \mathcal{P}^*\}\Big\} \\
&= Pl(P_1 \cup P_2) = Pl(\{a,b,c,d\})
\end{aligned}
$$

From the above properties, if $Bel_1, Bel_2, \ldots, Bel_N$ are carried by a partition $\mathcal{P}$, then the Equation 2.18d now can be replaced by the following formulas that involve only elements of $\mathcal{P}^*$ in the partitioned space $\mathcal{P}$.

$$
Pl(A) = K \sum\Big\{(-1)^{|B|^{\mathcal{P}}+1} Q_1(B) Q_2(B) \ldots Q_N \mid B \in \mathcal{P}^*, B \neq \phi, B \subseteq A\Big\} \tag{3.8.a}
$$

where $|B|^{\mathcal{P}}$ denotes the number of elements of $\mathcal{P}$ contained in B, and

$$
K^{-1} = \sum\Big\{(-1)^{|B|^{\mathcal{P}}+1} Q_1(B) Q_2(B) \ldots Q_N(B) \mid B \in \mathcal{P}^*, B \neq \phi, B \subseteq \Theta\Big\} \tag{3.8.b}
$$

$$
Q_i(A) = \sum\Big\{(-1)^{|B|^{\mathcal{P}}+1} (1 - Bel_i(\bar{B})) \mid B \in \mathcal{P}^*, B \neq \phi, B \subseteq A\Big\}. \tag{3.8.c}
$$

Again, we use the same example to illustrate this formula. Here $A = \{a,b,c,d\}$, the elements in $\mathcal{P}^*$ are $\{P_1\}, \{P_2\}$, and $\{P_1 \cup P_2\}$. Suppose that Bel_1 and Bel_2 are both carried by $\mathcal{P}$, then

$$
\begin{aligned}
Pl(A) = &K\Big\{(-1)^2 Q1(P_1) Q_2(P_1) \mid P_1 \in \mathcal{P}^*, P_1 \neq \phi, P_1 \subseteq A\}, \\
&\{(-1)^2 Q_1(P_2) Q_2(P_2) \mid P_2 \in \mathcal{P}^*, P_2 \neq \phi, P_2 \subseteq A\}, \\
&\{(-1)^3 Q_1(P_1 \cup P_2) \mid P_1 \cup P_2 \in \mathcal{P}^*, P_1 \cup P_2 \neq \phi, P_1 \cup P_2 \subseteq A\}\Big\}.
\end{aligned} \tag{3.9}
$$

Now, the question is whether the partition $\mathcal{P}$ discerns the interaction among these beliefs that is relevant to itself. That is, whether the following equation is valid:

$$(Bel_1^{\mathcal{P}} \oplus Bel_2\mathcal{P} \oplus \ldots \oplus Bel_N^{\mathcal{P}}) = (Bel_1 \oplus Bel_2 \oplus \ldots \oplus Bel_N)^{\mathcal{P}}. \tag{3.10}$$

Shafer (1987) gave the following condition: $\mathcal{P}$ discerns the interaction among $Bel_1, Bel_2, \ldots, Bel_N$ that is relevant to itself iff

$$S_1^{\mathcal{P}} \cap S_2^{\mathcal{P}} \cap \ldots \cap S_n^{\mathcal{P}} = (S_1 \cap S_2 \cap \ldots \cap S_N)^{\mathcal{P}} \tag{3.11.a}$$

which is equivalent to

$$S_i \cap P \neq \phi \Leftrightarrow S_1 \cap S_2 \cap \ldots \cap S_N \cap P \neq \phi \tag{3.11.b}$$

where $P \in \mathcal{P}$ and S_i is a focal element of Bel_i.

Given a partition $\mathcal{P} = \{P_1, P_2, P_3\}$ on Θ (Figure 4), suppose in Θ

Bel_1 focuses on $A = \{a, c\}$

Bel_2 focuses on $B = \{a, b\}$.

But in partition $\mathcal{P}$

$$P_1 = \{a\}$$

$$P_2 = \{b, c, d\}$$

$$P_3 = \{e\}.$$

That means

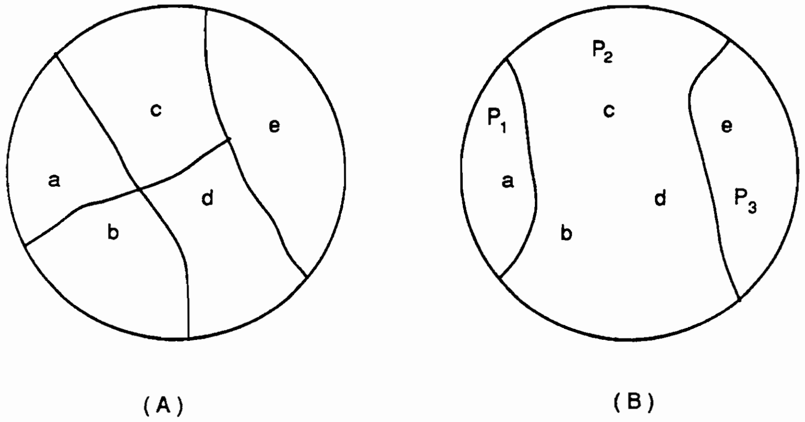


Figure 4. The partition 2 of the Θ.

$$B \not\subseteq P_1 \qquad B \not\subseteq P_2$$

$$A \not\subseteq P_1 \qquad A \not\subseteq P_2$$

$$A \cap B = \{a\} \subseteq P_1,$$

and

$$A^{\mathcal{P}} = \phi \qquad B^{\mathcal{P}} = \phi$$

$$(A \cap B)^{\mathcal{P}} = P_1.$$

Thus,

$$A^{\mathcal{P}} \cap B^{\mathcal{P}} \neq (A \cap B)^{\mathcal{P}}.$$

Therefore, $\mathcal{P} = \{P_1, P_2, P_3\}$ can not discern Bel_1 and Bel_2 to itself. However, if we choose the partition as $\mathcal{P} = \{P_1 \cup P_2, P_3\}$, then this partition can discern Bel_1 and Bel_2:

$$B \subseteq \{P_1 \cup P_2\}$$

$$A \subseteq \{P_1 \cup P_2\}$$

$$A \cap B = \{a\} \subseteq \{P_1 \cup P_2\}$$

$$A^{\mathcal{P}} \cap B^{\mathcal{P}} = (A \cap B)^{\mathcal{P}}.$$

Appropriately partitioning the visual event hypothesis space or a hypothesis space for spatial relationships among visual events is necessary for computer vision since these hypothesis spaces are large and hierarchically structured. The partition, however, should be carefully chosen such that the partition discerns the interaction among belief functions for these hypotheses that is relevant to itself. Then Dempster's combination rule can be used with lower computational complexity.

3.2.2 Applying Barnett's Technique in a Partitioned Space

Barnett (1981) has developed a technique by which the computational complexity for implementing Dempster's combination rule can be linear in the number of elements in Θ if the belief functions being combined are all simple functions focused on singletons or their complements. Shafer's partition ideas enable Barnett's technique extend to the case where belief functions focus on elements of some coarser partition $\mathcal{P}$.

Suppose a frame of discernment

$$\Theta = \{a, b, c\}$$

then

$$\bar{a} = \{b, c\}$$

$$\bar{b} = \{a, c\}$$

$$\bar{c} = \{a, b\}$$

and there are three sets of belief functions

$$\begin{array}{ccc} Bel_a(a) & Bel_a(\bar{a}) & Bel_a(\Theta) \\ Bel_b(b) & Bel_b(\bar{b}) & Bel_b(\Theta) \\ Bel_c(c) & Bel_c(\bar{c}) & Ble_c(\Theta) \end{array}$$

Now, we try to combine these three belief functions by Dempster's combination rule by computing $Bel_a \oplus Bel_b \oplus Bel_c$ and the normalization factor K. First, we compute

$Bel_a \oplus Bel_b$. The orthogonal sum of these two belief functions is shown in Figure 5. There are five different result subsets depending upon the original focal element $S \in \Theta$.

(1.) Exactly one S is a singleton. It is marked by $*$.

(2.) Two or more S are singletons. In this case $\cap S = \phi$. It is marked by $**$

(3.) No S is a singleton. In this case $S \in \{\{\bar{X}\}, \Theta\}$. It is marked by $***$.

(4.) All S are $\{\bar{X}\}$. It is marked by $****$.

(5.) Combinations of case 3 and case 4.

A much more complex situation for the orthogonal sum of these tree belief functions is shown in Figure 6. The same marks as in Figure 5 are used to represent four different cases.

By Equation 2.18b the normalization factor

$$K^{-1} = 1 - \sum_{A \cap B = \phi} m_1(A)m_2(B) = \sum_{A \cap B \neq \phi} m_1(A)m_2(B).$$

By examining Figure 6 we have that the normalization factor for this example is

$m_1 \oplus m_2$ m_1 m_2	$m_1(a)$	$m_1(\bar{a})$	$m_1(\theta)$
$m_2(b)$	$\{\emptyset\}^{**}$	$\{b\}^{*}$	$\{b\}^{*}$
$m_2(\bar{b})$	$\{a\}^{*}$	$\{c\}^{****}_{***}$	$\{\bar{b}\}^{***}$
$m_2(\theta)$	$\{a\}^{*}$	$\{\bar{a}\}^{***}$	$\{\theta\}^{***}$

Figure 5. The orthogonal sum of two functions.

$m_1 \oplus m_2 \oplus m_3$ m_3 $m_1 \oplus m_2$	$\{c\}$	$\{\bar{c}\}$	$\{\theta\}$
$\{a\} \cap \{b\} = \{\Phi\}$	$\{\Phi\}^{**}$	$\{\Phi\}^{**}$	$\{\Phi\}^{**}$
$\{\bar{a}\} \cap \{b\} = \{b\}$	$\{\Phi\}^{**}$	$\{b\}^{*}$	$\{b\}^{*}$
$\{\theta\} \cap \{b\} = \{b\}$	$\{\Phi\}^{**}$	$\{b\}^{*}$	$\{b\}^{*}$
$\{a\} \cap \{\bar{b}\} = \{a\}$	$\{\Phi\}^{**}$	$\{a\}^{*}$	$\{a\}^{*}$
$\{a\} \cap \{\theta\} = \{a\}$	$\{\Phi\}^{**}$	$\{a\}^{*}$	$\{a\}^{*}$
$\{\bar{a}\} \cap \{\theta\} = \{\bar{a}\}$	$\{c\}^{*}$	$\{b\}^{***}$	$\{\bar{a}\}^{***}$
$\{\bar{b}\} \cap \{\theta\} = \{\bar{b}\}$	$\{c\}^{*}$	$\{a\}^{***}$	$\{\bar{b}\}^{***}$
$\{\bar{a}\} \cap \{\bar{b}\} = \{c\}$	$\{c\}^{*}$	$\{\Phi\}^{***}_{****}$	$\{c\}^{***}$
$\{\theta\} \cap \{\theta\} = \{\theta\}$	$\{c\}^{*}$	$\{\bar{c}\}^{***}$	$\{\theta\}^{***}$

Figure 6. The orthogonal sum of three functions.

$$\begin{aligned}
K^{-1} &= \sum_{j=1}^{5} X_j + \sum_{j=1}^{5} Y_j + \sum_{j=1}^{5} Z_j + \sum_{j=1}^{3} T_j + R \\
&= \Big(\sum_{j=1}^{4} X_j + \sum_{j=1}^{4} Y_j + \sum_{j=1}^{2} Z_j\Big) + \Big(X_5 + Y_5 + Z_5 + R + W + \sum_{j=1}^{3} T_i\Big) - W \\
&= \sum_{\substack{exactly \;\; one \\ S_i \;\; is \;\; singleton}} \prod_{S_i \in \Theta} Bel_S(S_i) + \sum_{S_i \in \{\{\bar{S}\},\Theta\}} \prod_{S_i \in \Theta} Bel_S(S_i) - \sum_{\substack{all \;\; S_i = \{\bar{S}\} \\ such \;\; that \; \cap S_i = \phi}} \prod_{S_i \in \Theta} Bel_S(S_i)
\end{aligned} \tag{3.12}$$

where the X_j are the m functions whose focal element $S_i = \{b\}$ with the mark $*$ for $j = 1,2,3,4$ and with the mark $***$ for $j = 5$; Y_j are the m functions whose focal element $S_i = \{a\}$ with the mark $*$ for $j = 1,2,3,4$ and with the mark $***$ for $j = 5$; Z_j are the m functions whose focal element $S_i = \{c\}$ with the mark $*$ for $j = 1,2,3,4$ and with the mark $***$ for $j = 5$; T_1 is the m function whose focal element $S_i = \{\bar{a}\}$; T_2 is the m function whose focal element $S_i = \{\bar{b}\}$; T_3 is the m function whose focal element $S_i = \{\bar{c}\}$; $R = m(\{\theta\})$, and W is one of those m functions whose focal element is ϕ and is counted both in case 3 and in case 4.

This equation can be generalized by

$$\begin{aligned}
K^{-1} &= \sum_{\substack{exactly \;\; one \\ S_i \;\; is \;\; singleton}} \prod_{S_i \in \Theta} Bel_S(S_i) + \sum_{S_i \in \{\{\bar{S}\},\Theta\}} \prod_{S_i \in \Theta} Bel_S(S_i) - \sum_{\substack{all \;\; S_i = \{\bar{S}\} \\ such \;\; that \cap S_i = \phi}} \prod_{S_i \in \Theta} Bel_S(S_i) \\
&= \sum_{D \in \Theta} Bel_D(D) \prod_{S_i \neq D} (Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta)) \\
&+ \prod_{S_i \in \Theta} (Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta)) - \prod_{S_i \in \Theta} Bel_S(\{\bar{S}_i\})
\end{aligned} \tag{3.13}$$

If $Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta) \neq 0$ for all i, then

$$K^{-1} = \prod_{S_i \in \Theta} (Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta))$$
$$\Big(1 + \sum_{D \in \Theta} \frac{Bel_D(D)}{\prod_{S_i = D}(Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta))} - \prod_{S_i \in \Theta} \frac{Bel_S(\{\bar{S}_i\})}{(Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta))}\Big) \cdot \tag{3.14}$$

Note that

$$Bel_D(D) = Bel_S(S_i)$$

$$Bel_S(\{\bar{S}_i\}) + Bel_S(\Theta) = 1 - Bel_S(S_i)$$

$$Bel_S(\{\bar{S}\}) = Bel_S(\bar{S}_i)$$

then K^{-1} can be rewritten as

$$K^{-1} = \prod_{S \in \Theta} (1 - Bel_S(S)) \Big(1 + \sum_{S \in \Theta} \frac{Bel_S(S)}{1 - Bel_S(S)} - \prod_{S \in \Theta} \frac{Bel_S(\bar{S})}{1 - Bel_S(S)}\Big). \tag{3.15.a}$$

By eliminating $\prod_{S \in \Theta}(1 - Bel_S(S))$ which is the common factor when applying it to compute Bel or Pl, we have

$$K^{-1} = \Big(1 + \sum_{S \in \Theta} \frac{Bel_S(S)}{1 - Bel_S(S)} - \prod_{S \in \Theta} \frac{Bel_S(\bar{S})}{1 - Bel_S(S)}\Big). \tag{3.15.b}$$

Thus, we get the same result as that of Barnett (1981) and Shafer four et al.(1987) by a different approach.

From above discussion, it is not difficult to see that a partition can be taken such that the belief functions are dichotomous with dichotomy $\{A, \bar{A}\}$; this means that the belief function has no focal elements other than A, $\bar{A}$, and Θ. The algorithm begins with simple support functions for and against elements of partition $\mathcal{P}$. Let $Pl(A)$ be the plausibility function for the belief function of the orthogonal sum of $\{Bel_P | P \in \mathcal{P}\}$, where Bel_P is dichotomous with dichotomy $\{P, \bar{P}\}$. Let P^+ denote $Bel_P(P)$ and P^- denote $Bel_P(\bar{P})$. Then

$$Pl(A) = K\Big(1 + \sum_{P \subseteq A, P \in \mathcal{P}} \frac{P^+}{1 - P^+} - \prod_{P \subseteq A, P \in \mathcal{P}} \frac{P^-}{1 - P^+}\Big) \tag{3.16.a}$$

where

$$K^{-1} = 1 + \sum_{P \in \mathcal{P}} \frac{P^+}{1 - P^+} - \prod_{P \in \mathcal{P}} \frac{P^-}{1 - P^+} \tag{3.16.b}$$

Thus, a computational complexity which is linear in the number of subsets of a frame can be achieved (Shafer and Logan, 1987).

3.3 Problems with Dempster's Rule Under Conflicting Information

The original version of Dempster's rule is given by Equation 2.18. The idea of using the normalization factor in this formula can be described as follows:

Given a subset A of Θ, the probability mass which commits to A is $m_1(X)m_2(Y)$ such that $X \cap Y = A$. Since there may be more than one terms whose focal element is A, the total probability mass committed to A will be

$$\sum_{X \cap Y = A} m_1(X)m_2(Y).$$

However, it may occur that $m_1(X)m_2(Y) \neq 0$ such that $X \cap Y = \phi$. This contradicts the definition of *bpa* (m function) by which

$$m_1 \oplus m_2(\{\phi\}) = 0$$

$$\sum_{\substack{B \subseteq \Theta \\ B \neq \phi}} m(B) = 1.$$

To be consistent with the definition, the probability mass computed by the orthogonal sum should be normalized. The total probability mass for $X \cap Y \neq \phi$ is

$$1 - \sum_{X \cap Y = \phi} m_1(X)m_2(Y).$$

Thus,

$$m(A) = \frac{\sum_{X \cap Y = A} m_1(X)m_2(Y)}{1 - \sum_{X \cap Y = \phi} m_1(X)m_2(Y)}.$$

However, the normalization factor gives rise to problems with Dempster's rule under extremely conflicting information. In this case, Zadeh (1984) challenged Dempster's rule with the following example.

Let $\Theta = \{a, b, c\}$ and

$$m_1(a) = 0, \qquad m_1(b) = k, \qquad m_1(c) = 1 - k$$

$$m_2(a) = 1 - k, \qquad m_2(b) = k, \qquad m_2(c) = 0$$

where k is a small number. The *bpa* from two different sources are conflict strongly. m_1 suggests that a is absolutely impossible and c is almost completely confirmed, while m_2 suggests that a is almost completely confirmed and c is absolutely impossible. Both m_1 and m_2 suggest that b is unbelievable. The orthogonal sum of m_1 and m_2 given by Dempster's rule is

$$m(a) = 0, \qquad m(b) = 1, \qquad m(c) = 0.$$

This result suggests that b is completely confirmed and either a or b is absolutely impossible, which is certainly not true. This is caused by the normalization factor. Without the normalization, all the values for a, b, or c are very small. To resolve this problem, Yager (1987) slightly modified Dempster's rule by adding the weight of conflict term $\sum_{X \cap Y} m_1(X)m_2(Y)$ to the ignorance $m(\Theta)$. That is,

$$m(A) = \sum_{\substack{X \cap Y = A \\ A \neq \Theta, \phi}} m_1(X)m_2(Y) \qquad (3.17.a)$$

and

$$m(\Theta) = m_1(\Theta)m_2(\Theta) + \sum_{X \cap Y = \phi} m_1(X)m_2(Y) \qquad (3.17.b).$$

The combination result for the above example of this rule shows that a and c are impossible and b is not reliable since $m(b)$ is small and $m(\Theta)$ is almost 1. Note that for more than two pieces of evidence the second step (applying Equation 3.17b) should not be used until all pieces of evidence have been combined by Equation (3.17a).

However, Yager's modified Dempster's rule can only be used under the condition that the conflicting information sources are equally reliable. In above example, if only one of sources is reliable, then Yager's rule will yields the same misleading results.

In our system, partially redundant information from multiple sources is used to reduce the chance that several information sources will report results extremely conflict as in Zadeh's example. The above extreme case may not occur, and usually the number of pieces of evidence is larger to insure that most evidence will enhance each other. Since we are not sure whether all evidence sources are equally reliable or not in the reasoning process, Yager's modified rule can not be used in the whole evidence combination process. Our solution for this problem is to

(1) Apply Dempster's rule to combine one piece of evidence at a time, and start with the piece of evidence which has a small $m(\Theta)$.

(2) Calculate the conflicting term $\sum_{X \cap Y = \phi} m_1(X) m_2(Y)$

(3) If the conflicting term is quite large then push this evidence into a stack.

(4) Repeat Step 1 to Step 3 until all pieces of evidence are processed. Then the first orthogonal sum is obtained.

(5) Pop the pieces of evidence from that stack one by one and combine them by Dempster's rule into the next orthogonal sum according to Steps 1–3.

(6) Continue until there is no piece of evidence in the stacks.

(7) Combine all orthogonal sums obtained so far by repeating Step 1–7.

(8) Finally the number of pieces of evidence is reduced to two. Calculate the conflicting term; if this term is small then Dempster's rule is applied, and otherwise Yager's modified rule is applied.

This algorithm divides the body of evidence into several groups of evidence which that in each group evidence is not extremely conflicting. Then Dempster's rule is applied to each group of evidence. The orthogonal sums obtained from these groups are combined by the same procedure. Recursively applying this procedure, the number of orthogonal sums is finally reduced to two. Since these final orthogonal sums are obtained from several information sources these two can be assumed to be equally reliable. Then Yager's modified rule can be applied correctly. However, if one of the final two terms is only from a single evidence source and it still completely conflicts with the other, then this evidence can be discarded by majority voting. In this algorithm, Step 1–3 insure that all pieces of evidence combined by Dempster's rule are not completely conflicting. The conflicting term evaluation in the Step 2 plays a crucial role in this algorithm. If there are sufficient evidence sources this algorithm is also able to exclude a few unreliable evidence sources.

Chapter IV

HIERARCHICAL VISUAL EVENT SPACE

It has been widely believed that a hierarchical computational framework for reasoning from the raw image to a high-level symbolic description is a proper way to achieve the goal of computer vision. Typical examples in the domain of early vision are the $2\text{-}\frac{1}{2}$ D sketch (Marr, 1976, 1982) and intrinsic images (Barrow and Tenenbaum, 1981). Our view is that an active process of uncertainty reasoning about a hierarchical visual evidence space is a most important aspect of computer vision. Based upon this point of view, a succession of levels of representation for visual information is necessary for a computer vision system. The number of levels in the hierarchy depends upon the computational method. Too many levels of representation would considerably increase the computational complexity while too few would make it difficult to make a reasonable model of a structured world. Usually the initial levels are constrained by the essential primitives that must be computed directly from an image. The intermediate levels are constrained by what is available from preceding levels and what is required by succeeding levels. The top levels are directly constrained by the final processing goal. Regardless of how many levels are chosen, the visual event space should be hierarchically represented from the level of image primitives to a high-level interpretation. Multiple levels of representation and stages of processing are essential. Uncertainty reasoning plays an important role which purposely conducts information exchange between different processing levels and communication between the processing elements within each level. First, two definitions will be helpful for the following discussion.

Definition 4.1 **Visual Event :** A visual event is a physical occurrence in the visual field or, at a higher level, an interpretation of a physical occurrence.

Definition 4.2 **Visual Evidence :** Visual evidence is the quantitative support for a hypothesis that explains the visual event.

A hypothesis that explains a visual event may also be supported or refuted by evidence from other visual events or by other confirmed hypotheses. A confirmed hypothesis at some level of abstraction is a visual event that provides evidence for reasoning about hypotheses at a higher level of abstraction. Any visual event at a particular level of abstraction should provide as complete a description as possible of the visual events at preceding levels. It should provide as strong a support as possible for the next higher level representation.

There are many visual features, such as edges, textures, graytones, colors, etc., that are frequently used in computer vision systems. There are also many algorithms that are used to extract these features. The problem is that it is not easy to judge which features are efficient and which are not. Each feature provides a piece of information, but none is particularly reliable by itself. For example, the intensity of a line or the gradient amplitude of an edge do not necessarily correlate with the perceptual significance of the line or edge. Sometimes low contrast edges may correspond to true boundaries of objects (Figure 7). This causes problems for edge operators which label the edges from a graytone image only on the basis of gradient amplitude thresholding. Furthermore, some perceptually significant lines, such as subjective contours and illusory contours, may not be physically present in the image (Figure 8). Therefore, it is necessary to use multiple features at different representation levels and to integrate multiple visual evidence to deduce the most plausible events which explain the visual image.

Knowing what features are used by the human visual system may be helpful for determining an implementation of a computer vision system. Some researchers (Lowe, 1985; Walters, 1987; Malik, 1987; Nalwa, 1988) argue that line drawings convey much visual information. This is based upon the fact that we can so readily recognize objects from line drawings which lack color, motion, stereoptic depth cues, shading, and natural textures, etc. Therefore, for effective image primitives, we can often use linedrawing-like features, including edges, segments, lines, curves, contours, etc., as basic features and use regional features such as textures, graytone, and colors as auxiliary features.

The visual event hierarchy in our system has the following levels:

1. Significant points
2. Significant lines
3. Significant contours
4. Significant shapes
5. Primary components of objects
6. Objects
7. The scene
8. The world.

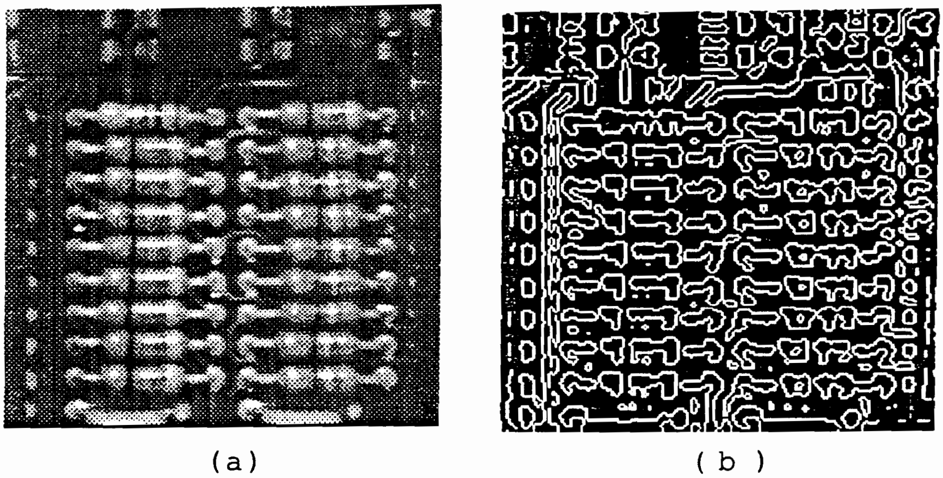


Figure 7. Edges may not correspond to true boundaries of objects. (a) Original image; (b) the edge image.

triangle).

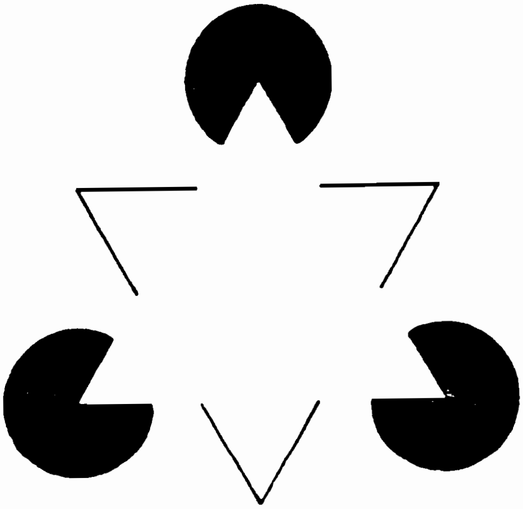


Figure 8. Illusory contours may not be physically present in an image (Kanizsa's triangle).

These visual events range from local to global, and regional events may spatially overlap local features at a lower level of abstraction. For example, levels 4 to 5 may be overlapped by regional properties such as texture, graytone, and colors etc..

In general, suppose there are I levels in the visual event hierarchy. Let $\mathcal{E} = \cup_{j=1}^{I-1}\mathcal{E}_j$ be a hierarchical visual event space. $\mathcal{E}_i = \{\mathcal{E}_{i,1}, \mathcal{E}_{i,2}, ..., \mathcal{E}_{i,N_i}\}$ is a finite set of possible visual events at level i, $i = 1, 2, ..., I-1$. Usually the evidence for events at level i come from confirmed events at level $i-1$ in a bottom-up process, although in principle, the evidence could come from confirmed events at any level, either higher levels or lower levels, including the raw data. The ability to have all levels of visual processing influence the current level visual interpretations would turns out to be very important. Let $\mathcal{G}_i \subseteq \mathcal{E}_i$ be the set of all mutually exclusive elements of $\mathcal{E}_i$. $\mathcal{G}_i = \{\mathcal{G}_{i,1}, \mathcal{G}_{i,2}, ..., \mathcal{G}_{i,G_i}\}$ where $G_i \leq N_i$. Let $E = \cup_{j=1}^{I-1}E_j$ be a hierarchical visual evidence space. E_i is a subset of E, and $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,M_i}\}$ is the finite set of evidence at level i, $i = 1, 2, ..., I-1$ which is used to evaluate the events in $\mathcal{E}_i$. The visual evidence space is associated with the visual event space and is the mapping results of the visual event space. A visual event can provide visual evidence for confirming or disconfirming hypotheses about other visual events. However, a piece of visual evidence may not necessarily be a valid visual event defined in the visual event hierarchy. Visual evidence is the information required for asserting other visual events. For example, a spatial relationship between two visual events may provide visual evidence although it is not a visual event defined in the hierarchy. Uncertainty reasoning will provide a method for combining multiple visual evidence into a consistent overall interpretation at each level and for filling in information in areas where no information is locally available.

Besides the visual event hierarchy, there is a hypothesis hierarchy. Let $H = \cup_{j=1}^{I}H_j$ be a hierarchical hypothesis space. H_i is a subset of H, where $i \neq I$ (*i.e.* level

i is not the top level), and $H_i = \{h_{i,1}, h_{i,2}, ..., h_{i,G_i}\}$ is a finite set of hypotheses. $h_{i,j}$ denotes the hypotheses about the jth visual event $\mathcal{G}_{i,j}$ at level i. For $1 \leq i < I$, if there are G_i hypotheses at level i, and if each hypothesis describes a possible visual event at level i, then Θ_i discerns these visual events at level i.

The finite set of hypotheses H_i at the level i can be decomposed into groups of mutually exclusive subsets, and each group can be considered to be a separate frame of discernment. Each such separate frame of discernment corresponds to a class of possible labels for valid visual events at a particular level of the visual event hierarchy. In the reasoning process at level i, we want to determine which visual event is believed to be true among a set of possibilities at that level. The frame of discernment for this case is $\Theta_i = \{h_{i,1}, h_{i,2}, ..., h_{i,G_i}\}$. A reasoning tree then can be constructed to represent this frame of discernment. In the tree, all the terminal nodes correspond to the elements of Θ_i. Higher nodes correspond to a cluster of visual events. Each corresponds to the union of the terminal nodes below it. Notice that only those subsets of Θ_i which are of semantic interest are represented in the tree. For example, to recognize a lake, a river, shadow of a man-made object, shadow of natural object, and other objects on a SAR radar image, the frame of discernment would be:

$$\Theta_i = \{\text{River, Lake, Sea, Shadow-m, Shadow-n, Others}\}.$$

All these six labels are elements of the frame of discernment. They are mutually exclusive and exhaustive at the object level. Each label represents a hypothesis corresponding to a visual event at the object level. Only one label can be true for a particular region at any one time. A tree can be constructed to represent this frame of discernment hierarchically as in Figure 9.

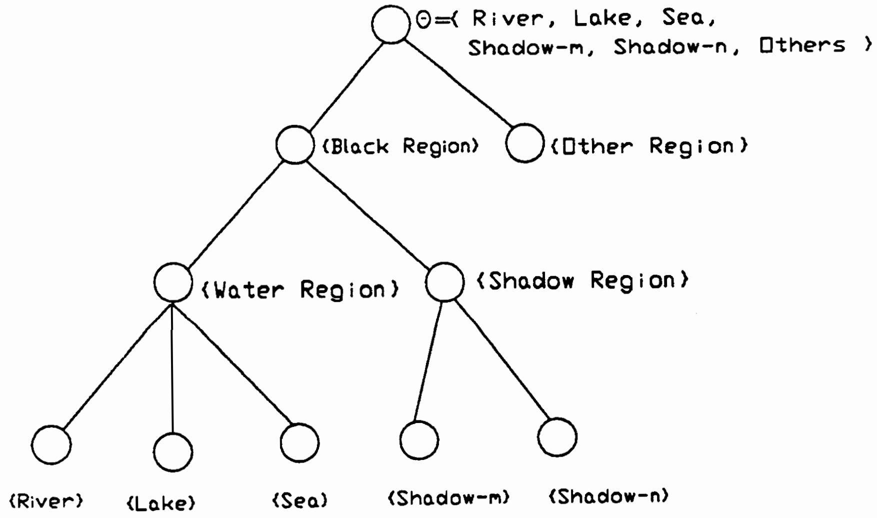


Figure 9. The hypothesis hierarchy at the object level for the water-shadow problem.

There are three levels of hierarchy in this tree. The {Black Region} is a subset of Θ_i. The {Water Region} and the { Shadow Region } are subsets of {Black Region}. {Lake} and {River} are subsets of {Water Region}. {Shadow-m} and {Shadow-n} are subsets of {Shadow}. All these nodes below Θ_i are the subsets of Θ_i. Other subsets of Θ_i which are not semantically significant to this problem are not represented in this tree. Note that this hierarchy is not a structural hierarchy but rather a "logical" hypothesis hierarchy.

One of the advantages of D-S model is that it is possible to assign belief to a subset. Even though some evidence does not specifically support a particular hypothesis, it still can support the superset of that particular hypothesis. For example, one piece of evidence suggests the {Water Region} to degree 0.7, although it suggests the {Lake Region} to degree 0.0. The uncertainty and conflict among them can be resolved when we compute the interactions among these beliefs in the tree.

The relationships between the hypothesis hierarchy and the visual event hierarchy are illustrated in Figure 10. Each reasoning tree represents an element of a partitioned hypothesis hierarchy at that level. The union of all reasoning trees at that level is the frame of discernment Θ. The reasoning results from these reasoning trees are asserted visual events at that level. These asserted visual events can be used as visual evidence for the next level visual event reasoning. Through a visual evidence collection, quantitative measurement, and visual evidence mapping process these pieces of visual evidence can mapped into evidence strengths for the hypothesis hierarchy at the next processing level.

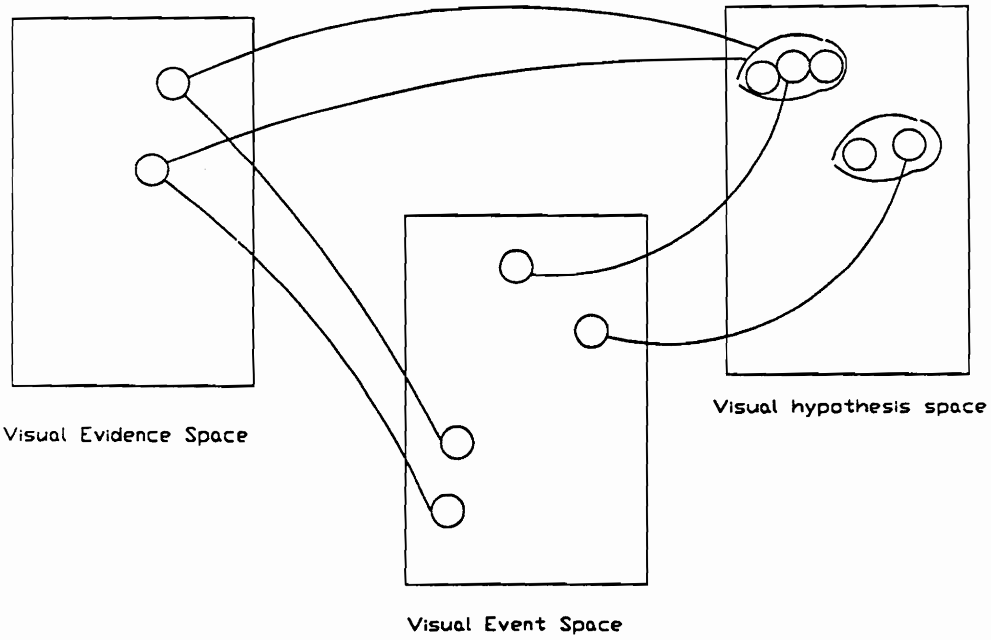


Figure 10. Relationships between hierarchical spaces.

Chapter V
THE REASONING ALGORITHM

In this chapter we propose a reasoning system to adapt the D-S model to computer vision. We direct particular attention to reasoning in a hierarchical visual evidence space while keeping the computational complexity under control. First, visual evidence collection and initial *bpa* assignment methods are introduced. Then, the interactions of hierarchical visual evidence are described. Finally, an algorithm for handling spatial relations is presented. These three processes are all based on the visual event hierarchy, the visual hypothesis hierarchy, and the reasoning tree defined in Chapter IV.

5.1 Evidence Collection and Initial bpa Assignment

After the construction of the reasoning tree described in Chapter IV, it is necessary to address the question of how to propagate the initial beliefs to each node of the tree, based upon measurements of features from image data or based upon measurements of visual events in a previous level of a visual event hierarchy. In this section, we describe a process which collects and combines a body of visual evidence to form an initial *bpa* for a hypothesis about a visual event. First, we define a set of visual evidence measurements Ω_i at level i. $\omega_{i,k} \in \Omega_i$ denotes the measurement provided by the kth processing module or provided by the kth image feature at level i. Through a mapping function, these visual evidence measurements or image feature measurements are mapped into visual evidence strengths for relevant hypotheses, which range from 0 to 1. Next, the evidence is combined by Barnet's technique into two separate belief functions. One is the evidence strength for a particular hypothesis. The other is the evidence strength against the hypothesis. Finally, by Dempster's rule, these two beliefs are combined into a single belief function with its dichotomy as the initial probability assignment for that hypothesis for the further reasoning.

Suppose there are N_i hypotheses $h_{i,1}, h_{i,2}, ..., h_{i,N_i}$ at level i of the visual event hierarchy. At level i there are also M_i pieces of evidence $e_{i,1}, e_{i,2}, ..., e_{i,M_i}$ which support or contradict each of these hypotheses. Each $e_{i,j}$ may based on visual evidence measurements $\omega_{i,j} \in \Omega_i$, which come from different processing modules at level i or from confirmatory hypotheses at level $i-1$. A mapping $V_{i,j}$ is a function which maps the visual measurement into the strengths of the visual evidence for and against the jth hypothesis in Θ_i.

Let $f_{i,k,j}$ be the strength of $e_{i,k}$ supporting the hypothesis $h_{i,j}$ and $a_{i,k,j}$ be the strength of $e_{i,k}$ against the hypothesis $h_{i,j}$. For a given $\omega_{i,k}$, we have N_i $f_{i,k,j}$ and N_i $a_{i,k,j}$ for $j = 1, 2, ..., N_i$:

$$V_{i,j} : (\omega_{i,k}) \longrightarrow (f_{i,k,j}, a_{i,k,j}) \qquad (5.1.a)$$

that is

$$\begin{pmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ V_{i,N_i} \end{pmatrix} : \begin{pmatrix} \omega_{i,1} \\ \omega_{i,2} \\ \vdots \\ \omega_{i,M_i} \end{pmatrix} \longrightarrow \begin{pmatrix} (f_{i,1,1}, a_{i,1,1}) & (f_{i,1,2}, a_{i,1,2}) & \cdots & (f_{i,1,N_i}, a_{i,1,N_i}) \\ (f_{i,2,1}, a_{i,2,1}) & (f_{i,2,2}, a_{i,2,2}) & \cdots & (f_{i,2,N_i}, a_{i,2,N_i}) \\ \vdots & \vdots & \ddots & \vdots \\ (f_{i,M_i,1}, a_{i,M_i,1}) & (f_{i,M_i,2}, a_{i,M_i,2}) & \cdots & (f_{i,M_i,N_i}, a_{i,M_i,N_i}) \end{pmatrix}$$

and

$$V_{i,\theta} : (\omega_{i,k}) \longrightarrow \left(f_{i,k,\theta} = 1 - \sum_{n=1}^{N_i} f_{i,k,n}, a_{i,k,\theta} = 1 - \sum_{n=1}^{N_i} a_{i,k,n}\right) \qquad (5.1.b)$$

$$\sum_n f_{i,k,n} = 1 \tag{5.1.c}$$

$$\sum_n a_{i,k,n} = 1 \tag{5.1.d}$$

$$f_{i,k,n} + a_{i,kn} \leq 1 \tag{5.1.e}$$

Several mapping methods which map the measurements $\omega_{i,k}$ into subjective probability assignments are now proposed. In the following mapping function, these initial *bpa*s can be obtained either from the training data, from the expertise of the expert, or from the accumulation of experience in a long run of experiments by the user. These probabilities are typically very subjective.

5.1.1 Threshold Operation Mapping

Given a threshold $T_{i,k}$ and a visual measurement $\omega_{i,k}$, we have

$$e_{i,k} = \begin{cases} 1 & \text{if } \omega_{i,k} \geq T_{i,k}; \\ 0 & \text{otherwise} \end{cases} \tag{5.2.a}$$

where $k = 1, 2, ...M_i$.

Then, for each $h_{i,n} \subseteq H_i$,

$$f_{i,k,n} = \begin{cases} P(h_{i,n}|e_{i,k}) & \text{if } e_{i,k} = 1; \\ 0 & \text{if } e_{i,k} = 0 \end{cases} \tag{5.2.b}$$

$$a_{i,k,n} = \begin{cases} P(\bar{h}_{i,n}|e_{i,k}) & \text{if } e_{i,k} = 1, \\ 0 & \text{if } e_{i,k} = 0, \end{cases} \tag{5.2.c}$$

where $n = 1, 2, ...N_i$, and $k = 1, 2, ...M_i$ for the ith hierarchy level.

For each piece of evidence with $e_{i,k} = 1$, $f_{i,k,n}$ is the strength of the evidence for supporting the hypothesis $h_{i,n}$, and $a_{i,k,n}$ is the strength of the evidence against hypothesis $h_{i,n}$ (see Figure 11(a)). If there are M_i pieces of evidence and N_i hypotheses at level i, then there will be $M_i N_i$ $f_{i,k,n}$ and $M_i N_i$ $a_{i,k,n}$ at that level.

5.1.2 Linear Mapping

The linear mapping function is shown in Figure 11(b). Let ω_{min} be a lower bound for accepting a visual event, let ω_{max} be an upper bound for that visual event, and let ω_k be the measurement value. Then

$$f_{i,k,n} = P(h_{i,n}|e_{i,k}) = \begin{cases} \frac{\omega_k - \omega_{min}}{\omega_{max} - \omega_{min}} & \text{if } \omega_{min} < \omega_k < \omega_{max} \\ 1 & \text{if } \omega_k \geq \omega_{max} \\ 0 & \text{if } \omega_k \leq \omega_{min} \end{cases} \qquad (5.3.a)$$

$$a_{i,k,n} = P(\bar{h}_{i,n}|e_{i,k}) = \begin{cases} \frac{\omega_{max} - \omega_k}{\omega_{max} - \omega_{min}} & \text{if } \omega_{min} < \omega_k < \omega_{max} \\ 1 & \text{if } \omega_k \leq \omega_{max} \\ 0 & \text{if } \omega_k \geq \omega_{min} \end{cases} \qquad (5.3.b)$$

5.1.3 Optimal Mapping

For this mapping function, there is a maximum initial *bpa* value which corresponds to a certain visual evidence measurement value. It will monotonically decrease as the value of ω diverges from that measurement value in either direction. This mapping function is illustrated in Figure 11(c).

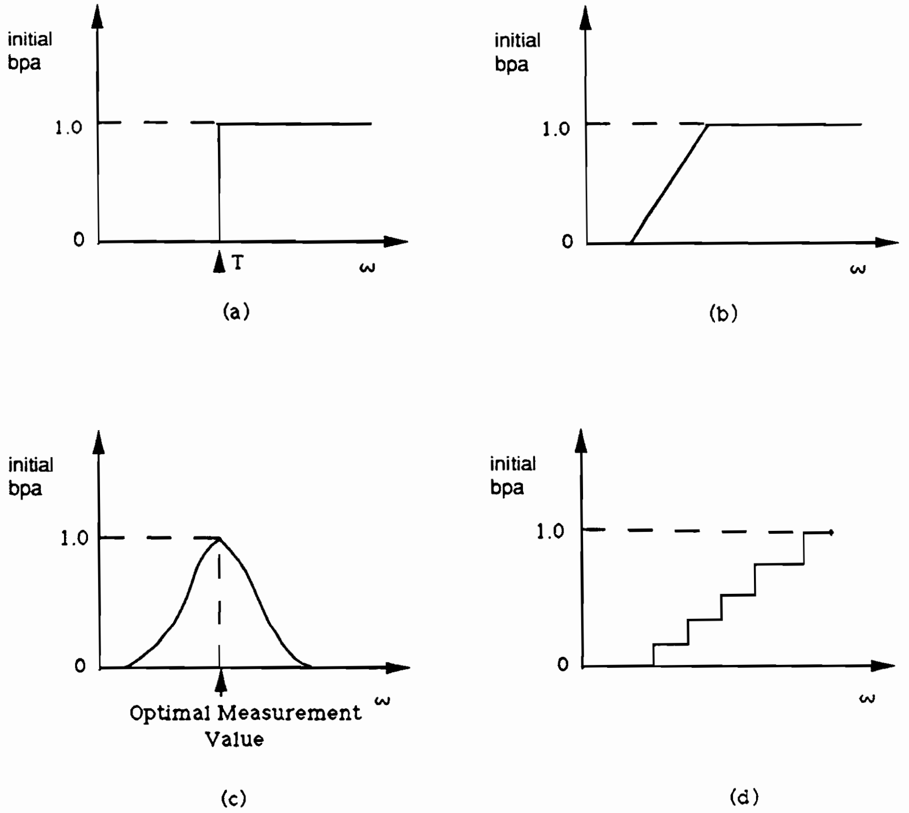


Figure 11. Mapping functions: (a) Threshold mapping (b) Linear mapping (c) Optimal mapping (d) Discrete mapping.

5.1.4 Discrete Table-look-up mapping

This mapping function may be a multiple threshold mapping function for a continue quantitative measurement. It may also be discrete mapping function for a set of logical relationships. An example is shown in Figure 11(d).

The above four mapping functions are actually part of the system knowledge base. They specify the initial degree of impact of each visual evidence measurement on the visual events. In fact, they are subjective weights for different uncertain information sources.

Since there are M_i visual evidence measurements, we have to combine all $f_{i,k,n}$ for $k = 1, 2, ..., M_i$ into $f_{i,n}$ and all $a_{i,k,n}$ into $a_{i,n}$. That is

$$f_{i,n} = m_1(h_{i,n}) = 1 - \prod_{k=1}^{M_i}(1 - f_{i,k,n}) \tag{5.4.a}$$

$$a_{i,n} = m_2(\bar{h}_{i,n}) = 1 - \prod_{k=1}^{M_i}(1 - a_{i,k,n}) \tag{5.4.b}$$

for each hypothesis $h_{i,n}$ at level i, where $n = 1, 2, ..., N_i$. $f_{i,n}$ is the total weight of the evidence in support of $h_{i,n}$, and $a_{i,n}$ is the total weight of the evidence against $h_{i,n}$.

According to the D-S model, we also have

$$m_1(\bar{h}_{i,n}) = m_2(h_{i,n}) = 0 \tag{5.5.a}$$

$$m_1(\theta_i) = 1 - f_{i,n} \tag{5.5.b}$$

$$m_2(\theta_i) = 1 - a_{i,n} \tag{5.5.c}$$

Now, the orthogonal sum of m_1 and m_2

$$m = m_1(x_i) \oplus m_2(y_i) \tag{5.6}$$

where

$$x_i \in \{h_{i,n}, \bar{h}_{i,n}, \theta_i\}$$

$$y_i \in \{h_{i,n}, \bar{h}_{i,n}, \theta_i\}$$

is computed according to Figure 12.

Thus,

$$K^{-1} = \sum_{x_i \cap y_i \neq \emptyset} m_1(x_i)m_2(y_i) = 1 - \sum_{x_i \cap y_i = \emptyset} m_1(x_i)m_2(y_i) = 1 - m_1(h_{i,n})m_2(\bar{h}_{i,n}) \tag{5.7.a}$$

$$m(h_{i,n}) = Km_1(h_{i,n})(1 - m_2(\bar{h}_{i,n})) \tag{5.7.b}$$

$$m(\bar{h}_{i,n}) = Km_2(\bar{h}_{i,n})(1 - m_1(h_{i,n})) \tag{5.7.c}$$

$$m(\theta_i) = K(1 - m_1(h_{i,n}))(1 - m_2(h_{i,n})) \tag{5.7.d}$$

where K^{-1} is the normalization factor, $m(h_{i,n})$ is the measure of support for $h_{i,n}$, $m(\bar{h}_{i,n})$ is the measure of evidence against $h_{i,n}$, and $m(\theta_i)$ is called the uncommitted belief. From the D-S model,

$$m(h_{i,n}) + m(\bar{h}_{i,n}) + m(\theta_i) = 1. \tag{5.7.e}$$

By this technique, we can compute the initial beliefs, which are exactly committed to each node on the hierarchy, bearing all visual evidence under consideration.

$m_1 \oplus m_2$ m_1 m_2	$m_1(h_{i,n})$	$m_1(\bar{h}_{i,n})$	$m_1(\Theta_i)$
$m_2(\bar{h}_{i,n})$	$\{\emptyset\}$	0	$\{\bar{h}_{i,n}\}$
$m_2(h_{i,n})$	0	0	0
$m_2(\Theta_i)$	$\{h_{i,n}\}$	0	$\{\Theta_i\}$

Figure 12. The orthogonal sum of m_1 and m_2.

To illustrate the algorithm, a simple example of discriminating water regions from shadow regions on an SAR radar image (Qian, 1985) is given. Suppose that the processing level i corresponds to the region level. The visual evidence measurement set $\Omega_i = \{\omega_{i,1}, \omega_{i,2}, \omega_{i,3}\}$ = {greytone measurement, texture measurement, spatial relation measurement between regions}. The frame of discernment is Θ_i = {Water region, Shadow region, Others}. The hypothesis set $H_i \subseteq \Theta_i$ and $H_i = \{h_{i,1}, h_{i,2}, h_{i,3}, h_{i,4}\}$ = {Black region, Others, Water region, Shadow region}. Note that only the interesting subsets are chosen here. A corresponding reasoning tree for this hypothesis hierarchy is shown in Figure 13. For the given Ω_i, the evidence strength for each hypothesis is mapped by the mapping functions to obtain the initial bpa attached to each node in the tree.

Based upon the greytone measurement $\omega_{i,1}$, we have the following initial $bpa's$:

$$f_{i,1,1} = 0.4 \qquad a_{i,1,1} = 0.3$$

$$f_{i,1,2} = 0.1 \qquad a_{i,1,2} = 0.2$$

$$f_{i,1,3} = 0.4 \qquad a_{i,1,3} = 0.3$$

$$f_{i,1,4} = 0.1 \qquad a_{i,1,4} = 0.2$$

Based upon the textural measurement $\omega_{i,2}$, we have another set of initial $bpa's$:

$$f_{i,2,1} = 0.3 \qquad a_{i,2,1} = 0.3$$

$$f_{i,2,2} = 0.1 \qquad a_{i,2,2} = 0.3$$

$$f_{i,2,3} = 0.3 \qquad a_{i,2,3} = 0.2$$

$$f_{i,2.4} = 0.2 \qquad a_{i,2,4} = 0.2$$

The third measurement $\omega_{i,3}$ suggests the following set of initial $bpa's$:

$$f_{i,3,1} = 0.3 \qquad a_{i,3,1} = 0.1$$

$$f_{i,3,2} = 0.1 \qquad a_{i,3,2} = 0.3$$

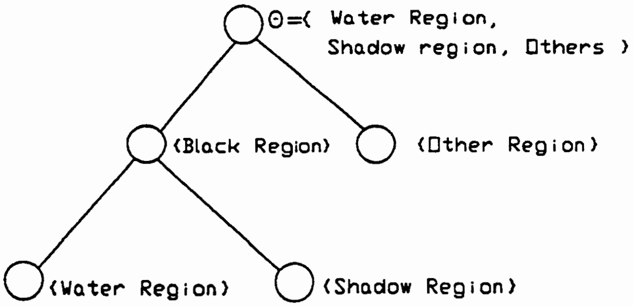


Figure 13. The hypothesis hierarchy in the example.

$$f_{i,3,3} = 0.4 \qquad a_{i,3,3} = 0.2$$

$$f_{i,3.4} = 0.1 \qquad a_{i,3,4} = 0.3$$

By combining three pieces of evidence, the mass functions which support the hypotheses $h_{i,j}$, $j = 1, 2, 3, 4$ are

$$m_1(h_{i,1}) = 0.71 \qquad m_2(\bar{h}_{i,1}) = 0.56$$

$$m_1(h_{i,2}) = 0.27 \qquad m_2(\bar{h}_{i,2}) = 0.61$$

$$m_1(h_{i,3}) = 0.75 \qquad m_2(\bar{h}_{i,3}) = 0.55$$

$$m_1(h_{i,4}) = 0.35 \qquad m_2(\bar{h}_{i,4}) = 0.55$$

Now, the orthogonal sum of m_1 and m_2 is computed from Equations 4.1.4a–d as follows.

$$m(h_{i,1}) = 0.51 \qquad m(\bar{h}_{i,1}) = 0.27 \qquad m(\theta_i) = 0.22$$

$$m(h_{i,2}) = 0.13 \qquad m(\bar{h}_{i,2}) = 0.53 \qquad m(\theta_i) = 0.34$$

$$m(h_{i,3}) = 0.57 \qquad m(\bar{h}_{i,3}) = 0.24 \qquad m(\theta_i) = 0.19$$

$$m(h_{i,4}) = 0.20 \qquad m(\bar{h}_{i,4}) = 0.44 \qquad m(\theta_i) = 0.34$$

These dichotomies are attached to each node $h_{i,j}$ in the tree as the initial beliefs which are committed exactly to each node of the tree for further computation.

5.2 The Interaction of Hierarchical Evidence

The interaction of hierarchical evidence is considered in this section. For a computer vision system, this kind of interaction should be taken into account to resolve the conflicts and ambiguity based upon incomplete and uncertain information from multiple visual measurements. It often occurs that one visual event itself may have weaker supporting evidence or even may not be directed supported by the visual evidence. However, its superset or its subsets may have stronger supporting evidence. The interaction of hierarchical evidence can provide a way to deduce the visual event indirectly from other relevant hypotheses. In this section, first some concepts and notation are introduced. Then, a more detailed description and derivation of Shafer's algorithm is given. Finally, an example is given to illustrate the theory.

Let A be any node in the reasoning tree which corresponds to a conceptual hierarchy of relevant visual events at level i of the visual event hierarchy, as described in Chapter IV. After the initial *bpa* assignment computation in the previous section, all dichotomous belief functions Bel_A with dichotomy $\{A, \bar{A}\}$ are attached to each node A of the tree. Note that the Bel_A is the belief that exactly commits to the node A, not to a subset of A. That is, Bel_A is expressed as $Bel_A(A)$ and $Bel_A(\bar{A})$. Now we need to compute the interaction of hierarchical evidence to resolve the conflicts and uncertainty which result from the initial *bpa* assignment computation. This requires the calculation of the belief function for a particularly interesting hypothesis considering all the evidence in the frame of discernment. In other words, it is necessary to calculate the value of $Bel^{\downarrow}_{\theta}(P)$ for P in the tree, where P is a node corresponding to a particularly interesting hypothesis.

For simplicity, the following notation due to Shafer is used (Shafer and Logan, 1987):

$Bel^{\downarrow}_{A}$: The orthogonal sum of Bel_B for all nodes B that are strictly below A.

$Bel^{\diamond}_{A}$: The orthogonal sum of Bel_B for all nodes B that are neither below A nor equal to A.

Bel_A: A single dichotomous belief function with the dichotomy $\{A, \bar{A}\}$ for some node A in the tree.

$A_0^+ = Bel_A(A)$: The initial *bpa* for A attached to node A. It is the belief that exactly commits to A.

$A_0^- = Bel_A(\bar{A})$: The initial *bpa* for $\bar{A}$ attached to node A. It is the belief that exactly commits to $\bar{A}$.

$A_\downarrow^+ = Bel_A^\downarrow(A)$: The belief value for A under the consideration of all evidence for A below the node A in the tree.

$A_\downarrow^- = Bel_A^\downarrow(\bar{A})$: The belief value for $\bar{A}$ under the consideration of all evidence for $\bar{A}$ below the node A in the tree.

$A^+ = (Bel_A \oplus Bel_A^\downarrow)(A)$: The orthogonal sum of the belief value that exactly commits to A and the belief value for A under the consideration of all evidence for A in the nodes below A in the tree.

$A^- = (Bel_A \oplus Bel_A^\downarrow)(\bar{A})$: The orthogonal sum of the belief value that exactly commits to $\bar{A}$ and the belief value for $\bar{A}$ under the consideration of all evidence for $\bar{A}$ in the nodes below the node A in the tree.

$A_\diamond^+ = (Bel_A \oplus Bel_A^\diamond(A)$: The orthogonal sum of the belief value that exactly commits to A and the belief value for A under the consideration of all evidence for A in the nodes neither below A nor equal to A in the tree.

$A_\diamond^- = (Bel_A \oplus Bel_A^\diamond(\bar{A})$: The orthogonal sum of the belief value that exactly commits to $\bar{A}$ and the belief value for $\bar{A}$ under the consideration of all evidence for $\bar{A}$ in the nodes neither below A nor equal to A in the tree.

$A^+_\theta = Bel^\downarrow_\theta(A)$: The belief value for A under the consideration of all evidence for A in the nodes below θ.

$A^-_\theta = Bel^\downarrow_\theta(\bar{A})$: The belief value for $\bar{A}$ under the consideration of all evidence for $\bar{A}$ in the nodes below θ.

For each node B other than Θ and its daughters, we have the following definitions:

$$B^+_A = Bel^\downarrow_A(B)$$

$$B^-_A = Bel^\downarrow_A(\bar{B})$$

$$B^*_A = Bel^\downarrow_A(B \cup \bar{A})$$

The next goal is to compute $Bel^\downarrow_\theta(P)$ for all hypotheses P of interest. There are two cases when $Bel^\downarrow_\theta(P)$ is computed. If the node P is a particular daughter of Θ, then only one bottom-up pass is needed. If the node P is other than the daughter of Θ, then an additional top-down computation is needed after the bottom-up process. As the node P moved from the daughters of Θ to the terminal nodes of the tree, the hypothesis set is narrowed down into smaller and smaller subsets until it reaches the singleton hypothesis. We will analyze these two cases next.

5.2.1 The bottom-up process:

Let φ_A denote all daughters of A in the tree. Beginning with the terminal nodes of the tree, $A^+_\downarrow$ and $A^-_\downarrow$ are calculated from B^+ and B^- for B in φ_A as in Figure 14.

From Equation 3.15.b we have

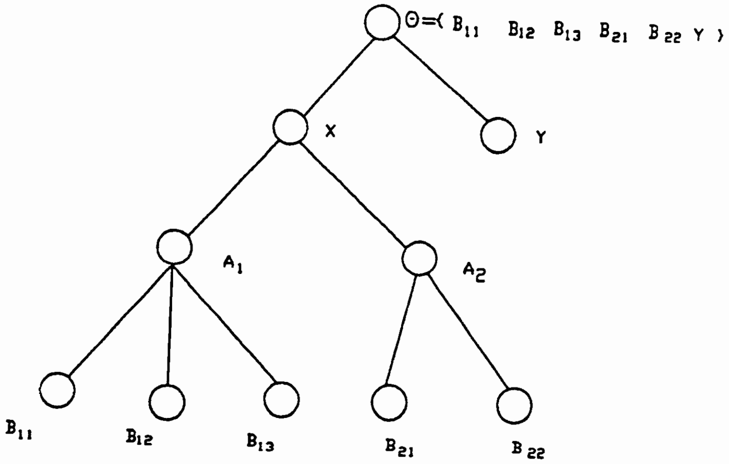


Figure 14. The bottom-up process 1.

$$K^{-1} = 1 + \sum_{B \in \varphi_A} \frac{B^+}{1 - B^+} - \prod_{B \in \varphi_A} \frac{B^-}{1 - B^-}.$$

Since B is a daughter of A and $\bar{B}$ is not a daughter of A, the last term on the right side of the above equation equals zero. Thus

$$K^{-1} = 1 + \sum_{B \in \varphi_A} \frac{B^+}{1 - B^+}. \tag{5.8}$$

$$\begin{aligned} A_{\downarrow}^{+} &= Bel_A^{\downarrow}(A) \\ &= 1 - Pl_A^{\downarrow}(\bar{A}) \\ &= 1 - K\Big(1 + \sum_{B \in \varphi_A B \not\subseteq A} \frac{B^+}{1 - B^+} - \prod_{B \in \varphi_A B \not\subseteq A} \frac{B^-}{1 - B^+}\Big) \end{aligned} \tag{5.9.a}$$

Note that the last two terms in the right side of the above equation are zero. Thus

$$A_{\downarrow}^{+} = 1 - K \tag{5.9.b}$$

$$\begin{aligned} A_{\downarrow}^{-} &= Bel_A^{\downarrow}(\bar{A}) \\ &= 1 - Pl_A^{\downarrow}(A) \\ &= 1 - K\Big(1 + \sum_{B \in \varphi_A} \frac{B^+}{1 - B^+} - \prod_{B \in \varphi_A} \frac{B^-}{1 - B^+}\Big) \end{aligned} \tag{5.10.a}$$

Substituting Equation 4.8 into the above equation, we have

$$A_{\downarrow}^{-} = K \prod_{B \in \varphi} \frac{B^-}{1 - B+} \tag{5.10.b}$$

Now the orthogonal sum of the belief for A at the node A and the belief for A below the node A will be computed. That is the calculation of A^+ and A^- from A_0^+, A_0^-, $A_\downarrow^+$, and $A_\downarrow^-$ as in Figure 15.

First, the normalization factor in this case is

$$K^{-1} = 1 - \sum_{\cap S_i = \phi} m_i(S_i) = 1 - A_0^+ A_\downarrow^- - A_0^- A_\downarrow^+ \tag{5.11}$$

where A_0^+ focuses on A, $A_\downarrow^-$ focuses on $\bar{A}$, and $A_0^+ A_\downarrow^-$ focuses on $A \cap \bar{A} = \phi$. For the same reason, $A_0^- A_\downarrow^+$ focuses on $\bar{A} \cap A = \phi$.

$$\begin{aligned} A^+ &= (Bel_A \oplus Bel_A^\downarrow)(A) \\ &= 1 - (Pl_A \oplus Pl_A^\downarrow)(\bar{A}) \\ &= 1 - K\{Q_A(X) Q_A^\downarrow(X) | X \subseteq \bar{A}\} \\ &= 1 - K Pl_A(\bar{A}) Pl_A^\downarrow(\bar{A}) \\ &= 1 - K(1 - Bel_A(A))(1 - Bel_A^\downarrow(A)) \\ &= 1 - K(1 - A_0^+)(1 - A_\downarrow^+) \end{aligned} \tag{5.12.a}$$

In the same way, we have

$$A^- = 1 - K(1 - A_0^-)(1 - A_\downarrow^-) \tag{5.12.b}$$

The above two steps are repeatedly applied all the way up the tree until the daughters A of Θ are reached. Then we can calculate A_θ^+ and A_θ^- from A^+ and A^- for A in the set of daughters of Θ.

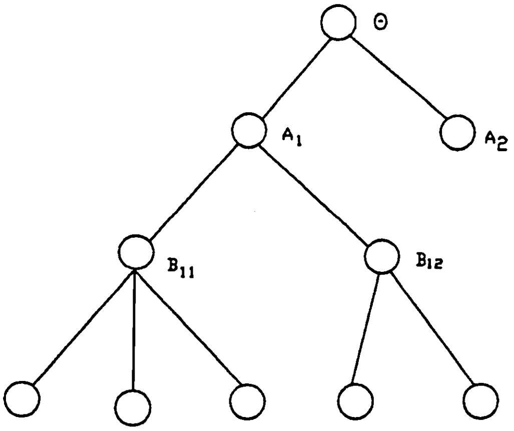


Figure 15. The bottom-up process 2.

$$K^{-} = 1 + \sum_{B \in \varphi_\theta} \frac{B^{+}}{(1-B)} - \prod_{B^{+} \in \varphi_\theta} \frac{B^{-}}{(1-B+)} \tag{5.13}$$

$$\begin{aligned} A_\theta^{+} &= Bel_\theta^{\downarrow}(A) \\ &= 1 - Pl_\theta^{\downarrow}(\bar{A}) \\ &= 1 - K\Big(1 + \sum_{B \in \varphi_\theta, B \neq A} \frac{B^{+}}{(1-B^{+})} - \prod_{B \in \varphi_\theta, B \neq A} \frac{B^{-}}{(1-B^{+})}\Big) \end{aligned} \tag{5.14}$$

$$\begin{aligned} A_\theta^{-} &= Bel_\theta^{\downarrow}(\bar{A}) \\ &= 1 - Pl_\theta^{\downarrow}(A) \\ &= 1 - K\Big(1 + \frac{A^{+}}{1-A^{+}} - \frac{A^{-}}{1-A^{+}}\Big) \\ &= 1 - K\frac{1-A^{-}}{1-A^{+}} \end{aligned} \tag{5.15}$$

That completes the bottom-up computation. For the case that P is a daughter of Θ $Bel_\theta^{\downarrow}(P)$ has been calculated considering the interactions of all other evidence in the frame of discernment. In most cases, however, P is a node other than the daughter of Θ. Then the top-down process is needed for the calculation of $Bel_\theta^{\downarrow}(P)$.

5.2.2 The top-down process:

Beginning with the nodes which are granddaughters of Θ, the B_A^{+}, B_A^{-}, and B_A^{*} are computed from C^{+} and C^{-} for $C \in \varphi_A$ as in Figure 16.

Following the derivation of Equation 4.8, the normalization factor in this case is

$$K^{-1} = 1 + \sum_{C \in \varphi_A} \frac{C^{+}}{1-C^{+}} \tag{5.16}$$

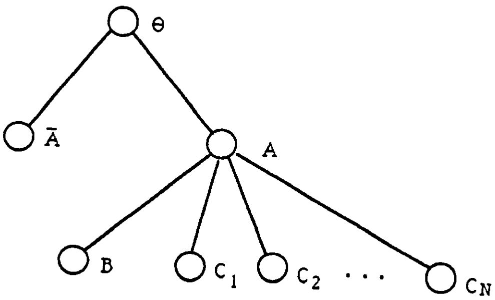


Figure 16. The top-down process 1.

The belief value for B below A is

$$\begin{aligned} B_A^+ &= Bel_A^{\downarrow}(B) \\ &= 1 - Pl_A^{\downarrow}(\bar{B}) \\ &= 1 - K\Big(1 + \sum_{C \in \varphi_A, C \neq B} \frac{C^+}{1 - C^+} - \prod_{C \in \varphi_A, C \neq B} \frac{C^-}{1 - C^+}\Big). \end{aligned} \tag{5.17.a}$$

Since C and B both are daughters of A and $\cap \bar{C}_i \neq \phi$ for all i, the last term on the right side of the above equation does not exist. Thus

$$B_A^+ = 1 - K\Big(1 + \sum_{C \in \varphi_A, C \neq B} \frac{C^+}{1 - C^+}\Big). \tag{5.17.b}$$

The belief value for $\bar{B}$ below A is

$$\begin{aligned} B_A^- &= Bel_A^{\downarrow}(\bar{B}) \\ &= 1 - Pl_A^{\downarrow}(B) \\ &= 1 - K\Big(1 + \frac{B^+}{1 - B^+} - \frac{B^-}{1 - B^+}\Big) \\ &= 1 - K\Big(1 + \frac{B^+ - B^-}{1 - B^+}\Big) \\ &= 1 - K\frac{1 - B^-}{1 - B^+}. \end{aligned} \tag{5.18}$$

By using the partition shown in Figure 17 we have,

$$\begin{aligned} B_A^* &= Bel_A^{\downarrow}(B \cup \bar{A}) \\ &= 1 - Pl_A^{\downarrow}(B \bar{\cup} \bar{A}) \\ &= 1 - Pl_A^{\downarrow}(C) \\ &= 1 - K\Big(1 + \sum_{C \in \varphi_A, C \neq B} \frac{C^+}{1 - C^+} - \prod_{C \in \varphi_A, C \neq B} \frac{C^-}{1 - C^+}\Big) \end{aligned} \tag{5.19}$$

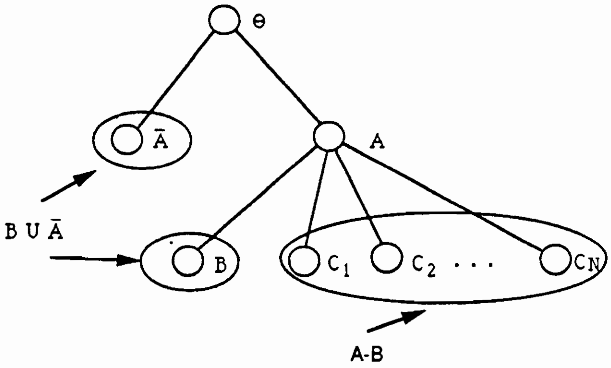


Figure 17. The partition of $\{B, A-B, \bar{A}\}$ in the top-down process 2.

Before the calculation of $B^{\downarrow}_{\theta}$, $A^{+}_{\diamond}$ and $A^{-}_{\diamond}$ should be computed from A^{+}_{θ}, A^{-}_{θ}, $A^{+}_{\downarrow}$, and $A^{-}_{\downarrow}$. Note that

$$A^{+}_{\diamond} = (Bel_{A} \oplus Bel^{\diamond}_{A}(A)) \tag{5.20.a}$$

$$Bel^{\diamond}_{A} = \oplus Bel_{B} \tag{5.20.b}$$

where B is neither below A nor equal to A.

From the D-S model, the commonality function $Q(X)$ has the following values:

for $X = \{A\}$ and $|X| = 1$

$$Q^{\downarrow}_{\theta}(X) = 1 - Bel^{\downarrow}_{\theta}(\bar{A}) = 1 - A^{-}_{\theta}\,, \tag{5.21.a}$$

$$Q^{\downarrow}_{A}(X) = 1 - Bel^{\downarrow}_{A}(\bar{A}) = 1 - A^{-}_{\downarrow}\,; \tag{5.21.b}$$

for $X = \{\bar{A}\}$ and $|X| = 1$

$$Q^{\downarrow}_{\theta}(X) = 1 - Bel^{\downarrow}_{\theta}(A) = 1 - A^{+}_{\theta}\,, \tag{5.22.a}$$

$$Q^{\downarrow}_{A}(X) = 1 - Bel^{\downarrow}_{A}(A) = 1 - A^{+}_{\downarrow}\,; \tag{5.22.b}$$

for $\{A\} \in X$ and $|X| > 1$

$$Q^{\downarrow}_{\theta}(X) = 1 - Bel^{\downarrow}_{\theta}(A) - Bel^{\downarrow}_{\theta}(\bar{A}) = 1 - A^{+}_{\theta} - A^{-}_{\theta}, \tag{5.23.a}$$

$$Q^{\downarrow}_{A}(X) = 1 - Bel^{\downarrow}_{A}(A) - Bel^{\downarrow}_{A}(\bar{A}) = 1 - A^{+}_{\downarrow} - A^{-}_{\downarrow}. \tag{5.23.b}$$

Let

$$Pl^{+}_{\diamond} = (Pl_{A} \oplus Pl^{\diamond}_{A})(A) \tag{5.24.a}$$

$$Pl^{-}_{\diamond} = (Pl_{A} \oplus Pl^{\diamond}_{A})(\bar{A}) \tag{5.24.b}$$

Then

$$\begin{aligned} Pl^{-}_{\diamond} &= KQ^{-}_{\diamond} \\ &= KQ_{A}(\bar{A})Q^{\diamond}_{A}(\bar{A}) \\ &= KQ_{A}(\bar{A})Q^{\diamond}_{A}(\bar{A})\frac{Q^{\downarrow}_{A}(\bar{A})}{Q^{\downarrow}_{A}(\bar{A})} \\ &= K\frac{Q^{\downarrow}_{\theta}(\bar{A})}{Q^{\downarrow}_{A}(\bar{A})} \\ &= K\frac{1 - A^{+}_{\theta}}{1 - A^{+}_{\downarrow}}. \end{aligned} \tag{5.25}$$

Similarly,

$$\begin{aligned} Pl^{+}_{\diamond} &= KQ^{+}_{\diamond} \\ &= K\frac{1 - A^{-}_{\theta}}{1 - A^{-}_{\downarrow}}. \end{aligned} \tag{5.26}$$

It is obvious that

$$Q^{-}_{\diamond} = \frac{1 - A^{+}_{\theta}}{1 - A^{+}_{\downarrow}} \tag{5.27.a}$$

$$Q^{+}_{\diamond} = \frac{1 - A^{-}_{\theta}}{1 - A^{-}_{\downarrow}}. \tag{5.27.b}$$

Now $Pl^{\theta}_{\diamond}$ is computed

$$\begin{aligned} Pl^{\theta}_{\diamond} &= (Pl_A \oplus Pl^{\diamond}_{A})(\theta) \\ &= KQ^{\theta}_{\diamond} \\ &= KQ_A(\theta)Q^{\diamond}_{A}(\theta) \\ &= KQ_A(\theta)Q^{\diamond}_{A}(\theta)\frac{Q^{\downarrow}_{A}(\theta)}{Q^{\downarrow}_{A}(\theta)} \\ &= K\frac{Q^{\downarrow}_{\theta}(\theta)}{Q^{\downarrow}_{A}(\theta)} \\ &= K\frac{1 - A^{+}_{\theta} - A^{-}_{\theta}}{1 - A^{+}_{\downarrow} - A^{-}_{\downarrow}}. \end{aligned} \tag{5.28.a}$$

That is

$$Q^{\theta}_{\diamond} = \frac{1 - A^{+}_{\theta} - A^{-}_{\theta}}{1 - A^{+}_{\downarrow} - A^{-}_{\downarrow}}. \tag{5.28.b}$$

From the equation

$$Pl(\theta) = 1 = \sum\{(-1)^{|X|+1}K\prod Q(X)|\phi \neq X \subseteq \theta\},$$

the normalization factor for this case is obtained

$$\begin{aligned} K^{-1} &= \sum\{(-1)^{|X|+1} \prod Q_{\diamond}(X) | \phi \neq X \subseteq \theta\} \\ &= Q_{\diamond}^{-}(\bar{A}) + Q_{\diamond}^{+}(A) - Q_{\diamond}^{\theta}(\theta) \\ &= \frac{1 - A_{\theta}^{+}}{1 - A_{\downarrow}^{+}} + \frac{1 - A_{\theta}^{-}}{1 - A_{\downarrow}^{-}} - \frac{1 - A_{\theta}^{+} - A_{\theta}^{-}}{1 - A_{\downarrow}^{+} - A_{\downarrow}^{-}} \end{aligned} \tag{5.29.a}$$

together with the belief functions

$$A_{\diamond}^{+} = 1 - Pl_{\diamond}^{-} = 1 - K\frac{1 - A_{\theta}^{+}}{1 - A_{\downarrow}^{+}} \tag{5.29.b}$$

$$A_{\diamond}^{-} = 1 - Pl_{\diamond}^{+} = 1 - K\frac{1 - A_{\theta}^{-}}{1 - A_{\downarrow}^{-}} \tag{5.29.c}$$

Finally, the belief function B_{θ}^{+}, B_{θ}^{-} can be computed from $A_{\diamond}^{+}$, $A_{\diamond}^{-}$, $A_{\downarrow}^{+}$, $A_{\downarrow}^{-}$, B_{A}^{+}, B_{A}^{-} and B_{A}^{*}, since B is a daughter of A:

$$K^{-1} = 1 - \sum_{X \cap Y = \phi} m_1(X) m_2(Y) = 1 - A_{\downarrow}^{+} A_{\diamond}^{-} - A_{\downarrow}^{-} A_{\diamond}^{+} \tag{5.30}$$

Note that $A_{\downarrow}^{+}$ and $A_{\diamond}^{+}$ focus on A, $A_{\diamond}^{-}$ and $A_{\diamond}^{-}$ focus on $\bar{A}$, $A_{\downarrow}^{+} A_{\diamond}^{-}$ and $A_{\downarrow}^{-} A_{\diamond}^{+}$ focus on $\bar{A} \cap A = \phi$.

$$B_{\theta}^{+} = K \sum_{X \cap Y \subseteq B} m_1(X) m_2(Y) = K(A_{\diamond}^{+}(B_{A}^{*} - A_{\downarrow}^{-}) + (1 - A_{\diamond}^{+} - A_{\diamond}^{-}) B_{A}^{+}) \tag{5.31}$$

Note that $A_{\diamond}^{+}$ focuses on A, $(B_{A}^{*} - A_{\downarrow}^{-})$ focuses on $(B \cup \bar{A} - \bar{A})$, $A_{\diamond}^{+}(B_{A}^{*} - A_{\downarrow}^{-})$ focuses on $A \cap (B \cup \bar{A} - \bar{A}) = B$, $(1 - A_{\diamond}^{+} - A_{\diamond}^{-})$ focuses on θ, B_{A}^{+} focuses on B, and $(1 - A_{\diamond}^{+} - A_{\diamond}^{-}) B_{A}^{+}$ focuses on $\theta \cap B = B$. This is shown in Figure 17.

$$\begin{aligned} B_\theta^- &= 1 - Pl_\theta(B) \\ &= 1 - KQ_\Diamond^+(A)Q_A^\downarrow(B) \qquad (5.32) \\ &= 1 - K(1 - A^- \Diamond)(1 - B_A^-) \end{aligned}$$

In the following, the example used in previous sections is continued to illustrate the above computations. At the end of Section 5.1, all dichotomous belief functions Bel_P with dichotomy $\{P, \bar{P}\}$ are attached to each node P of the tree (Figure 13). Suppose we want to know whether the region is a water region or a shadow region considering the interactions of all the visual evidence. This will require computing belief values for B_1, $\bar{B}_1$, B_2, and $\bar{B}_2$, considering all evidence for these nodes below Θ. That is, we want to compute $Bel_\theta^\downarrow(B_1)$, $Bel_\theta^\downarrow(B_2)$, $Bel_\theta^\downarrow(\bar{B}_1)$, and $Bel_\theta^\downarrow(\bar{B}_2)$. Since the nodes B_1 and B_2 are their terminal nodes in the tree, to compute the above belief value requires both the bottom-up and the top-down processes.

In the bottom-up process, first the belief value for the black region considering all evidence for the black region below the node A_1 (Black region) is computed. The same is done for the node A_2 (Other regions).

By Equations 4.2.1a-c,

$$K^{-1} = 0.39 \qquad (A_1)_\downarrow^+ = 0.61 \qquad (A_1)_\downarrow^- = 0.12$$

Since the node A_2 is a terminal node, it is not necessary to compute the belief value below that node. The orthogonal sum of the beliefs is the same as the initial *bpa* that attached to the node A_2 previously. Now the orthogonal sum of the belief for the Black region at the node A_1 and the belief for the black region below the node A_1 will be computed by Equations 4.2.2a-c.

$$K^{-1} = 1.29 \qquad {A_1}^+ = 0.75 \qquad {A_1}^- = 0.17$$

Note that nodes A_1 and A_2 are daughters of Θ in this example. Then we can promptly calculate A_θ^+ and A_θ^- from A^+ and A^- for A in the set of daughters of Θ. Otherwise, it is necessary to repeat above two steps all the way up to the daughters of Θ.

By Equations 4.2.3a-c,

$$K^- = 0.26 \qquad (A_1)_\theta^+ = 0.86 \qquad (A_1)_\theta^- = 0.11$$

Because $Bel_\theta^\downarrow(A_1)$=0.86 and because it is impossible that two disjoint sets both have degree of belief greater than 0.5, it is not necessary to compute $Bel_\theta^\downarrow(A_2)$. That completes the bottom-up process. The Black region has a dominating belief value. However, the goal is to discriminate water regions the shadow regions. That is, P is a node other than the daughter of Θ. Then the top-down process is needed for the calculation of $Bel_\theta^\downarrow(P)$.

Beginning with granddaughters of Θ, the B_A^+,B_A^- and B_A^* are computed from C^+ and C^- for $C \in \varphi_A$. Where $A = A_1$, $C = B_2$ for $B = B_1$, and $C = B_1$ for $B = B_2$ in this example. The normalization factor is given by Equations 4.2.4a is $K^{-1} = 0.39$. The belief values for B below A is given by Equations 4.2.4b-d. For the water region, $B = B_1$ and $C = B_2$, we have

$$(B_1)_A^+ = 0.51 \qquad (B_1)_A^- = 0.31 \qquad (B_1)_A^* = 0.73$$

For the shadow region, $B = B_2$ and $C = B_1$, we have

$$(B_2)_A^+ = 0.10 \qquad (B_2)_A^- = 0.73 \qquad (B_2)_A^* = 0.31$$

Before the calculation of $B_\theta^\downarrow$, $A_\diamond^+$ and $A_\diamond^-$ should be computed from A_θ^+, A_θ^-, $A_\downarrow^+$, and $A_\downarrow^-$, where $A = A_1$ in this example.

The normalization factor for this case from Equation 4.2.5a is $K^{-1} = 0.80$. Belief functions are obtained from Equations 4.2.5b-c:

$$A_{\diamond}^{+} = 0.71 \qquad A_{\diamond}^{-} = 0.20$$

Finally, the belief function B_{θ}^{+}, B_{θ}^{-} for $B = B_1$ and $B = B_2$ are computed by Equations 4.2.6a-c, respectively. Here $K^{-1} = 1.26$.

For the water region, $B = B_1$, we have

$$(B_1)_{\theta}^{+} = 0.60 \qquad (B_1)_{\theta}^{-} = 0.31$$

For the shadow region, $B = B_2$, we have

$$(B_2)_{\theta}^{+} = 0.18 \qquad (B_2)_{\theta}^{-} = 0.73$$

5.3 Spatial Relationship Handling

It is important to emphasize that spatial relationships among visual events are important visual evidence for asserting visual events at the next representation level. In the visual event hierarchy, each visual event at level $i + 1$ consists of structural descriptions of a set of relevant visual events at the level i and their interrelationships. A hypothesis for a visual event might be verified by an exact match between a stored model and observed data in an ideal world without noise and uncertainty. In the real world, however, it often occurs in a matching process that some components and some relationships are missing due to uncertain information from lower level processing. To handle this problem, it is necessary to have a measure for such uncertainty and inexactness for the imperfect matching. Inexact matching has been addressed by Shapiro and Haralick (Shapiro and Haralick, 1981). In this section, we develop an algorithm to measure structural matching and to handle spatial relationships by using the D-S formalism. First the whole structural description of the visual event at level $i+1$ is decomposed into K pairs of visual events at level i. A method for determining the spatial relationship between each pair of visual events are given. Then each valid spatial relationship is treated as separate visual evidence for or against a hypothesis for that event at level $i+1$. Each such spatial relationship contributes a partial belief to the hypothesized visual event. These beliefs are combined into a single belief function, which reflects the degree of belief in the hypothesis in terms of spatial relation aspect, through Dempster's rule. This belief, in fact, is a confidence measure in the inexact match. Finally, the belief will be combined with other visual evidence measurements for that event at the level $i+1$. Thus numerous subsidiary reasoning problems involving spatial relationships may be solved at each level of the event hierarchy.

Suppose there are K basic components of the jth visual event $\mathcal{E}_{i+1,j}$ in a predefined visual event hierarchy. These components correspond to K visual events at level i. There will be $\binom{K}{2}$ pairs of interrelationships among these K events, denoted by $\{R_{j,k} : (\mathcal{E}_{i,j}, \mathcal{E}_{i,k}) | j \neq k, j = 1, \ldots, K, k = 1, \ldots, K\}$. Suppose S visual events have been asserted at level i in the previous uncertainty reasoning, where $S \leq K$. There will be $\binom{S}{2}$ pairs of interrelationships among these S events, denoted by $\{R_{j,k} : (\mathcal{E}_{i,j}, \mathcal{E}_{i,k}) | j \neq k, j = 1, \ldots, S, k = 1, \ldots, S\}$. Not every visual event in the predefined visual hierarchy has a corresponding event in the reasoning data since some of the events might be missing in previous processing. The same is true of the interrelationships. Also, not all components and their interrelationships have the same impact on the hypothesis about a visual event at the next level. Some components are more important than others, and some relationships are more important than others. Therefore, different evidence strengths can be assigned to different confirmed spatial relationships.

To collect the spatial relationship evidence for a hypothesis about a visual event at the next level, a two-step method is presented. The first step is to determine the spatial relationship between a pair of visual events. The second step is to determine if there is a corresponding spatial relationship in the predefined visual event hierarchy. If there is one, then that spatial relationship evidence is true and an initial *bpa* is assigned to that evidence. Otherwise, the evidence is false, and no *bpa* is assigned for that relationship.

An example of the frames of discernment for interrelationships between each pair of visual events at a certain level is shown in Figure 18. There are five levels in the interrelationship hierarchy. At the first level, the frame of discernment is Θ_s = {spatial relations between two lines, spatial relations between two regions}. At the second level, Θ_s = {one line parallels the other, one line does not parallel the other, one region surrounds the other, one region is adjacent to the other}. At the third level, Θ_s = {two lines far apart are parallel, two closer lines are parallel, two lines are not parallel and are separate, two lines are not parallel and intersect, region A surrounds region B, region A is surrounded by region B, region A is above region B, region A is below region B, region A is left to region B, region A is right to region B, region A is above-right to region B, region A is above-left to region B, region A is below-right to region B, region A is below-left to region B}. In the same way, in Figure 18 we can see the frame of discernment at the fourth and fifth levels. If we eliminate all the descendant nodes below the level we consider, then the nodes at that level become terminal nodes. That level is called the terminal level. All terminal nodes at that level in the hierarchy may be considered to be the elements of the frame of discernment. They are mutually exclusive and exhaustive. A subtree which is rooted by a nonterminal node above that level may be considered to be a separate frame of discernment which discerns a group of spatial relationships denoted by the terminal nodes of that subtree. Considering the fifth level as the terminal level, for example, the node {two line intersection} at the third level is the nonterminal node. The subtree rooted by this node can be considered to be a separate frame of discernment which discerns different intersection relationships of two lines at the terminal level. All three corner types and four fork types and their rotations at the fifth level are exclusive and exhaustive. Only one of them will be true at any one time for a pair of visual events.

Comparing the tree (or its any subtree) in Figure 18 with the hypothesis tree described in Chapter IV, we find that they are the same in the following respects.

a. All the terminal nodes correspond to elements of a frame of discernment.

b. Higher nodes correspond to a class of their descendant nodes. Each of them corresponds to the union of the terminal nodes below it.

c. Only these subsets of the frame of the discernment which are of semantic interest are represented in the tree.

The only difference between these two trees is that the spatial relationships are used here instead of the visual events. With this image in mind, we can apply the algorithms presented in Sections 5.1-5.2 of this chapter to determine the spatial relationship between a pair of visual events.

There are many possible spatial relations between a pair of visual events. These relations include parallelism or intersection of two lines or two planes, and adjacency or surroundedness between two regions. Due to distortion, different view points or different projections, the appearance of a spatial relation between two visual events in an image may differ from that in the real world. Therefore, the nonquantitative

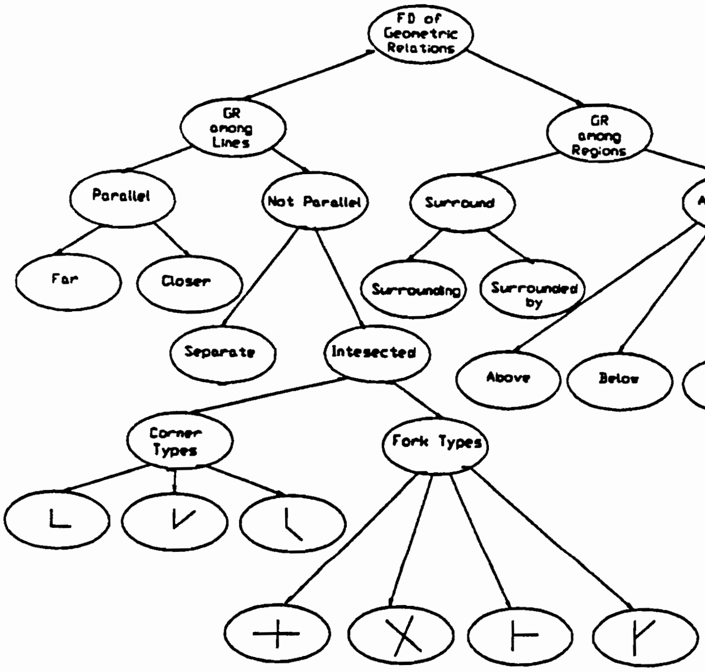


Figure 18. The frames of discernment for spatial relations between a pair of visual events.

definition of these spatial relations does not take near-misses into account and may cause ambiguity and uncertainty in such relations. For handling uncertainty in such spatial relations, it is necessary to have the quantitative definition and quantitative measures for these spatial relations. Quantitative definitions and measures of these spatial relations for computer vision can be found in (Rosenfeld and Klette, 1985; Mulgaonkar, 1984). The task in here is to map these measures into evidence strength for hypothesized spatial relations.

Suppose there are N hypothesis $h_1, h_2, ..., h_N$ representing N possible spatial relations. There are also M pieces of evidence $e_1, e_2, ..., e_M$ which support or contradict each of these hypotheses. Each e_j may be based upon spatial relation measurements $\omega_j \in \Omega_s$, which may come from different view points, from different projections, or from different parameter calculations. Each measurement provides evidence for the degree of spatial relations, such as the degree of adjacency or surroundedness and the degree of parallelism or intersection, for a pair of visual events.

A mapping function V maps the spatial relation measures into the strengths of the spatial evidence for and against a particular jth hypothesis in Θ_s. Let $f_{k,j}$ be the strength of e_k supporting the hypothesis h_j and $a_{k,j}$ be the strength of e_k against the hypothesis h_j. For a given ω_k, we have N $f_{k,j}$ and N $a_{k,j}$ for $j = 1, 2, ..., N$, given by Equation 5.1:

$$V_j : (\omega_k) \longrightarrow (f_{k,j}, a_{k,j})$$

For M pieces of spatial relationship measurements, the mapping results are

$$\begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{pmatrix} : \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_M \end{pmatrix} \longrightarrow \begin{pmatrix} (f_{1,1}, a_{1,1}) & (f_{1,2}, a_{1,2}) & \cdots & (f_{1,N}, a_{1,N}) \\ (f_{2,1}, a_{2,1}) & (f_{2,2}, a_{2,2}) & \cdots & (f_{2,N}, a_{2,N}) \\ \vdots & \vdots & \ddots & \vdots \\ (f_{M,1}, a_{M,1}) & (f_{M,2}, a_{M,2}) & \cdots & (f_{M,N}, a_{M,N}) \end{pmatrix}$$

and the uncommitted *bpa's* are

$$V_\theta : (\omega_k) \longrightarrow (f_{k,\theta} = 1 - \sum_{n=1}^{N} f_{k,n}, a_{k,\theta} = 1 - \sum_{n=1}^{N} a_{k,n}).$$

They satisfy the following relations:

$$\sum f_{k,x} = 1$$

$$\sum a_{k,x} = 1$$

To compute the impacts of all M spatial evidence measurements on each hypothesized spatial relation, we have to combine $f_{k,n}$ for $k = 1, 2, ..., M$ into f_n and $a_{k,n}$ into a_n. That is given by Equations 5.4.a-b:

$$f_n = m_1(h_n) = 1 - \prod_{k=1}^{M} (1 - f_{k,n})$$

$$a_n = m_2(\bar{h}_n) = 1 - \prod_{k=1}^{M} (1 - a_{k,n})$$

for each hypothesis $h_n \in \Theta_s$, where $n = 1, 2, ..., N$. That is

$$\begin{pmatrix} (f_{1,1}, a_{1,1}) & (f_{1,2}, a_{1,2}) & \cdots & (f_{1,N}, a_{1,N}) \\ (f_{2,1}, a_{2,1}) & (f_{2,2}, a_{2,2}) & \cdots & (f_{2,N}, a_{2,N}) \\ \vdots & \vdots & \ddots & \vdots \\ (f_{M,1}, a_{M,1}) & (f_{M,2}, a_{M,2}) & \cdots & (f_{M,N}, a_{M,N}) \end{pmatrix} \longrightarrow ((f_1, a_1) \quad (f_2, a_2) \quad \ldots \quad (f_N, a_N))$$

So far, we have considered the evidence for a hypothesis and against a hypothesis separately. Now, we need to combine these two bodies of evidence by Dempster's rule. The orthogonal sum of m_1 and m_2 is computed by Equations 5.7.a-d:

$$K^{-1} = 1 - m_1(h_n)m_2(\bar{h}_n)$$

$$m(h_n) = Km_1(h_n)(1 - m_2(\bar{h}_n))$$

$$m(\bar{h}_n) = Km_2(\bar{h}_n)(1 - m_1(h_n))$$

$$m(\theta_s) = K(1 - m_1(h_n))(1 - m_2(h_n))$$

where $m(h_n)$ is the belief supporting $h_{i,n}$, $m(\bar{h}_n)$ is the belief supporting $\bar{h}_n$, and $m(\theta_s)$ is the uncommitted belief. They are the initial dichotomy beliefs, which are exactly committed to each hypothesized spatial relation, based upon the spatial relationship measurements.

To consider the interaction of spatial evidence for all hypotheses in a frame of discernment of spatial relations, the algorithms provided by Section 5.2 can be applied to it. The computational complexity will be significantly reduced if the frame of discernment is constructed more precisely. For example, if a pair of visual events are two lines, then the branch of region relations can be completely ignored.

Furthermore, if the uncertainty is only imbedded in a subset of these spatial relations, then only the subtree corresponding to that subset is considered as a frame of discernment. The bottom-up and top-down processes can be carried out only within that subtree. Based upon the results of the computations, a particular spatial relation among others for a pair of visual events can be determined by its belief value, its plausibility value, or it belief interval. The first step is completed if the spatial relations for all pairs of visual events are determined.

Let $Q = \binom{S}{2}$ be the number of spatial relationships for $\binom{S}{2}$ pair of visual events, where S is the number of visual events asserted at the current level i. At this level, a set of interrelationships is denoted by $R_c = \{r_{c,1}, r_{c,2}, ..., r_{c,Q}\}$. Let $P = \binom{K}{2}$ be the number of interrelationships for $\binom{K}{2}$ pair of components of the predefined model at level $i+1$, where K is the number of components in that model. A set of interrelationships for this case is denoted by $R_t = \{r_{t,1}, r_{t,2}, ..., r_{t,P}\}$. Associated with each $r_{t,p}$, there is a pair of belief values $(f_{t,p}, a_{t,p})$ to indicate how strong that interrelationship supports or contracts this model. Suppose there are N hypotheses representing N possible models at level $i+1$. For the nth model, the set of interrelationships is $R_{t,n}$. These Q pairs of visual events and their interrelationships at level i provide evidence to support or to contradict these hypotheses at level $i+1$ as shown in Figure 19. The task of this step is to map these confirmatory visual events and their interrelationships into evidence strengths and to combine this body of evidence for hypotheses about visual events to be asserted at next higher level. In this step, the mapping process is a table-lookup process.

For given evidence $r_{c,k} \in R_c$, where $k = 1, 2, ..., Q$, the strengths of this evidence supporting and contradicting the hypothesis h_n, denoted by $f_{c,k,n}$ and $a_{c,k,n}$, are

$$f_{c,k,n} = \begin{cases} f_{t,j,n} & \text{if } r_{c,k} \in R_{t,n}; \\ 0 & \text{Otherwise} \end{cases}$$

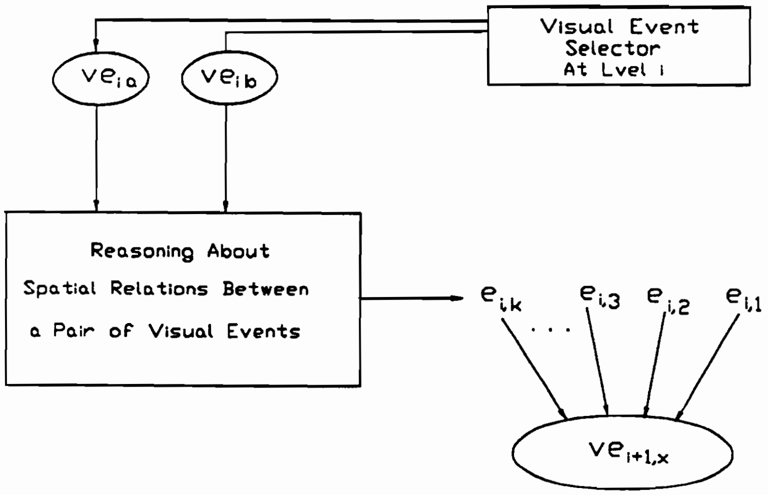


Figure 19. The spatial relationship handling.

$$a_{c,k,n} = \begin{cases} a_{t,j,n} & \text{if } r_{c,k} \in R_{t,n}; \\ 0 & \text{Otherwise} \end{cases} ,$$

where $n = 1, 2, ...N_i$, $1 \leq j \leq P$, and $f_{t,j,n}$ and $a_{t,j,n}$ are predefined evidence strengths associated with the interrelationships $r_{t,j,n} \in R_{t,n}$. $f_{t,j,n}$ and $a_{t,j,n}$ satisfy the relations:

$$\sum f_{t,j,x} = 1$$

$$\sum a_{t,j,x} = 1$$

with the uncommitted $bpa's$

$$f_{c,j,\theta} = 1 - \sum_{n=1}^{N} f_{c,j,n}$$

$$a_{c,j,\theta} = 1 - \sum_{n=1}^{N} a_{c,j,n}.$$

For Q interrelationships, the mapping results are

$$\begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{pmatrix} : \begin{pmatrix} r_{c,1} \\ r_{c,2} \\ \vdots \\ r_{c,Q} \end{pmatrix} \longrightarrow \begin{pmatrix} (f_{c,1,1}, a_{c,1,1}) & (f_{c,1,2}, a_{c,1,2}) & \cdots & (f_{c,1,N}, a_{c,1,N}) \\ (f_{c,2,1}, a_{c,2,1}) & (f_{c,2,2}, a_{c,2,2}) & \cdots & (f_{c,2,N}, a_{c,2,N}) \\ \vdots & \vdots & \ddots & \vdots \\ (f_{c,Q,1}, a_{c,Q,1}) & (f_{c,Q,2}, a_{c,Q,2}) & \cdots & (f_{c,Q,N}, a_{c,Q,N}) \end{pmatrix}$$

By using Equations 5.4.a-b, we combine Q $f_{c,j,n}$ into $f_{c,n}$ and combine Q $a_{c,j,n}$ into $a_{c,n}$ for each hypothesis. That is

$$\begin{pmatrix} (f_{c,1,1}, a_{c,1,1}) & \cdots & (f_{c,1,N}, a_{c,1,N}) \\ (f_{c,2,1}, a_{c,2,1}) & \cdots & (f_{c,2,N}, a_{c,2,N}) \\ \vdots & \ddots & \vdots \\ (f_{c,Q,1}, a_{c,Q,1}) & \cdots & (f_{c,Q,N}, a_{c,Q,N}) \end{pmatrix} \longrightarrow ((f_{c,1}, a_{c,1}) \quad (f_{c,2}, a_{c,2}) \quad \cdots \quad (f_{c,N}, a_{c,N}))$$

In the same way as before, the orthogonal sum of $f_{c,n}$ and $a_{c,n}$ can be computed by Equations 5.7.a-d. The result is the dichotomies $\{m(h_n), m(\bar{h}_n)\}$ and the uncommitted beliefs $m(\theta)$ for each hypothesis. These belief values can be attached to the corresponding nodes of the reasoning tree as described in Chapter IV. The interactions among these nodes then can be computed by the algorithms in Section 5.2 to resolve the conflicts and ambiguities. Finally, a decision can be made to assert one of the hypotheses as a valid visual event at the next higher level based upon belief values, plausibility values, or the belief intervals.

Chapter VI
SYSTEM IMPLEMENTATION

The uncertainty reasoning algorithms for hierarchical visual recognition at multiple levels of abstraction provide bidirectional active vision through both bottom-up and top-down processes. Generally speaking, vision is passive in the sense that the sensors do not have to radiate energy for measuring scene irradiances. However, vision is also active in the sense that eyes act less like cameras, but rather more like a set of goal-oriented detectors. Recently some researchers (Aloimonos, Weiss, and Bandyopuahyay, 1987; Bajcsy, 1985) have used the term *active vision* to denote a passive sensor employed in an active fashion, purposefully changing the sensor's state parameters according to sensing strategies. In this paper, however, *active vision* is viewed as a process of actively searching and reasoning about uncertain information, driven by organized knowledge and sensed data, in a hierarchical visual evidence space. It is a dynamic process of hypothesis generation, hypothesis selection, and hypothesis verification.

6.1 The Reasoning Paradigm

The paradigm to be used in this system is shown in Figure 20. It consists, essentially, of three steps:

1. Hypothesis generation and selection.

This is basically a bottom-up process in which primary evidence is used to form a set of alternative hypotheses at each level. Primary evidence is partial information

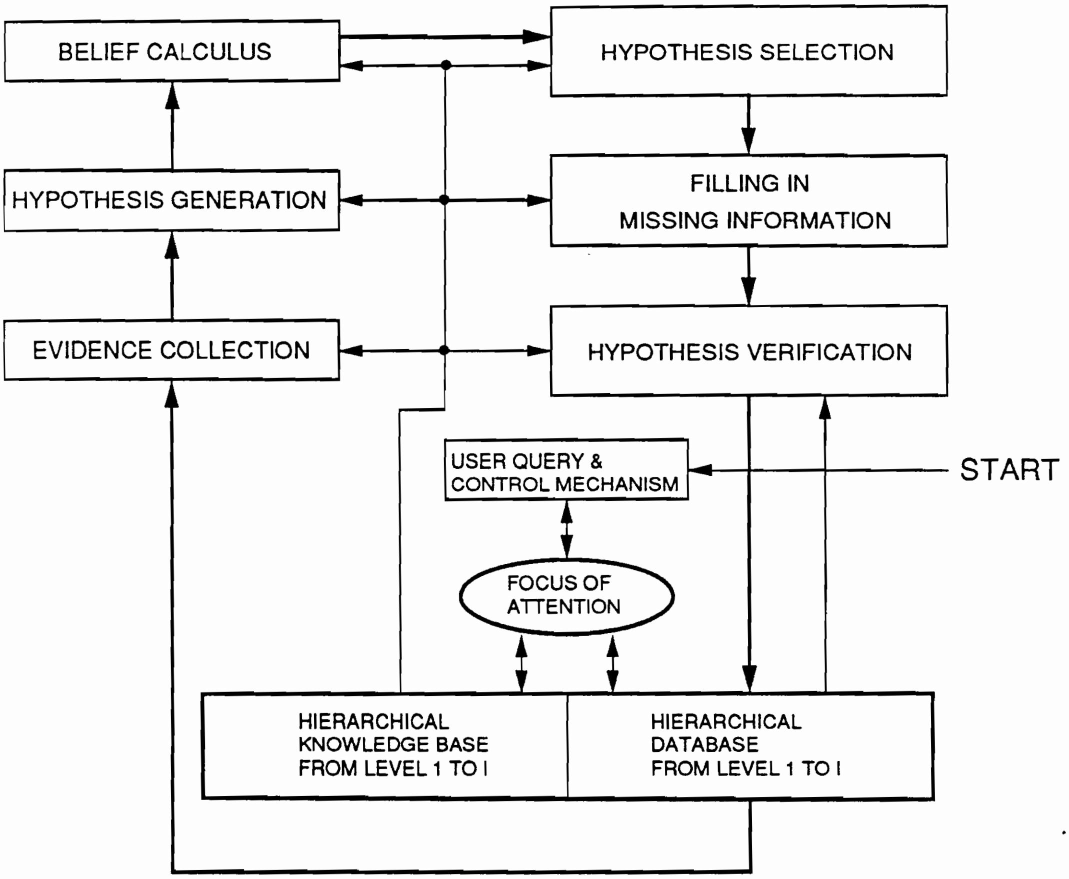

BELIEF CALCULUS
HYPOTHESIS SELECTION
HYPOTHESIS GENERATION
FILLING IN MISSING INFORMATION
EVIDENCE COLLECTION
HYPOTHESIS VERIFICATION
USER QUERY & CONTROL MECHANISM
START
FOCUS OF ATTENTION
HIERARCHICAL KNOWLEDGE BASE FROM LEVEL 1 TO I
HIERARCHICAL DATABASE FROM LEVEL 1 TO I

Figure 20. The reasoning system paradigm.

which is the minimum amount of information necessary for the hypothesis formation process. The hypothesis formation process can be applied at any level of abstraction in the visual event hierarchy. A belief function for a set of hypotheses at a particular level is calculated based upon available evidence from lower levels. A hypothesis is selected if and only if it has the maximum belief value that is above a specified threshold. This is analogous to the process of "take a glance" and "guess what it is" in human vision.

2. Filling in missing information

This is basically a top-down process. Associated with each hypothesis there is a detailed description of the hypothesized visual event. As soon as a hypothesis is selected, all properties and spatial relations about that visual event in the associated description are inherited by the hypothesis. This step is like that of connecting a more general concept to an incomplete instance in the human vision system. Since the information obtained from the lower level is usually partial and uncertain, information gaps (missing information from lower level processing) then can be filled by the associated detailed description. The process of "information-gap filling" approximately corresponds to the "it should have particular appearances" after an "educated guess" in human vision. Filling in a gap changes the descriptions associated with events at lower levels.

3. Hypothesis verification.

This is a process of "looking for more evidence" based upon the detailed description inherited by a hypothesis. It is actually a goal-oriented or higher-level knowledge-based identification of a lower-level event analogous to the process of looking for expected information in human vision. Evidence at one level causes selection of a hypothesis at a higher level, which due to its inherited description, initiates verification processes back at the lower levels. Sometimes, it may be necessary to reextract some local features from the original image under the direction of high-level interpretation. The correct interpretation of the visual events at a low-level can depend upon the high-level interpretation. This process provides an opportunity for correcting interpretations of the lower-level events. On the other hand, if low-level events can be interpreted correctly under the constraint of high-level interpretation, these interpretations will further verify the high-level interpretation. This is a check for consistency of interpretations made at the lower levels.

If a hypothesis can not be verified at this stage, then another hypothesis is selected, and the above processes is repeated. Since the current reasoning level is not necessarily the decision making level for the final interpretation, this process is still consistent with Marr's notion of "least commitment". During uncertainty reasoning, oscillations between hierarchy levels should be avoided when no hypotheses are verified in a given cycle. By setting appropriate threshold values for the corresponding belief functions associated with hypotheses, the degree of the system's sensitivity can be adjusted and the system stability can be maintained.

Several typical perceptual phenomena can be handled within the above paradigm. Among these phenomena are the following:

A) **Ambiguity Figures:** Sometimes, equal evidence is presented for two different hypothesis, such as the Necker cube (Figure 21) and the wife/mother-in-law. Although there is no evidence favouring either of two different hypotheses, the evidence mapping functions which are usually subjectively determined and which depend upon stored knowledge might change the perceptual bias in this ambiguous example. Remember that in Chapter 3, the belief values imbedded in a multi-viewpoint object model can be set unequally according to experience. As soon as one of the hypotheses is selected at step 1, the system will fill in the detailed information for that hypothesis at step 2. Then, one face (or cube) is 'seen' and this perception is enhanced by step 3, since this hypothesis can be verified at the lower level. The first perception is kept as the dominating perception until the system knowledge base changes the belief value (or change the evidence mapping function) such that another hypothesis is selected. Then step 1 to step 3 is repeated, and the perception is switched to another object.

B) **Paradox:** When people hold incompatible perceptual and conceptual hypotheses, they see a paradox. The most famous example of a paradox is the *impossible triangle* (Figure 22). Paradoxes can be caused by conflicting visual evidence or caused by hypotheses from false assumptions. This can be simulated within the above paradigm. When a hypothesis selected at the step 1 is incompatibly linked to a similar concept which holds the detailed description about that hypothesized visual event at step 2, the system 'sees' a paradox. A paradox may be detected at step 3 by utilizing high level knowledge. However, when the system goes back to step 1, the incompatible hypothesis may be selected again and the system may oscillate in this case. If an appropriate threshold is use to accept the reasoning result at step 3 and thereby maintain system stability, then the system outputs a paradox. This paradox would not be changed unless a critical piece of new evidence arises (for example, the change of view point).

C) **Perceptual fiction:** Perceptual fictions are often presented as illusory contours which are caused by surprising gaps in a picture (Figure 23). If we

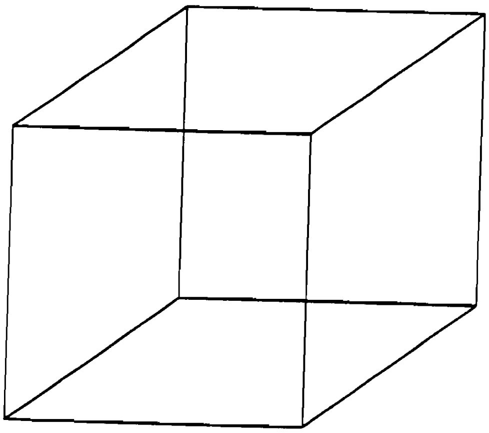
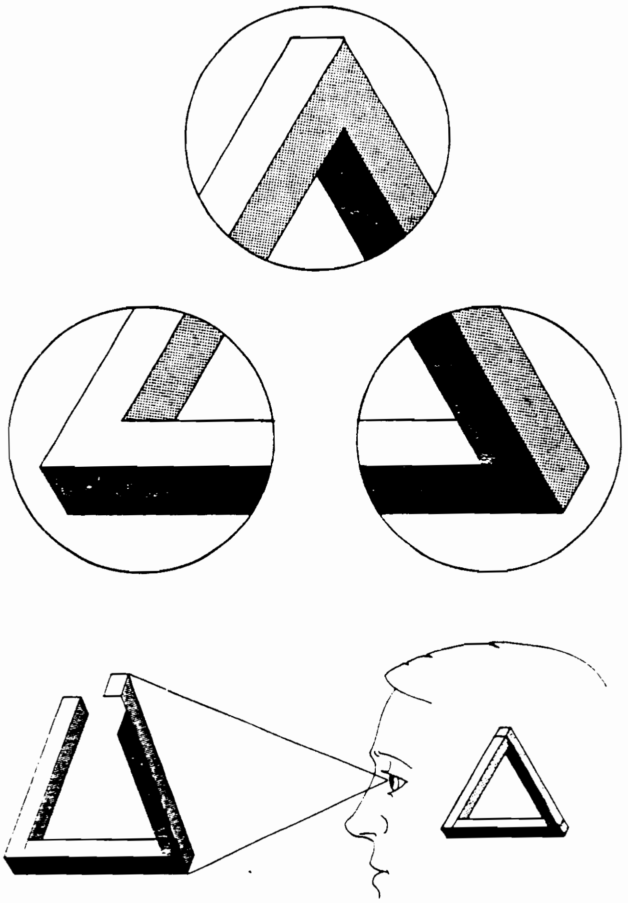


Figure 21. The Necker cube.



Observer sees the impossible triangle and not the real object as it 'should' be seen.

Figure 22. The impossible triangle (after Frisby, 1980).

accept the these gaps as visual evidence in step 1, then with the help of step 2 an illusion can be 'seen' by the system. An illusory contour might be verified at step 3 by examining large discontinuities in low level events under the guidance of high level knowledge.

6.2 The Database and the Knowledge Base

The database and the knowledge base in this reasoning system are all organized hierarchically. For each level of the database there is a corresponding level of the knowledge base. The collected visual evidence and the asserted visual events are stored in their appropriate database level. Visual event models, evidence extractors, evidence mapping functions, and threshold tables for hypotheses are stored in their appropriate knowledge base level. Uncertainty reasoning procedures are applied to the relevant level of the hierarchy through the focus of attention and control mechanisms.

Since the database is accessed so frequently, the performance of the reasoning system is critically dependent upon its data structures and its control mechanisms. Special efforts are required to make symbolic data structures work efficiently with multiband 2-D data arrays. The immediate difficulty is that of locating the spatially related visual events which need to participate in each deduction at each level of abstraction. The data structure must be so designed that the time spent making deductions about spatially unrelated evidence is minimized.

In the database, all data are treated as PROLOG fact rules asserted at their appropriate level of hierarchy. This database consists the following components:

1. an invertible index

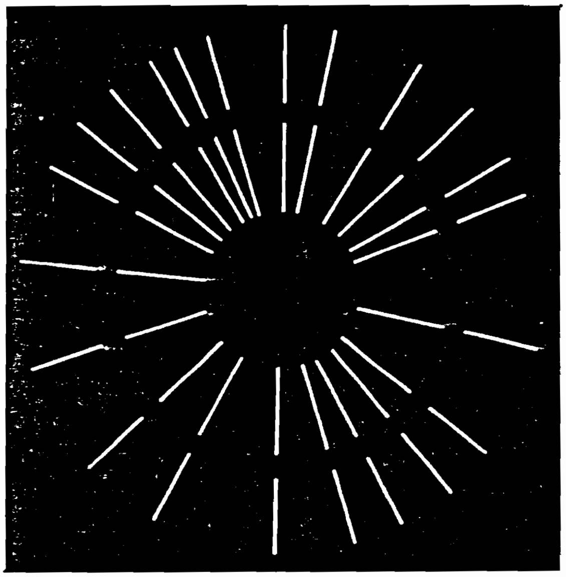


Figure 23. The surprising gaps and illusory contour.

2. the visual event data

3. the intermediate data generated by the reasoning process.

To access all relevant events efficiently, an inverted indexing method is used for the spatial information. A hierarchical partitioning method similar to quad trees or N trees is used to partition the coordinate space into blocks. Each level represents a different degree of zoom for the focus of attention mechanism. All visual events are indexed according to their locations in the data structure. For example, for linear visual events such as lines, edges, or linearly aligned features, an endpoint index is constructed for each level of the hierarchy that stores the names of the blocks that contain the endpoints. In addition to the endpoint index, a space spanning index is used to indicate which blocks are spanned by each event. This method of indexing makes it possible to locate spatially related segments quickly and reject unrelated events at an early stage.

The index table is a linear list of elements. Each element has the following form:

```
((LEVEL-ID VISUAL-EVENT-ID
  *VISUAL-EVENT-TYPE
  *INDEX-TYPE
  *SEARCH-FLAG
  *REASONING-STATUS
  *HORIZONTAL-INDEX
  *VERTICAL-INDEX
  *PRIOR-BELIEF
  *UPDATED-BELIEF
```

```
        *RELATED-EVENT-AT-HIGHER-LEVEL
        *RELATED-EVENT-AT-LOWER-LEVEL))
```

There are several flags in each element to indicate visual event types and reasoning status. For example, each open endpoint of a segment has one index element with *INDEX-TYPE=0. However, each indexing element with *INDEX-TYPE=1 corresponds to one block which the segment spanned. If a segment spans S blocks, then there will be S index elements for that segment. By using the PROLOG matching mechanism, it is easy to locate the indices of interesting events given both the visual event number and the level number. On the other hand, it is easy to locate all spatially related visual event numbers at a particular level by specifying vertical and horizontal index numbers, and the level number. It is efficient because the whole search and matching process involves only indices and not the visual event data (see Figure 24).

A typical data structure for visual events might include four parts: a unique visual event label, a visual event parameter list, a list of data samples at each linear segment end or on each region boundary, and an extended chain code. This database contains all the information required to reconstruct a raster image after reasoning has been completed. This permits the visualization of lower level events based upon the higher level interpretations.

Each level of the knowledge base consists of a set of evidence extractors, a set of evidence mapping functions, a threshold table for hypotheses, a set of visual event models, and a set of local inference procedures. The evidence extractor does the evidence measuring and evidence collection from lower levels. The evidence mapping functions map the collected evidence into strengths of evidence for the

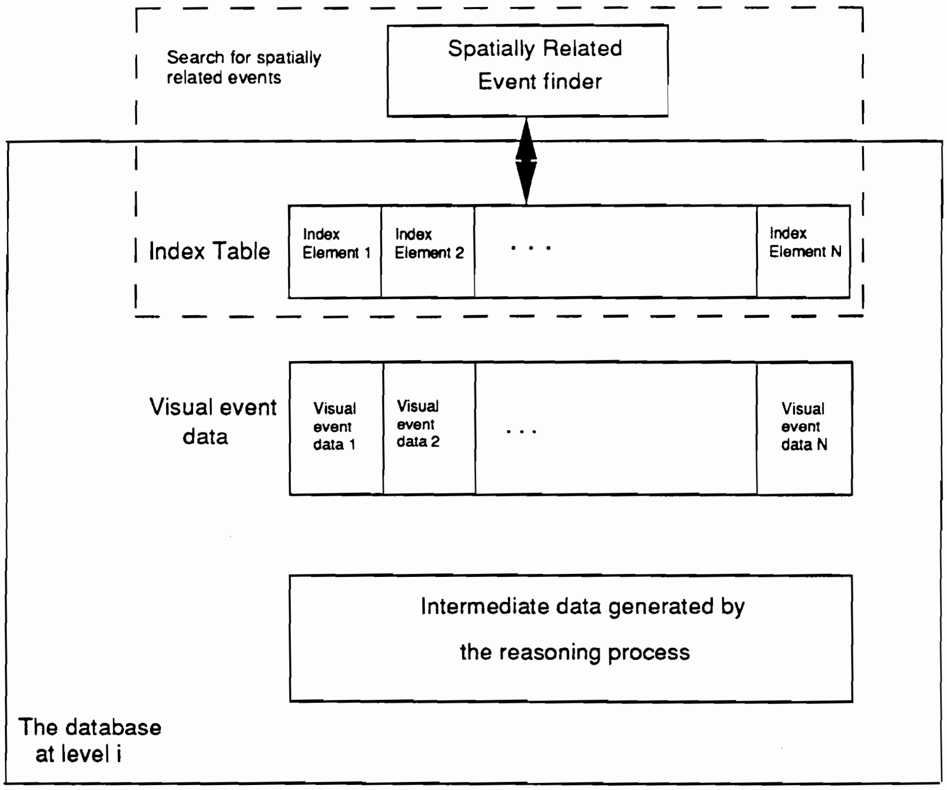


Figure 24. The database and the spatially related event search.

generated hypotheses. The threshold table is used for selecting hypotheses and for adjusting the stability of the reasoning system. The visual event model is a relational graph, with propagated belief values on branches, for describing that visual event. A group of mutually exclusive models forms a frame of discernment. These models are organized into the tree as described in Chapter IV. These trees or their subtrees are used to generate hypotheses and to calculate the interaction of hierarchical visual evidence (see Figure 25). Finally, the local inference procedures are used to implement the evidential reasoning using the D-S model.

6.3 The Focus of Attention

A focus of attention mechanism plays an important role in the uncertainty reasoning system. It is driven by the process of selecting spatially relevant visual events and their appropriate levels of abstraction. In other words, attention has to be focused both on geometric space and on visual evidence space when the reasoning is carried out. Given a visual event, the focus of attention mechanism returns a spatial search distance and a range of abstraction levels that describes the sphere of influence of that event. Let D^s be the spatial search distance and D^l be the level search distance. The spatial search distance provides an upper limit to the spatial neighborhood involved when the algorithm focuses attention on a particular visual event. This distance specifies how many neighboring blocks at the next lower level should be searched to generate hypotheses. The value of D^s is determined according to the size of that visual event. The value of D^l depends upon the level of abstraction i where the visual event is defined and relevant levels where visual evidence for the visual event is available. Suppose the size of jth visual event at level i is Z_{ij}. Then

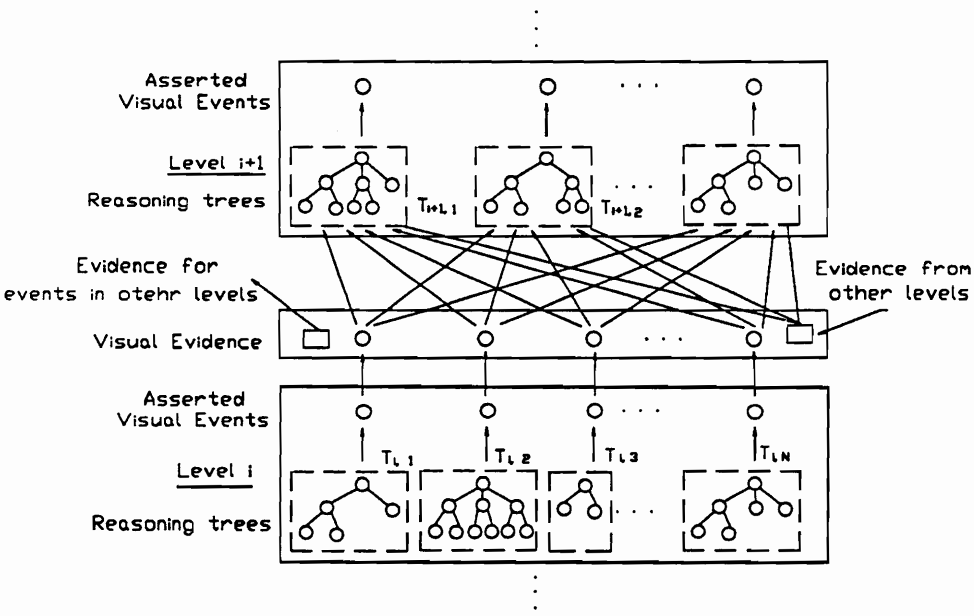


Figure 25. The hypothesis hierarchy in the visual event hierarchy.

$$D^s = CZ_{i,j}$$

where C is a constant, usually $1 \leq C \leq 2$.

The level search distance D^l usually spans two reasoning levels adjacent on either side of the current level. In the case that the evidence at next lower level is not available, however, the focus of attention should extend further down.

The focus of attention function $F(D^s, D^l)$ selects relevant visual events within the spatial search distance at the next lower level. As soon as $F(D^s, D^l)$ selects an element in the index table, the search flag in its index element is set to 1. This indicates that visual event has been examined and may be subsequently skipped in the top-down selection process. Thus, the visual event number, level number, visual event type, and spatial indices of the event in horizontal and vertical directions are obtained from the index element simultaneously. Then, a module called a finder calculates all possible search indices which fall within a search radius specified in the knowledge base. As a result, a search index list is generated which contains all N index pairs. Then, N pairs of features, which are spatially related to the selected visual event, can be accessed according to these N index pairs.

For all N pairs returned by the search, the hypothesis generator uses the search index list to generate hypotheses one by one. The algorithms for spatial relationship handling described in Section 5.3 of Chapter V can then be implemented. The data for each pair of visual events participating in the hypothesis are brought to the blackboard by decomposing the hierarchical data list from the database. The visual evidence extractor computes visual evidence measurements. By using evidence mapping functions, this evidence is mapped into evidence strengths which are propagated into the built-in inference network. For each hypothesis, a belief function which accumulates all the evidence can be computed following the inference network for the D-S model. Finally, N belief functions are calculated for N hypotheses generated according to the N pairs of indices. The decision maker or the hypothesis selector examines the result to find the maximum belief function value and compares it to a threshold specified in the knowledge base. A hypothesis is accepted if and only if its belief function is above the threshold and if its value is maximal among the alternatives. The other hypotheses are rejected. The selected hypothesis may invoke a hypothesis verification process. The focus of attention function can pick up visual evidence for this process at a chosen level by searching the inverted index table. Through the focus of attention function, the uncertainty reasoning can be carried out at any desired level and can go either top-down or bottom-up.

An example is given to illustrate the above process. Suppose that the task here is to recognize and to locate electronic parts on a printed circuit board (PCB) which contains an array of resistors. Suppose that we have already completed the uncertainty reasoning at the lowest levels, which determines the beliefs in significant lines and regions from the significant points and other visual events. Examples of reasoning at the line-level can be found in (Qian, Ehrich, and Campbell, 1990). Now we are at the level of reasoning about the primary components of objects based upon region-level evidence. The visual events about which reasoning takes place at this level are components of resistors. A frame of discernment for components of resistors is constructed, which corresponds to a single reasoning tree. There may be more reasoning trees at this level if there are other object types besides resistors to be recognized in the image.

Let i denote the current reasoning level. Given a visual event at level i, the focus of attention function returns the names of the levels to be analyzed: level i (the level of components of objects), level $i-1$ (the region level), and level 1 (the raw data level). The focus of attention function also returns the search distance: 1.5 times the maximum component size. Then, the focus of attention function selects an index element of a visual event from the invertible index table at the level $i-1$ (the region level). According to the spatial search distance, the finder searches for all index elements of the table which fall within the search radius. A search index list is generated which contains all N index pairs. Then, N pairs of regions, which are spatially related to the selected region, can be accessed by the hypothesis generator according to these N index pairs.

In this example, visual evidence at level $i-1$ is available for supporting visual events at the component level. Some evidence can be calculated from the level $i-1$. Some, such as the mean gray level and the variance gray level, should be reextracted from the original raw data under the guidance of the events at the region level. The visual evidence extractor computes these visual evidence measurements. Evidence mapping functions map these measurements into evidence strengths. Following the procedures in Section 5.1 of Chapter V that are stored as sequential procedures in the knowledge base, the initial beliefs which are exactly committed to each hypothesis are computed. These initial beliefs are propagated into the built-in reasoning tree. Then, the interactions among pieces of hierarchical evidence can be determined to deduce the type of component. For this step, the algorithm is described in Section 5.2 of Chapter V, and the implementation is given in the next subsection. This implementation is also stored as a part of the knowledge base. After the completion of the reasoning process at level i, all asserted components of resistors and the spatial relations among them are mapped into evidence strengths at level $i+1$

(object level) for deducing electronic parts at that level. The same reasoning steps as at level i may be applied at this level. As soon as a visual event is determined to be a resistor, the detailed description of that resistor will be propagated downward to level i by the focus of attention mechanism where it is used to check the consistency of related visual events at level i. Meanwhile, missing information which results from low-level processing alone is filled in. This completes the verification process for the hypothesis about a resistor at level $i+1$. The detailed experimental results for this example will be presented in Chapter VII.

6.4 The control of Evidence Interactions

To implement the reasoning algorithms of Sections 5.1-5.3 of Chapter V, we must generate a tree which denotes a hypothesis hierarchy as described in Chapter IV. The relationship between this hypothesis hierarchy and the visual event hierarchy is shown in Figure 25. For convenience, each tree is restructured as a binary tree as shown in Figure 26. Let the general tree be arranged such that the leftmost branch from the root to the leaf has the deepest levels. The binary tree is so formed that all siblings of a node of the general tree are connected and all links from a node of the general tree to its children, except for the link to its leftmost child, are deleted. Each node in this binary tree consists of groups of fields for the initial *bpa's* and the different orthogonal sums of beliefs required in Sections 5.2-5.3 of Chapter V. In order to generate such a tree automatically, an ordered list of hypotheses in that tree is required. The reasoning system reads in the ordered list and the initial dichotomous beliefs. A reasoning tree is then generated for the computation of evidence interactions. The procedures for the evidence interactions in the tree are described next.

First, the tree pointer is initialized to the root node. Then it moves down the tree along the leftmost path. A level counter is used to count the number of levels and addresses of the nodes along the leftmost path as they are pushed into a pointer stack. As soon as the bottom of the tree is reached, the bottom-up process 1 begins. The normalization factor and belief values for this process are computed from the current node and its siblings. Since the binary tree representation of the general tree is used, all the siblings are linked together. This is not a difficult process. Then the level counter decreases by one and the pointer stack pops the next higher level address. The process moves one level up. At this step, the first task is to check whether any sibling at this level has children nodes that have not completed the above process. The process should be completed for all nodes below the current level before the higher level computation begins. All the belief values obtained from the bottom-up process are stored in their corresponding nodes for further computations.

The above process is repeated all the way up to the daughters of the root node. Then the top-down process begins. When this process moves down one level, the pointer is reset to its leftmost node. A branch with maximum belief value is found by comparing the belief values of all siblings at that level. The computation is done only along that branch. This process repeats until the belief value of a desired node is computed. Again, because of the structure of the binary tree representation of the general tree, the computation at each level is carried only within a short linked list.

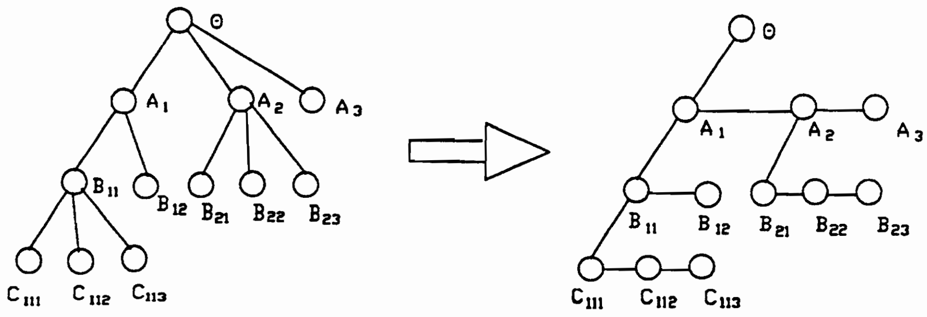


Figure 26. A binary tree representation of a general tree for the hypothesis hierarchy reasoning trees.

Chapter VII
APPLICATIONS

The research work for this dissertation consists of the theoretical development of a framework for an active vision system with uncertainty reasoning, coupled with experimental work on some constrained applications that demonstrate its merit. Some of this research work has been published or has been submitted for publication (Qian and Ehrich, 1989, 1990). Several results in support of these applications have already been obtained and are described in this chapter. These include both low-level and high-level processing.

At the preprocessing level, our uncertainty reasoning algorithm has been applied to a Multi-Threshold Adaptive Filtering (MTAF) application for preserving texture while removing noise.

At the low level, an edge labeling procedure has been developed which generates single pixel width edges without thinning. Methods have been designed for extracting spatial relationships from regions and lines, and a method for converting 2-D data into a symbolic database have been completed. Then the uncertainty reasoning algorithm may be extended to the shape completion problem.

At the high level, an uncertainty reasoning system using D-S theory for automatically extract the drainage networks has been constructed along with the control mechanism and data structure. Also, this framework has been applied to the recognition of electronic parts on a printed circuit board. The experimental results shown promises on this framework.

7.1 Texture Preservation and Noise Removal

In computer vision, sometimes we need to discriminate among objects whose differences appear only as subtle differences in tone and weak texture. A typical example of such a task is the discrimination of water regions from shadow regions on SAR imagery (Qian and Haralick, 1990). Because both water and shadow regions in a radar image are not rich in texture, the noise will tend to mask whatever texture there is and make the recognition task more difficult. Preprocessing is needed to remove the noise while preserving the weak texture and other subtle details.

There is, of course, a compromise between noise removal and texture preservation. Usually, a filter which has a powerful noise-cleaning capability may remove or spatially distort edge, line, and texture information. In contrast, a filter which preserves subtle detail will tend to have low noise-cleaning capability. Generally speaking, a real world image consists of many regions in which local activity varies from region to region. It is difficult to be satisfied with the image using only one simple filter. Based on this consideration, a way in which several simple filters can be combined to form a more efficient and more flexible context dependent filter is desired so that the advantages of simple filters can be preserved, their drawbacks avoided and, at the same time, an optimal effect can be obtained.

A Multiple Threshold Adaptive Filtering (MTAF) has been proposed to achieve this goal. It uses a generalized gradient function and a local variance function, which measures the local contextual information, as evidence to determine the nature of the filtering for each local neighborhood.

First, the generalized gradient function for every pixel is computed.

Let Z_{ij} be the pixel gray tone value to be filtered and let G_{ij} be the corresponding local generalized gradient function within a 7×7 window. In order to avoid the effects of random noise in computing G_{ij}, we use the local gray tone submean M_{kd} instead of an individual pixel value. If it is assumed that the size of any random noise cluster is three or fewer pixels, then the above process is sufficient for avoiding the effects of random noise in computing $G_{i,j}$.

Let (s, r) be a set of pixels neighboring (i, j) having gray tone intensity Z_{sr}. The gray tone submeans are computed by the overlapped 3×3 subwindows. That is

$$M_{k,l} = \sum_{s=q}^{f} \sum_{r=p}^{g} Z_{sr} \tag{7.1}$$

where

$$q = 1 + 2(k - 1)$$

$$f = q + 2$$

$$p = 1 + 2(l - 1)$$

$$g = p + 2$$

and k,l = 1, 2, 3.

Then for the four directions, the local gradient G_s's are

$$G_s^0 = |(M_{11} + M_{12} + M_{13}) - (M_{31} + M_{32} + M_{33})| \tag{7.2.a}$$

$$G_s^{\frac{\pi}{4}} = |(M_{11} + M_{12} + M_{21}) - (M_{32} + M_{33} + M_{23})| \tag{7.2.b}$$

$$G_S^{\frac{\pi}{2}} = |(M_{11} + M_{21} + M_{31}) - (M_{13} + M_{23} + M_{33})| \tag{7.2.c}$$

$$G_S^{\frac{3\pi}{4}} = |(M_{12} + M_{13} + M_{23}) - (M_{31} + M_{32} + M_{21})| \tag{7.2.d}$$

and the final generalized gradient is proportional to the sum of the directional gradients

$$G_{ij} = C \sum_{n=0}^{3} G_s^{\frac{n\pi}{4}} \tag{7.3}$$

where C is the scaling factor.

The resulting G_{ij} is a good measure of local texture contrast.

(2) Second, the local variance is computed over a $N \times N$ moving window.

Let $\bar{Z}_{ij}$ be the local mean, V_{ij} be the local variance and $p = (N-1)/2$. Then, we have

$$\bar{Z}_{ij} = \frac{1}{N^2} \sum_{s=i-p}^{i+p} \sum_{r=j-p}^{j+p} Z_{sr} \tag{7.4.a}$$

and

$$V_{ij} = \frac{1}{N^2} \sum_{s=i-p}^{i+p} \sum_{r=j-p}^{j+p} (Z_{sr} - \bar{Z}_{ij})^2 \qquad (7.4.b)$$

It is well known that the local variance near an edge is greater than the variance in either side of the edge. It is also a good measure characterizing the edge information.

More other functions may be chosen as evidence of local activities and different determining methods may be applied for different filtering tasks. A more detailed description on the MTAF can be found in (Qian, Yu, Haralick, 1986; Qian, 1985). Here we describe only briefly this filtering algorithm with a knowledge-based presegmentation procedure using Dempster-Shafer evidence theory.

The algorithm is the following:

A frame of discernment Θ for presegmenting the noisy image consists of the following exhaustive and exclusive subsets:

1. Regions which are homogeneous. They will be filtered by a moving average filter.

2. Regions which have relative weak edges or intermediate ramp edges. They will be filtered by the sigma filter (Lee, 1983).

3. Regions which have sharp edges that are parallel to the direction of the moving window. They will be filtered by the eight edge direction weighted filter (Lee, 1981).

4. Regions which have sharp edges that are not parallel to the window motion direction. They will be filtered by a median filter (Yang and Huang, 1980).

Each subset of Θ corresponds to a hypothesis. The frame of discernment, Θ, delimits a sample space which contains all possible segmentation situations, only one of which is true at any one time. Let the set of hypotheses be $\{H_a, H_s, H_e, H_m\}$, respectively. Then, the frame of discernment consists of $\{H_a, H_s, H_e, H_m\}$ which corresponds to the above four situations, respectively.

The evidence provided by each measurement ω_k is mapped to basic probability assignments (*bpa*) over the hypotheses discerned by a frame Θ. The *bpa* represents the impact of each distinct piece of evidence on the subsets of a frame Θ. According to the values of the measurements, a set of thresholds can be used to extract the local evidence. Let $\{e_{A,k}, e_{S,k}, e_{E,k}, e_{M,k}\}$ be a set of local evidence representing "no edges," "ramp edges," "sharp edges," etc., respectively. Let $T_{k,s}$ be the set of thresholds for the kth measurement, $k = 1, 2, ..N$ the number of measurements, $s = 1, 2, ..S$ the number of thresholds. Only one of them is true at location (ij). For example, for the generalized gradient function, ω_1, we have

$$\begin{cases} e_{A1} = true & \text{if } \omega_1(ij) < T_{1,1}; \\ e_{S1} = true & \text{if } T_{1,1} \leq \omega_1(ij) \leq T_{1,2}; \\ e_{E1} = true & \text{if } \omega_1(ij) > T_{1,3}) \text{ AND } (\omega_1(ij) = GD); \\ e_{M1} = true & \text{if } (\omega_1(ij) > T_{1,3}) \text{ AND } (\omega_1(ij) \neq GD). \end{cases} \tag{7.5}$$

where *GD* is the generalized gradient value in the direction of the window motion (Qian, Yu, and Haralick, 1986). These thresholds give the processor a flexible choice for different image types and different processing purposes. Each piece of extracted evidence will be mapped into a *bpa* over the set of hypotheses. Applying the Dempster's rule rule, the procedure pools the multiple bodies of evidence obtained from the different measurements to get new belief functions. Finally, a single hypothesis among others is verified, if and only if the belief function has a maximum value. An presegmentation image is shown in Figure 27.

In our experiments, the filtered pixel value W_{ij} is given by

$$W_{ij} = \begin{cases} A_{ij} & \text{if } H_a \text{ is verified;} \\ S_{ij} & \text{if } H_s \text{ is verified;} \\ E_{ij} & \text{if } H_e \text{ is verified;} \\ M_{ij} & \text{if } H_m \text{ is verified.} \end{cases} \tag{7.6}$$

where W_{ij} is the filtered pixel value, A_{ij} is the output of the 5×5 averaging filter, S_{ij} is the output of the 7×7 sigma filter, E_{ij} is the output of the eight edge direction weighted filter, and M_{ij} is the output of the 3×3 median filter.

As a result, the averaging filter is applied only to the very homogeneous regions, which avoids blurring the weak texture. The median filter is applied only to those regions which have sharp edges, and these sharp edges are not parallel to the median filter moving direction, which avoids eliminating the lines and small objects and avoids creating artifacts in other regions. The sigma filter is applied only to these regions where preserving the weak texture is more important than just noise smoothing. Thus, a balanced texture preserving and noise removal effect can be simultaneously achieved. To see how the MTAF filter improves the low-level labeling result more obviously, we added Gaussian normal noise $N(0, \sigma^2)$ with different σ values to the SAR images. Then the same statistical labeling procedure was applied to the noisy one, the filtered one, and the original one, respectively. The relative labeling accuracy is measured by comparing them with that of original image labeling. Our experiments shown that the relative accuracy can be improved from 61.9 percent to 92.2 percent (with noise $\sigma = 10$), even form 65.5 percent to 96.5 percent (with noise σ=5).

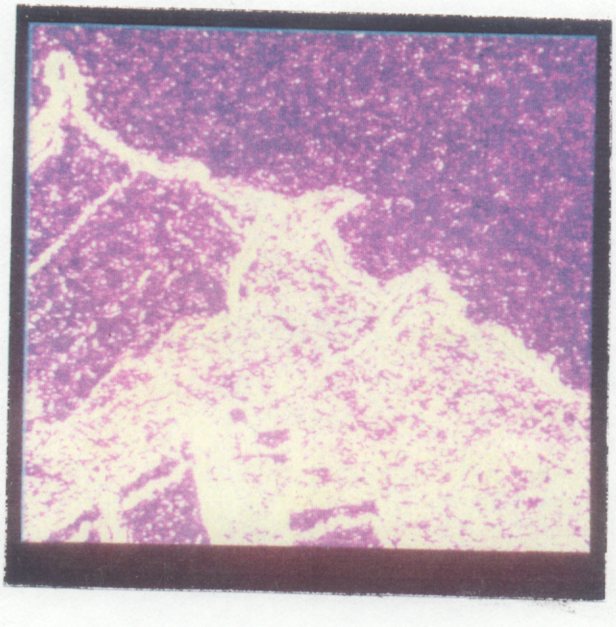


Figure 27. The presegmentation image.

7.2 One-pixel Width Edge Detection Without Thinning

Applications of the uncertainty reasoning system at the lowest processing level have two specific characteristics. One is that frames of discernment are relatively simple and the number of levels of hypotheses is relatively small. The other is that hypotheses about a low level visual event may be combined with the event locations. That is, the reasoning process not only has to determine if there is a event, but also has to determine where the event is. Some applications are given in the following sections to show how this reasoning system can be implemented in processes from pixel to edge and from edge to contour.

Edge detection plays a crucial part in any vision system. It is also the first necessary step in the shape completion. As described before, line-drawing-like features will be the primary features used in the proposed vision system. Therefore, the extraction of one-pixel width line features is an important part of the vision system. The following algorithm provides a way to extract line features from a graytone image.

In many edge detection schemes, the image $I(x,y)$ is first filtered and then a second differential operator D^2 is applied to the filtered image $\hat{I}(x,y)$. The main goal of filtering and differentiation is to produce a representation of second derivative zero crossings and extrema (Marr and Hildreth, 1980). Edges are then identified in correspondence to the zero-crossings of $D^2\hat{I}(x,y)$ which provides a compact representation of intensity changes. The geometrical properties of the locus of zero-crossings is defined by

$$D^2\hat{I}(x,y) = 0 \tag{7.7}$$

where $\hat{I}(x,y)$ is the filtered image and D^2 can be either an RID (Rotational Invariant Derivative) or a DD (Directional Derivative).

One of the important properties of zero-crossings which are produced by the RID is that zero-crossing contours are closed curves or curves that terminate at the boundary of the image. In contrast, both the gradient and template edge operators generally produce disconnected and irregular edges which require some type of postprocessing such as relaxation, thinning, or line fitting to connect edge segments into closed regions.

Since an edge represents an intensity change between two tonal regions, ideally the edge should be located between two pixels. In order to locate the edge as accurately as possible, the edge width should not exceed one pixel, the labeled zero-crossing contours should preserve the size and shape of the regions of opposite sign which they bound, and the labeled zero-crossing contours should prevent anomalies, such as the creation of extraneous edges along oblique edge segments. There are two problems:

1. How to label the zero-crossing contour such that it locates the edge more precisely.

2. How to make adequate use of the information provided by $D^2\hat{I}(x,y)$.

Haralick (1981) uses 8-connected neighboring pixels to check the zero-crossings for every pixel. If $D^2\hat{I}(x_0,y_0) < -\alpha$ and one of its eight neighbors

$$D^2\hat{I}(x_i,y_j) > \alpha \tag{7.8}$$

then the central pixel is labeled as an edge pixel, where α is a preset threshold.

This technique turns out to have two disadvantages. One is that it handles zeros of $D^2\hat{I}$ improperly and results in some holes along the edge (Figure 28). The other is that the labeled edge is quite thick, which requires that a thinning algorithm be used to reduce the edge to one pixel.

Huertas and Medioni (1985) use a predicate-based algorithm for locating zero-crossings. 11 predicates and almost 50 masks are required by this algorithm. Also there is no evidence that this method will guarantee that the edge width is not wider than one pixel. They still use Nevatia's algorithm (Nevatia and Babu, 1980), which uses thinning first and a linking and tracking post-processing stage.

Instead of above algorithms, an algorithm has been developed which is able to locate zero-crossing contours to the nearest pixel with a more precise position. This algorithm is based upon the fusion of multiple evidence sources through D-S model. This is a low level application of the uncertainty reasoning. Here, the visual events are edges at different locations. The visual evidence is some intensity discontinuity measurements from raw data. These measurements include (but not limited to)

(1) local intensity $max(I(x,y))$ or $min(I(x,y))$ which corresponds to the possible dark lines or light lines;

(2) local gradient values $max|\nabla I(x,y)|^2$ which corresponds to the possible edge pixels;

(3) zero crossing of $D^2\hat{I}(x,y)$;

(4) the neighboring edge pixels.

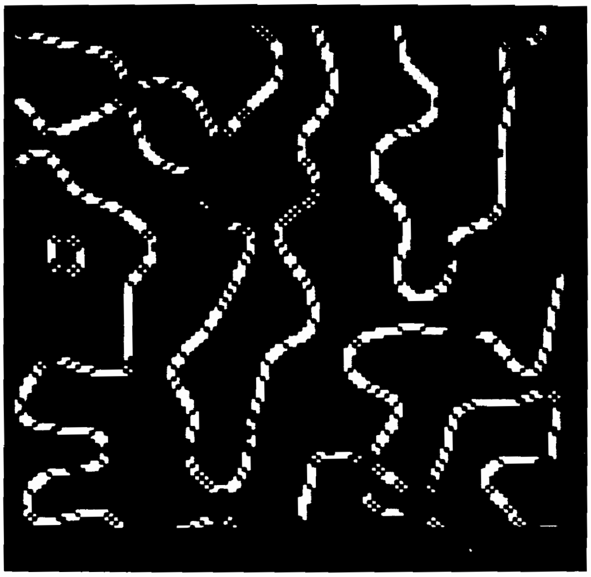


Figure 28. The edge image with holes within edges.

For each location (x, y), the frame of discernment Θ is simple. That is, there is an edge pixel or a no-edge pixel. However, if we move an $N \times N$ window along a possible edge zone, then Θ will consist of N^2 hypothesized edge locations within the window plus the no-edge-in-window hypothesis. Each measurement is mapped into a belief value which provides its support to this frame of discernment. These belief values are combined by the procedures described in Chapter III Section 3.3. An $N \times N$ window moves along the hypothesized edge such that the hypothesized edge pixel is always in the center of the window. A distance D_E between the initially hypothesized edge pixel and the possible edge location suggested by a piece of evidence is computed within that window. D_E is mapped into belief values for the hypothesis. These pieces of evidence will confirm or disconfirm this hypothesis as their belief values are combined. Since zero crossings convey much more edge information, they can be used as an initial hypothesis about edge pixels. This will simplify the procedure further. Here, we only described our new method for the zero crossing measurement.

Two narrow masks which are 90 degree rotations each other (Figure 29) are used. The central pixel is marked as possible edge pixel if any of the following conditions is met.

(1) $D^2\hat{I}(x_0, y_0) = 0$ and any one of the following sign tests is true.

(a) $D^2\hat{I}(x_0, y_{-1})$ and $D^2\hat{I}(x_0, y_1)$ are of opposite sign.

(b) $D^2\hat{I}(x_{-1}, y_0)$ and $D^2\hat{I}(x_1, y_0)$ are of opposite sign.

(2) $D^2\hat{I}(x_0, y_0)$ and any one of its four-neighbors are of opposite sign, and the following ratio test is true.

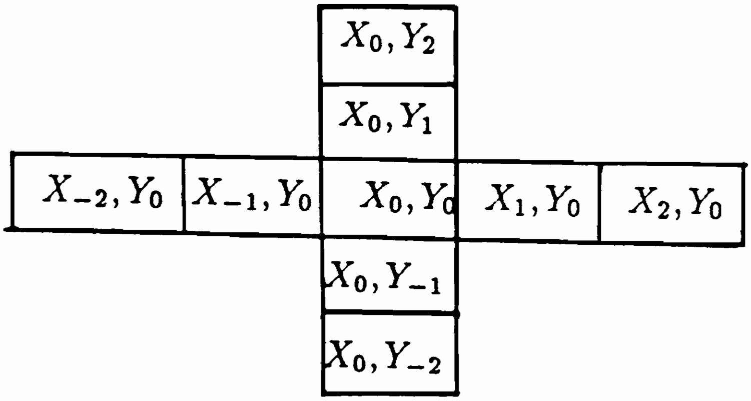


Figure 29. The masks for locating zero-crossings.

(a) If the neighbor is $D^2\hat{I}(x_0, y_1)$, then the ratio test is

$$\frac{|D^2\hat{I}(x_0,y_0)|}{|D^2\hat{I}(x_0,y_1)|+|D^2\hat{I}(x_0,y_{-1})|} \le \frac{|D^2\hat{I}(x_0,y_1)|}{|D^2\hat{I}(x_0,y_2)|+|D^2\hat{I}(x_0,y_0)|} \tag{7.9}$$

(b) If the neighbor is $D^2\hat{I}(x_0, y_{-1})$, then the ratio test is

$$\frac{|D^2\hat{I}(x_0,y_0)|}{|D^2\hat{I}(x_0,y_1)|+|D^2\hat{I}(x_0,y_{-1})|} \le \frac{|D^2\hat{I}(x_0,y_{-1})|}{|D^2\hat{I}(x_0,y_{-2})|+|D^2\hat{I}(x_0,y_0)|} \tag{7.10}$$

(c) If the neighbor is $D^2\hat{I}(x_1, y_0)$, then the ratio test is

$$\frac{|D^2\hat{I}(x_0,y_0)|}{|D^2\hat{I}(x_1,y_0)|+|D^2\hat{I}(x_{-1},y_0)|} \le \frac{|D^2\hat{I}(x_1,y_0)|}{|D^2\hat{I}(x_2,y_0)|+|D^2\hat{I}(x_0,y_0)|} \tag{7.11}$$

(d) If the neighbor is $D^2\hat{I}(x_{-1}, y_0)$, then the ratio test is

$$\frac{|D^2\hat{I}(x_0,y_0)|}{|D^2\hat{I}(x_1,y_0)|+|D^2\hat{I}(x_{-1},y_0)|} \le \frac{|D^2\hat{I}(x_{-1},y_0)|}{|D^2\hat{I}(x_{-2},y_0)|+|D^2\hat{I}(x_0,y_0)|} \tag{7.12}$$

This algorithm only produces one pixel-width edges from the $D^2\hat{I}(x,y)$. If there are three adjacent pixels, the middle of which is zero-valued and the other two are of opposite sign in either horizontal direction or vertical direction, then the zero-valued pixel is labeled as an possible edge pixel. If there are two adjacent pixels which are of opposite sign in either horizontal direction or vertical direction, only one of these two pixels are chosen as an edge pixel by the ratio tests. The ratio tests compare the relative magnitudes of $D^2\hat{I}(x,y)$ and provide information for deciding which pixel is the nearest one to the zero-crossing. In this process, only the horizontal direction $\vec{X}$ and the vertical direction $\vec{Y}$ are considered. Since in R^2 space the derivative in any arbitrary direction can be expressed in terms of $\frac{\partial}{\partial X}$ and $\frac{\partial}{\partial Y}$, rotationally invariant differential operators or a large set of directional derivatives requires only the use of two narrow directional derivatives. Accordingly, the zero-crossing can be located by using two narrow masks in the $\vec{X}$ and $\vec{Y}$ directions, respectively. Edges have been successfully labeled by this algorithm for several type of images (Figure 30 – 36). This algorithm produces a line-drawing like feature image for further processing.

7.3 The Spatial Evidence Extraction from Intermediate Level Visual Event

7.3.1 Extracting Structural Information from Line-drawing-like Visual Events

Based upon the extracted line-drawing-like visual events, an algorithm which extracts spatial relationships from these events and then converts them into a symbolic database has been developed. The output of the procedure in previous section is taken as the input of this procedure. The following algorithm is specific to the case where line segments link to form trees. It can be generalized to more general line-drawing-like data in which all lines have a single pixel width.

A special tracing algorithm is used to measure the set of spatial properties of each segment generated by the edge detector. Each segment has two endpoints; an endpoint is called open if it is not shared by other segments, and it is called a fork point if it is. Fork points are points where segments merge to form trees. For implementation efficiency, a doubly linked chain of pixels is generated for each segment according to the twenty predefined linking patterns shown in Figure 37.

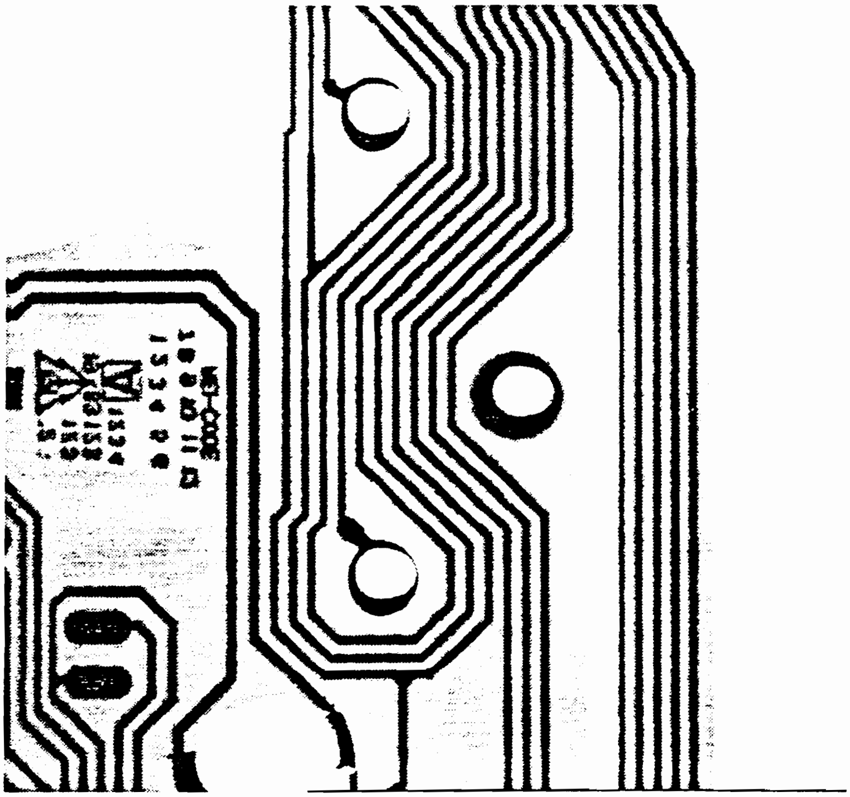


Figure 30. The test image 1: the printed circuit board. Size: 512 × 512.

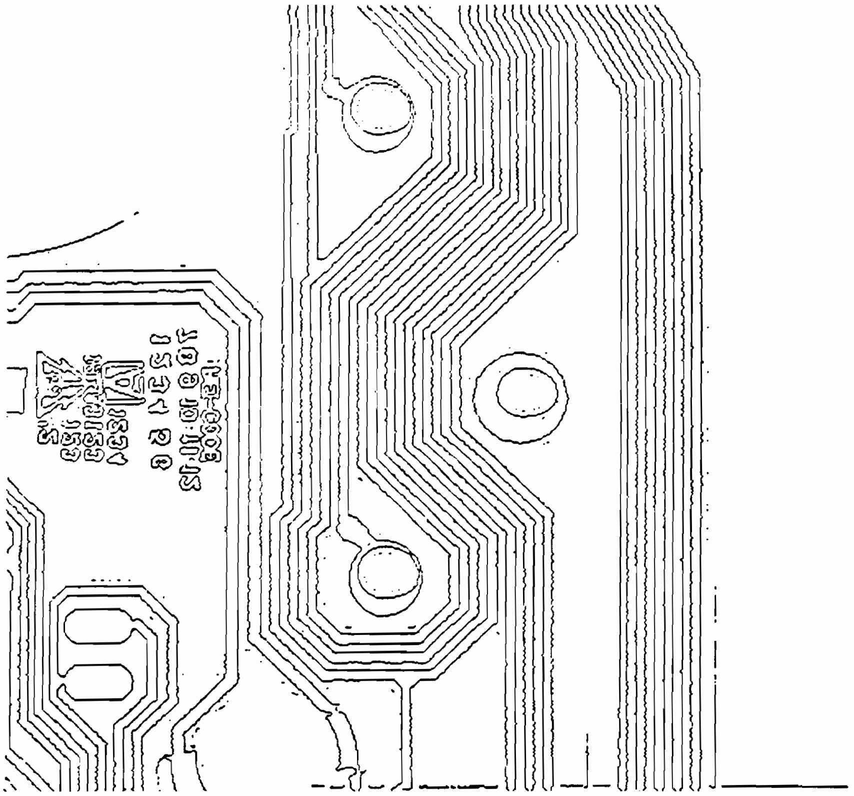


Figure 31. The edge image labeled by the algorithm for Figure 30.

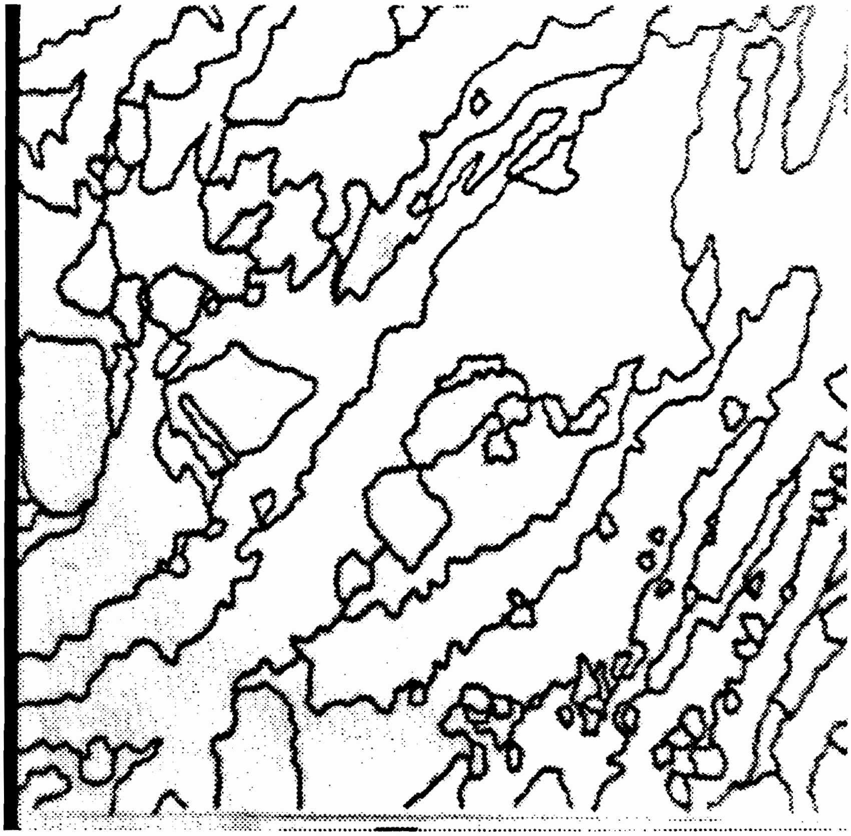


Figure 32. The test image 2: the hand-drawing-line picture. Size: 512 × 512.

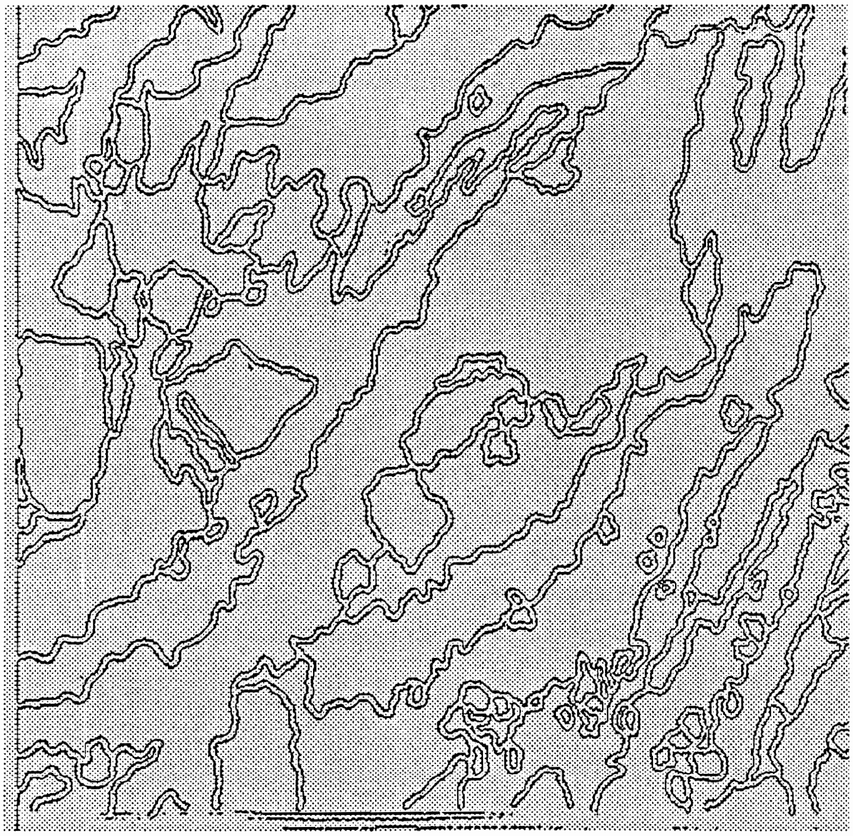


Figure 33. The edge image labeled by the algorithm for Figure 32.

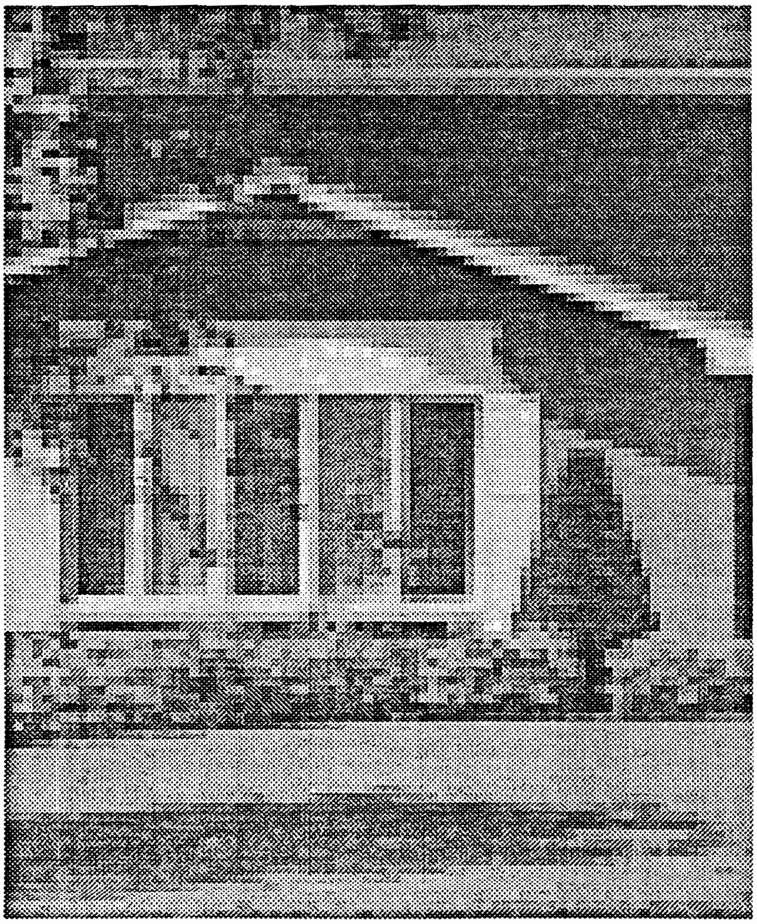


Figure 34. The test image 3: the house image in an outdoor scene.

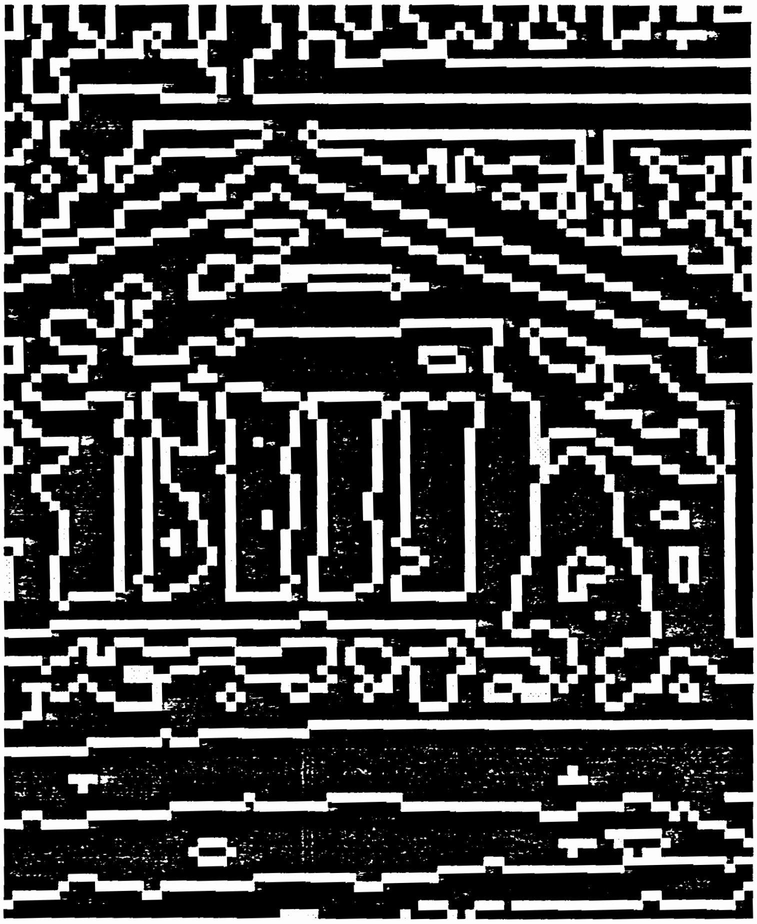


Figure 35. The result 1 of Figure 34.

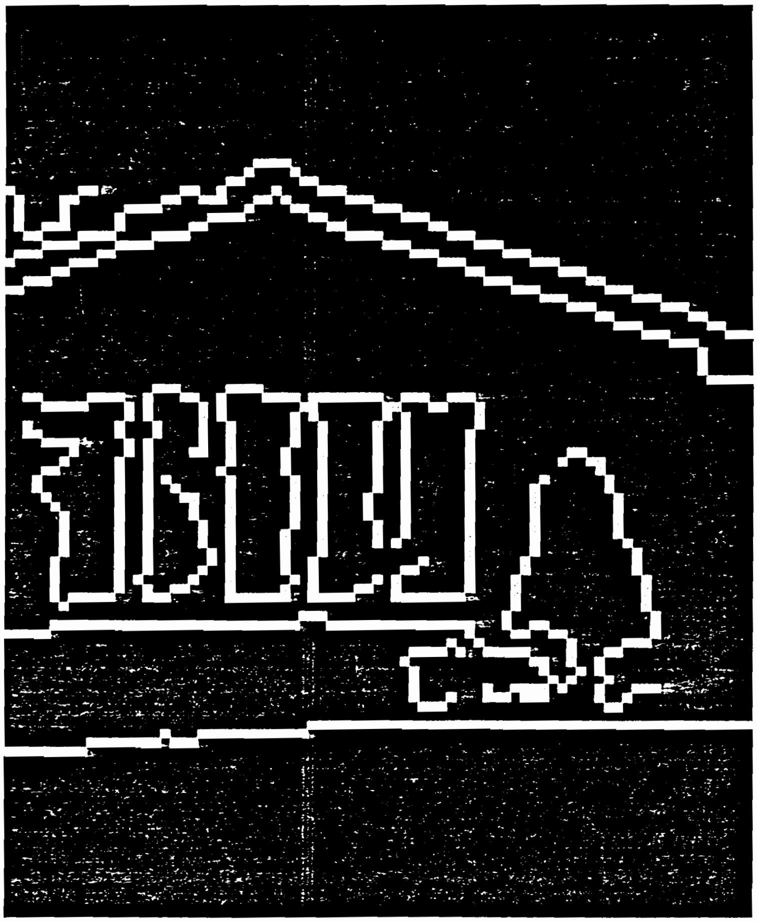


Figure 36. The result 2 of Figure 34.

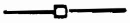
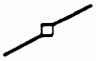
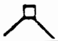
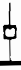
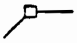
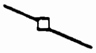
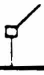
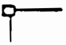
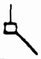
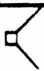
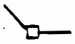
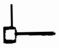
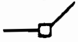
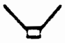
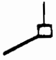
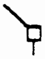
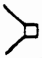
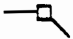
type code	pattern	type code	pattern
1		11	
2		12	
3		13	
4		14	
5		15	
6		16	
7		17	
8		18	
9		19	
10		20	

Figure 37. The link patterns.

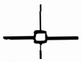
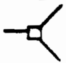
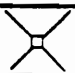
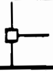
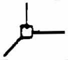
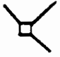
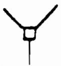
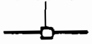
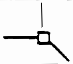
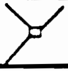
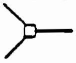
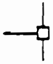
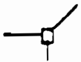
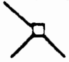
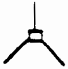
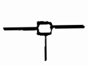
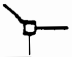
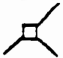
type code	pattern	type code	pattern
1		10	
2		11	
3		12	
4		13	
5		14	
6		15	
7		16	
8		17	
9		18	

Figure 38. The fork patterns.

In the same pass, all fork points in trees are detected (see Figure 38), and all open endpoints of tree branches are marked. These three tasks are done in one pass. A bidirectional linked chain is a sequence of pixels along a segment in which each pixel between its endpoints has pointers to its two neighbors. Bidirectional linked chains facilitate the network tracing process since the tracing sequence for the pixels in different segments may be different. The bidirectional linked chains, the fork points, and the open endpoints provide a clear guide for tracing and measuring all spatial information for segments simultaneously.

The network tracing process illustrated in Figure 39 is the following:

1. Scan the image from top to bottom, line by line.

2. If an open endpoint is encountered, the scanning process is interrupted, and a tracing and measuring process is initiated to follow the bidirectional link codes.

3. Record the segment length, positions, elevations, and tangent directions for all segment pixels and compute the elevation slopes, mean curvatures, end directions, and mean elevations for both ends of each segment. Assign the segment label and tree label for the traced segment. This process terminates when a fork point or an open endpoint is reached. All pixels which have been traced are set to zero.

4. If a fork point is found, push the fork position onto a fork stack. Search the next untracked branch leaving the fork point in clockwise order.

5. If an open endpoint is encountered, then check the fork stack. If it is not empty, pop the stack and start tracing a new segment from the popped position by

repeating Step 3 to Step 5. If the fork stack is empty, then the whole tree in the network has been completely traced. Continue scanning with Step 1.

After the whole image has been scanned, the output file contains all the spatial information for each segment. All segments have their own unique labels. The output of this procedure is a property file which can be converted to a database written in PROLOG.

7.3.2 The Method for Extracting Spatial Regional Relationships

Uncertainty reasoning in a hierarchical visual space also requires spatial information about regions. This information provides contextual cues for higher level reasoning. A procedure for extracting this type of information among regions has been developed.

A reference direction, which is defined as a direction parallel to an image plane, is chosen to determine spatial relationships of regions in the image. Several structural measurements are selected for the regions. They are: the size of the region; the relative position of different adjacent regions along certain look directions; the maximum length of the region along each look direction, the region state, and the number of region boundary pixels. The most important thing is how to measure these spatial relations among regions. The algorithm for this is as follows.

1. First, for the convenience of spatial reasoning, each region in the symbolic image is identified by a unique region identifier. Therefore, the algorithm defines new region indices sequentially for the different categories of objects which are obtained from low-level labeling. For example, for a three digit

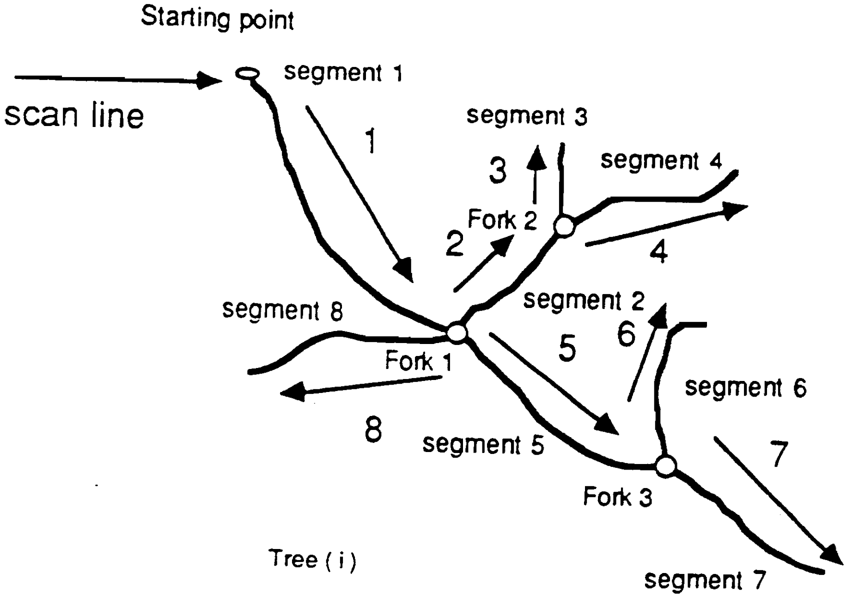


Figure 39. The tracing order.

index, the first digit may represent the original low level labeling. The lower two digits may represent the sequences of regions within each category. Thus, we still can recognize the initial assigned label from the new sequential index for each region.

2. Second, a linear geometric transformation is used to rotate the symbolic image to a position in which the horizontal scan line just is parallel to the reference direction.

3. Third, the region symbolic image is scanned line by line in the reference direction. If the scan line meets a new region label, the scanning process will be interrupted and a tracing process which traces the region external boundary will start. The following is a one-pass, depth first boundary tracing procedure using a left first, clockwise directed four connected neighborhood search technique for tracing a region's boundary:

(1) Record the coordinates of starting point of the region, keep the region always to the right side of tracing direction, and trace the boundary in a clockwise direction.

(2) For each successor, detect the next tracing direction by searching the same label from four connected neighbors in the order starting from the left side of previous direction of motion, then the front, the right, finally the back (Figure 40).

(3) Record the left side adjacent region label, its starting position, and its ending position in the tracing process.

(4) Count the number of pixels of the region boundary in the tracing process.

(5) Mark the region's state, (0 indicates a closed region which lies completely within the image. 1 indicates that the region touches the image boundary.)

(7) If next point=starting point, then stop tracing, go to Step 7, else go to Step 2.

(8) Continue the scanning process from the break point, compute the maximum region length along the reference direction, and compute the region size,

(9) Check every label that the scan line met within a hash table. If the region has been traced, go to Step 7, else go to Step 1 until the last line.

The test result for a test image (Figure 41) by this method can be found in Figure 42.

High level image analysis techniques require rapid access to region information and to the relations between regions. For every region, therefore, the region attributes and the relations of adjacent regions are stored in a large list. A hash function is used for fast searching and accessing of structural information. After that, a property file which contains a list of property values for the above measurements can be created. Since it is the same as the property file produced by the procedure of extraction of line-drawing features, this property file can be also converted into a symbolic database written by PROLOG. Then the spatial reasoning process can be carried out. This procedure has been successfully applied to discriminating water regions from shadow regions in SAR radar images (Qian and Haralick, 1990).

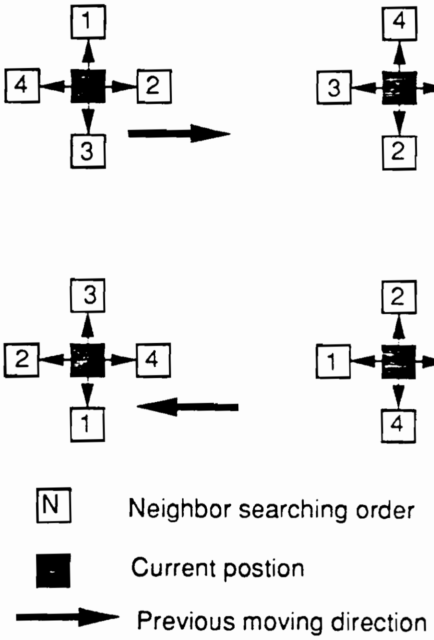


Figure 40. Direction and order of searching: previous motion direction (1) left to right, (2) top to bottom, (3) bottom to top, (4) right to left..

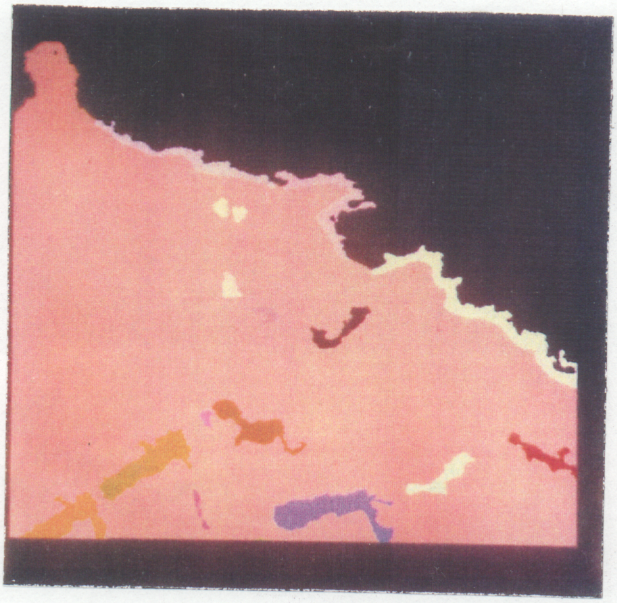


Figure 41. A symbolic image tested by the tracing procedure.

Region Number	Region Index	Maximum Length	Adjacency		State Mark	Region Size	Boundary Length
			R_1	R_2			
1	100	319	300	204	1	89684	2325
2	101	21	300	215	1	148	63
3	200	21	300	100	0	1088	440
4	201	27	100	300	0	1375	541
5	202	18	300	0	0	129	57
6	203	15	300	0	0	107	52
7	204	35	100	300	1	3070	927
8	205	24	300	0	0	203	69
9	206	23	300	0	0	640	227
10	207	14	300	0	0	180	99
11	208	24	300	0	0	1220	339
12	209	16	300	0	0	104	56
13	210	22	300	0	1	1012	377
14	211	34	300	0	0	1777	338
15	212	21	300	0	0	592	201
16	213	31	300	0	0	2235	438
17	214	17	300	0	0	131	66
18	215	32	101	300	1	1946	406
19	216	22	300	0	0	382	89
20	217	16	300		1	296	82
21	300	448	200	100	1	154802	2962

Figure 42. The property list of the tracing results for Figure 41.

7.4 DNESYS–a System Level Uncertainty Reasoning Application

DNESYS is a Drainage Network Extraction System (see Figure 43) for automatically extracting drainage networks from digital elevation data (Qian, Ehrich, and Campbell, 1990). The main purpose of DNESYS is to deduce a complete visual event (drainage networks) at a high level by applying uncertainty reasoning to broken segments, misconnected segments, missing segments, and other defects which usually result from the low level operators. The principles, algorithms, and data structures used in the DNESYS system may be extended to other line reasoning systems and shape completion systems, which are important components of computer vision systems.

7.4.1 The Overview of DNESYS

The determination of drainage networks and drainage basins is one of the more tedious yet important uses of topographic maps, and geographic information systems are now used extensively as a manual aid to facilitate that task. However, the wide availability of digital elevation maps has stimulated attempts to automate the process even further. Significant problems are associated with the use of local point operators for extracting drainage systems from DEM data. Performance can be dramatically improved by making use of spatial reasoning. The DNESYS system described here is capable of performing this task at a high level based on organized expert knowledge. A stream representation called a parameterized directed graph (PDG) is constructed to model a drainage system. The construction of a model begins with an initial pixel labeling procedure. Then a network tracing and property measurement procedure converts the 2-D low-level labeling information into a symbolic database for high-level processing. By applying uncertainty reasoning system developed in previous chapters, evidence collection and uncertain reasoning are performed against the DNESYS knowledge-base that contains the drainage system model and the organized expert knowledge. By discarding erroneous information and supplying missing information, DNESYS produces a complete PDG which can be converted into the final drainage system. The experimental results show that this expert system performs extremely well. Some additional useful features of the DNESYS system are:

1. it is trainable and the knowledge base is easy to modify;

2. it has the flexibility for the user to choose the desired reasoning object—the specific segment, the specific tree, the specific subarea, or the entire database after it has been constructed from the low-level processing results;

3. it can be used as a system for direct acquisition of drainage systems from DEM maps for use in a geographic information system. The results from the DNESYS system are both in 2-D map format and in database format with all required attributes and distinct labels;

4. results register to the original elevation data and will have the same resolution as the original data. Therefore the results can be superimposed on the original elevation data.

The problem of delineating drainage basins may be approached by first solving the problem of determining the drainage network. Then the drainage network and the ridge lines may be used to determine the corresponding basins. The usual method for deriving the drainage network is to trace or digitize streams as recorded on topographic maps. This is not always practical, especially if detailed data are required for a large region, due to the problems in handling and matching data from large numbers of large-scale maps. In addition, conventional cartographic representations of the drainage network are not always satisfactory in the context of a GIS. Drainage information derived from the stream network on topographic maps may be incomplete due to arbitrary definitions of streams, omission of the smallest branches of the drainage network, inconsistencies in interpretation by cartographers, and effects of cartographic generalization.

As a result there may be good reasons to abandon traditional cartographic representations of drainage networks in favor of interpretations of digital elevation data. Once stream channels have been identified within the digital data, many of their most important characteristics, including size, channel length, gradient, and sinuosity, can be calculated. The major problem is to derive the drainage network from the topographic information accurately and efficiently. In principle, the task is straightforward — water drains from high elevation to low along channels that are connected to one another according to well-defined rules. Exceptions are so rare or so easily recognized that they do not influence the usefulness of a successful technique.

The problem is not quite so easy in practice. The sampling of topography by the digital elevation data generates ambiguities, including artificial pits and ridges. Additional errors may be artifacts of the digitization process used to derive the digital elevations from the source maps or photogrammetric models. Due to noise and quantization errors, true valley points may be obscured completely. Streams that flow through narrow gorges may appear to cross ridgelines. In smooth terrain or when a stream passes through a body of water, local features lose their coherence and cannot be traced easily. A practical method of identifying stream channels must be capable of recognizing and resolving such problems.

Most previous efforts to derive drainage information from digital elevation data have used a moving window to derive information concerning the configuration of the land surface within a local neighborhood. For example, the notable techniques proposed by O'Callaghan and Mark (1984) and Jenson (1984) are based on low level pixel labeling by using a local operator and subsequent pixel grouping according to local criteria. Using a local operator, O'Callaghan and Mark first label the drainage direction for each pixel. Then they carry out an iterative computation of *drainage accumulation values*. In each iteration, each pixel is relabeled with a weighted sum of the drainage accumulation values of neighboring pixels that drain into that pixel. Then the drainage channels are labeled according to the accumulation values which are greater than the user-specified threshold. Jenson uses a 3×3 pixel window to label so-called *drainage pixels* by searching for local minima between two of its non-adjacent neighbors. Minima are located in any of 12 configurations within the 3×3 window. Then, based on local criteria, a drainage basin linkage is established from the resulting drainage label map given user-specified distance and elevation thresholds.

Although such approaches produce a satisfactory approximation of portions of a drainage system, we have found that they do not yield a reasonable representation of drainage systems considered as a whole. The deficiencies in these techniques are the following:

- The extracted drainage networks are broken into disconnected segments because the gaps between them may be quite large. It would exceed the ability of these algorithms to establish linkages between spatially separated segments.

- Some stream segments may erroneously be merged with others, even when separated by a ridge line or a saddle point. Local techniques may not be able to reject these kinds of mislabelings.

- A single extracted segment may have two opposing elevation slopes within the segment. This means that the segment has two conflicting flow directions along its length.

- Stream segments may be so thick that the exact stream locations cannot be determined.

DNESYS uses both local operators and global reasoning to extract the drainage networks and ridge lines from the DEMs by merging lower level structures into globally consistent networks. Unlike other techniques, the DNESYS system organizes the initial labels directly into more global abstractions called *segments* without a pixel-by-pixel grouping process. The system views these as the drainage primitives which are more reliable than individual pixels for the grouping and reasoning process later on. An attribute vector containing all factors that participate in the reasoning process is assigned to each segment. After that, a hypothesis generator proposes links between pairs of spatially related segments. For each pair of segments, N_p predicates are evaluated to produce an N_p-ary evidence vector for asserting a new segment or rejecting it. Finally, all segments with monotonically decreasing elevations are connected to their *stream tree*, which is a hierarchical data structure used to represent a portion of the stream network. The roots of all the stream trees correspond with segments that intersect the edge of the DEM and carry water out of the map area. Since there are many segments and many trees, all segments are uniquely labeled by their segment numbers and tree numbers. DNESYS also extracts topographic ridge lines as part of the reasoning process.

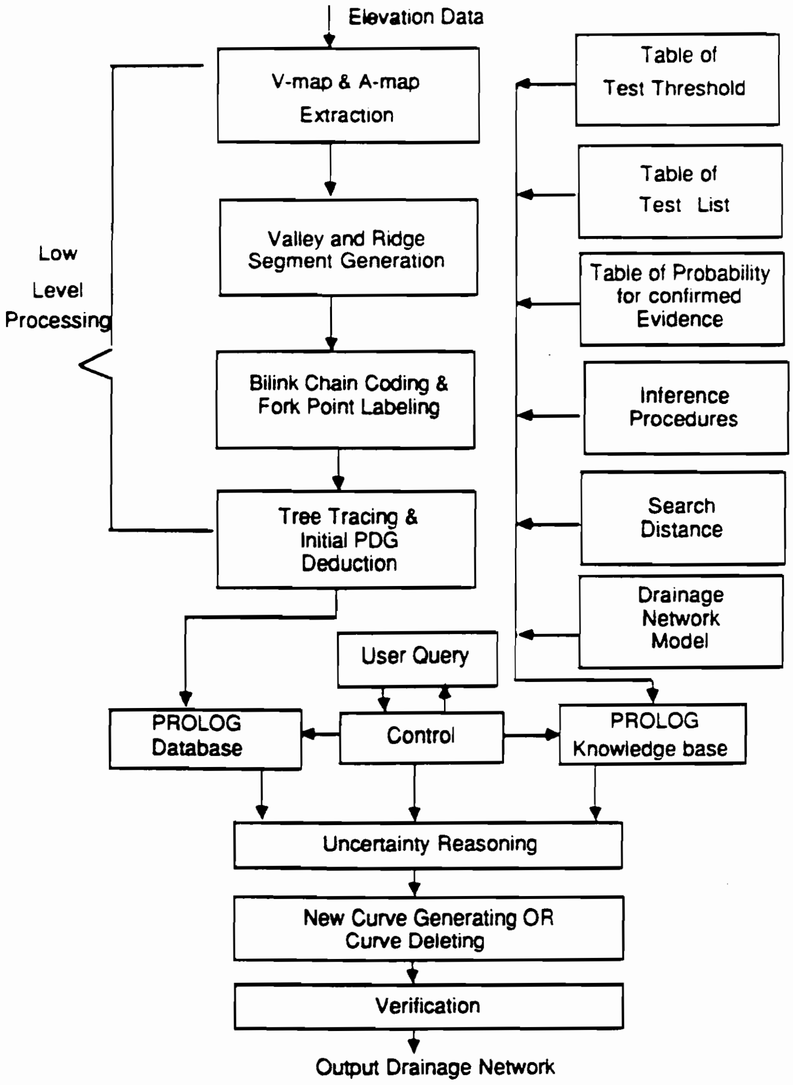


Figure 43. The DNESYS system diagram.

The DNESYS system consists of three separate parts: low-level primitive labeling and attribute vector measurement, high-level spatial reasoning, and an interface between these two parts. This interface includes a set of special procedures to construct the PROLOG database and to convert the PROLOG database back into a 2-dimensional image. These low-level procedures are written in RATFOR and are implemented as GIPSY (the General Image Processing System) commands. The high-level expert is written in PROLOG to carry out the uncertainty reasoning. The interface is designed to construct a PROLOG data base automatically with an intermediate representation of all useful spatial information about drainage primitives, which the high-level expert can access efficiently to determine relevant events.

7.4.2 The Model of Drainage Networks

To reason about an event at a high level, it is important to have knowledge about the event at that level. The knowledge and model of drainage networks described in this section will be organized as part of the knowledge base in the other sections.

The principal topographical features upon which drainage networks are based are called *valley segments*. A valley segment is a connected chain of pixels which are local minima in the direction normal to the segment. Hydrographically, a drainage network consists of all valley segments on which runoff is sufficiently concentrated and through which the flow is downhill. These are called *stream segments*. Valley segments and stream segments differ in that valley segments are measured features whereas stream segments are components of an actual drainage network. Due to soil porosity, geologic structure, slope, and climatic conditions, many relatively large and deep valleys contain no definite stream channel. These are still included in the network determined by DNESYS. A single unbranched stream segment may have one open end which may be considered to be its flow source. However, its other end must connect to other channels which drain the water that it collects.

In DNESYS, the model of a drainage system is a *parameterized directed graph* (PDG) deduced from the DEM's data. The graph is, in fact, a set of loop-free stream trees, and the parameters are the attributes associated with each stream segment. The nodes of the graph are either source nodes to which no other stream segments connect, sink nodes which are on the edge of the DEM, or fork nodes where stream segments merge together. Water flow direction is assumed to be from the source nodes toward the sink nodes. The total flow into a node must equal the flow out from the node (flow conservation). IN degree and OUT degree are used to describe the connectivity of a node. Source nodes have IN degree zero and OUT degree one. For fork and sink nodes, the IN degree is equal to the number of directly connected branches which have negative end-elevation slopes, and the OUT degree is equal to the number of directly connected branches which have positive end-elevation slopes. Any node with zero OUT degree is a sink node. The branches of the PDG are the stream segments, and the reasoning process will determine which valley segments are stream segments and which are not. The parameters associated with each stream segment include length, drainage direction, and the elevation slope of the segment.

Stream segments have the following properties:

1. The width is only one picture element.

2. The length is arbitrary.

3. The elevation values along it decrease monotonically in the flow direction.

4. There is no fork point on the segment between its endpoints.

5. The elevation value of any point on the segment is a local minimum point of the stream segment cross section.

Any branch which does not satisfy above five conditions is called an *inconsistent branch* in the PDG. A graph with any inconsistent branch is an *incomplete PDG*. For example, a valley segment between two adjacent nodes but with conflicting flow directions is an inconsistent branch in that graph, since it does not satisfy Condition 3. It must be split into several consistent branches by inserting nodes at suitable points on the segment so that all five conditions are satisfied. Since the inserted nodes will have either zero OUT degree or zero IN degree, linkages to other branches must be located. A graph with a sink node occurring anywhere except on the map edge is also called an incomplete PDG since sink nodes don't make sense except at the physical edges of the DEM where valleys are discontinuous.

When the reasoning process begins, each branch consists of a source node and a sink node, and sinks and sources are joined as the merging process progresses. However, in some cases global evidence will cause sinks and sources to be connected even though that is not supported by local evidence. The the direction of a branch of the PDG is the flow direction, i.e. the direction of decreasing elevation along that segment.

Branching angle is another important issue in reasoning about the structure of a drainage network. The angular relationships among segments incident at a node are determined by drainage slopes near the node; the angles between inward flowing streams are usually less than 90 degrees (Zerniz, 1932; Haralick, Campbell, and Wang, 1985). In the literature there are two principal theoretical models for determining branching angle from slope—the Hortonian model (Horton, 1945) and the Minimum Power model (Howard, 1971). From the theoretical point of view, Roy found that both angular geometry models are based upon the optimality principle, which implies that the drainage system must "perform its task" with maximal efficiency and minimum cost (Roy, 1983). Because both models yield similar theoretical angles and because the Hortonian model is simpler to implement, DNESYS uses the Hortonian model as part of its knowledge base.

Under the assumption that overland flow on the valley slopes follows the line of steepest gradient, Horton's model hypothesizes that

$$\cos\theta = \frac{S_c}{S_g}, \tag{7.13}$$

where θ is the angle between the line of overland flow and the receiving stream, S_c is the stream gradient, and S_g is the ground slope. For the case that a receiving stream is joined by a single tributary stream, Horton modified his model to

$$\cos\theta = \frac{S_0}{S_1}, \tag{7.14}$$

where θ is the junction angle, S_0 is the gradient of receiving stream, and S_1 is the gradient of tributary stream as in Figure 44. Usually, $S_1 > S_0$. However, when two streams merge with nearly equal gradient, i.e. $S_1 \approx S_0$, this model will erroneously predict a zero junction angle. Nevertheless, this model clearly states that the junction angle θ must vary within a range of $0 < \theta < 90$ degrees. This is true for most practical cases.

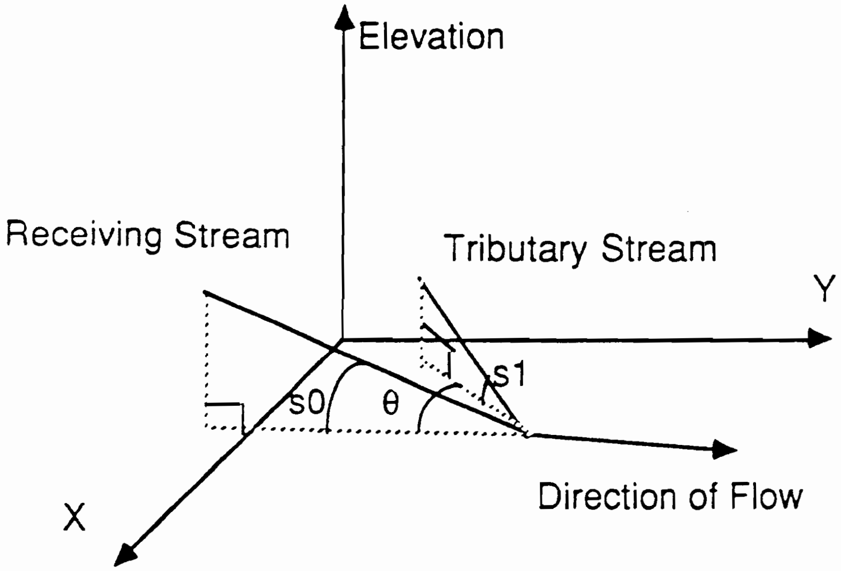


Figure 44. The stream branching model (Hortonian model).

In DNESYS, an incomplete parameterized graph with sink nodes, source nodes, and possibly inconsistent branches is obtained from low-level processing. The graph consists of many disjoint trees, and it is the function of the high-level reasoning process to make mergers of trees and to split branches in order to remove inconsistencies due to conflicting flow directions.

7.4.3 Evidence Collection and Uncertainty Reasoning

The procedures of low level processing including the initial drainage point marking, network tracing, and property measuring, can be found in (Qian, Ehrich, Campbell, 1990) and are omitted here. The method presented in this section is based upon the output information of that low level processing.

Reasoning about a drainage system at a high level requires that DNESYS have the ability to handle uncertainty. Uncertainty is caused both by imprecise data and by imprecision in the rules used for extracting the drainage networks. In the reasoning process, the evidence from the observed data is often incomplete or conflicting; some rules are based simply upon intuitive knowledge. Therefore, there are no simple measures or rules for deciding how a pair of broken segments should be connected or how an inconsistent segment should be split. To reason with uncertainty and to resolve possible conflicts in the data produced by the low-level measurements, DNESYS applies the uncertainty reasoning algorithm to these problems.

The uncertainty reasoning system is used to construct a global representation using partial representations, each of which demonstrates local consistency. Thus, it can simultaneously form and evaluate alternative hypotheses. An interpretation of a portion of a drainage network depends not only upon each individual piece of evidence but also upon interactions among various pieces of evidence. Then the problem becomes one of multicriteria optimization. To solve it requires consideration of all the evidence from each partial representation and utilization of higher level knowledge to resolve conflicts.

A set of functions are designed to extract specific partial evidence in support of all hypotheses that are generated. The output of the feature extractors are inputs to a set of tests which are organized as a procedural part of the knowledge base of the expert system. These tests are made at higher-level processing stages. The test results are used as inputs to the process of reasoning with uncertainty, which is discussed later. Let $\tau_1, \tau_2, \ldots, \tau_8$ be the set of tests. The following are the tests made on the low-level spatial features in the PROLOG database.

Distance Test τ_1: This test provides support for the connection of two segments based upon proximity of their endpoints. Let $P_i(x_i, y_i)$ and $P_j(x_j, y_j)$ be endpoints of two arbitrary segments; then the geometric distance between P_i and P_j is given by the usual Euclidean distance

$$D = \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2}. \tag{7.15}$$

Given a threshold, T_1, then

$$\tau_1 = \begin{cases} true, & \text{if } D \le T_1; \\ false, & \text{otherwise.} \end{cases} \tag{7.16}$$

Curvature Similarity Test τ_2: Since most stream segments are locally smoothly connected, this test is for confirming this property. Let CV_1 and CV_2 be the mean curvatures of the endpoints of two candidate segments and let T_2 be a threshold. Then CV_i is defined as

$$CV_i = \frac{1}{n}\sum_{j=1}^{n}\frac{\theta_{j1}-\theta_{j2}}{\Delta S_j} \tag{7.17}$$

where $(\theta_{j1} - \theta_{j2})$ is the local measure of change of the orientation of the tangent vector and ΔS_j is the change in arc length corresponding to above angle changes. Thus

$$\tau_2 = \begin{cases} true, & \text{if } |CV_1 - CV_2| \leq T_2; \\ false, & \text{otherwise.} \end{cases} \tag{7.18}$$

Segment Length Test τ_3: The system assigns higher credibility to longer segments. All isolated valley points are ignored. Let L_i be the lengths of two distinct segments and let T_3 be the test threshold. Then

$$\tau_3 = \begin{cases} true, & \text{if } L_i \geq T_3 \text{ for } i = 1,2; \\ false, & \text{otherwise.} \end{cases} \tag{7.19}$$

Elevation Slope Test τ_4: This test is to confirm that if the stream flow directions of two segments are consistent, then the signs of the elevation slopes at the nearest endpoints of these two segments should be opposite according to the reference direction. The positive reference direction is defined as the direction from the center of the segment toward the endpoint. The elevation slope of the endpoint is calculated from the end section of valley segment. Then let ES_1 and ES_2 be the elevation slopes at the ends of the valley segment.

$$\tau_4 = \begin{cases} \mathit{true}, & \text{if } (ES_1 \geq 0 \text{ and } ES_2 \leq 0) \text{ or } (ES_1 \leq 0 \text{ and } ES_2 \geq 0); \\ \mathit{false}, & \text{otherwise.} \end{cases} \tag{7.20}$$

Elevation Similarity Test τ_5: This test is to confirm that the mean elevation difference between the endpoints of two mergable segments does not exceed a threshold value, given the search distance. Let ME_1 and ME_2 be the mean elevations in a local neighborhood near the ends of the segments, let T_5 be the threshold for this test, and let β_1 and β_2 be the predicates

$$\begin{aligned} \beta_1 &= ES_1 < 0 \text{ and } ES_2 > 0 \text{ and } 0 < (ME_1 - ME_2) \leq T_5 \\ \beta_2 &= ES_1 > 0 \text{ and } ES_2 < 0 \text{ and } 0 < (ME_2 - ME_1) \leq T_5. \end{aligned} \tag{7.21}$$

Then

$$\tau_5 = \begin{cases} \mathit{true}, & \text{if } \beta_1 = \mathit{true} \text{ or } \beta_2 = \mathit{true}; \\ \mathit{false}, & \text{otherwise.} \end{cases} \tag{7.22}$$

Forward Elevation Test τ_6: This test is based on the knowledge that the elevation slope does not usually change rapidly in a local neighborhood. Therefore if a segment is extrapolated toward another endpoint based on its slope, there should not be a large disparity between the actual elevation and extrapolated elevation at that endpoint. Let EE_1 and EE_2 be the elevations at the ends of two valley segments, let ES_1 and ES_2 be the corresponding elevation slopes, let D be the Euclidean distance between these two ends, and let T_6 be the test threshold.

$$\tau_6 = \begin{cases} \mathit{true}, & \text{if } |EE_1(1 + ES_1 D) - EE_2| \leq T_6 \text{ or } |EE_2(1 + ES_2 D) - EE_1| \leq T_6; \\ \mathit{false}, & \text{otherwise.} \end{cases} \tag{7.23}$$

Orientation Test τ_7: Since local curvatures of valley segments tend to be low, one would expect that a segment connecting the endpoints of two disjoint segments would not introduce sharp local curvatures. Therefore τ_7 compares the orientations at segment endpoints and the orientation of a segment between them. The smaller the orientation differences, the higher the probability that the endpoints should be merged. The following three angle differences completely specify the geometric orientation and transition properties of three joined segments.

Let P_1 and P_2 be the endpoints of two distinct segments. Let θ_3 be the direction of P_1P_2, θ_1 be the angle of the tangent vector to the flow direction at P_1, and let θ_2 be the angle of the tangent vector to the flow direction at P_2 as in Figure 45. Then the orientation test checks the orientation of the line P_1P_2 with respect to these orientations. Specifically,

$$\tau_7 = \begin{cases} \mathit{true}, & \text{if } |\theta_3 - \theta_2| \le T_{71} \text{ and } |\theta_3 - \theta_1| \le T_{72} \text{ and } |\theta_1 - \theta_2| \le T_{73}; \\ \mathit{false}, & \text{otherwise.} \end{cases} \tag{7.24}$$

where T_{71}, T_{72}, and T_{73} are the thresholds for this test.

Ridge Intersection Test τ_8: This test is for checking for a ridge crossing in the gap between the endpoints of two distinct segments. Let R be any ridge line. Let $P_i(x_i, y_i)$ and $P_j(x_j, y_j)$ be the endpoints of two distinct segments. Then the straight line P_iP_j is defined by the slope

$$y = \frac{\Delta y}{\Delta x}(x - x_j) + y_j \tag{7.25}$$

where $x_i \le x \le x_j$, $y_i \le y \le y_j$, $\Delta y = y_j - y_i$, and $\Delta x = x_j - x_i$.

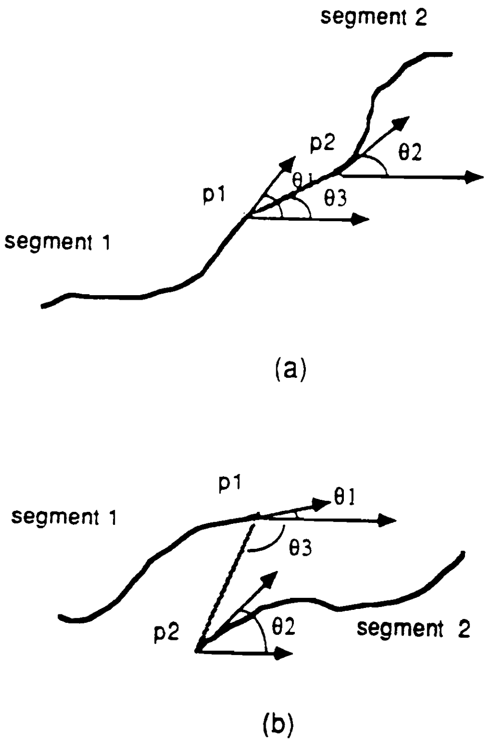


Figure 45. The angles in the orientation test.

$$\tau_8 = \begin{cases} true, & \text{if } P_iP_j \text{ does not intersect R;} \\ false, & \text{otherwise.} \end{cases} \qquad (7.26)$$

DNESYS maps the evidence that is extracted from the observed data into a Boolean logic set by using the functions $\tau_1, \tau_2, ..., \tau_8$. In general, each τ_i is a function of a data vector Z_i and a parameter vector T_i, that is, $\tau_i = f_i(Z_i, T_i)$. In the case of $1 \leq i \leq 7$, the parameters are all thresholds which are determined in one of two ways — from training data or by experts. An individual hypothesis is supported according to the success of its corresponding tests. Each τ_i is merely a contribution to the overall support, and the influence of τ_i on hypothesis h is determined either by a conditional probability $p(h|\tau_i)$ or by experts subjectively. DNESYS interprets this probability as evidential strength or as a degree of confirmation. An inference network that propagates evidence strength can then be constructed.

Let h be a hypothesis, let $\tau_1, ..., \tau_8$ be the evidence for that hypothesis, and let $p(h|\tau_1), ..., p(h|\tau_8)$ be the evidence strengths. Note that several of the τ_i may logically conflict, but other τ_i may still verify the hypothesis. DNESYS chooses a belief function which is able to synthesize all partial representations of evidence and to achieve the multicriteria optimization. DNESYS accumulates partial supports from different evidence iteratively throughout the inference network. In DNESYS, a frame of discernment Θ for produce a complete PDG, consists of the following exhaustive and mutually exclusive subsets:

1. A source node is connected to a sink node.

2. A source node remains open (ie, not connected to a sink node).

3. A source node is backward merged to a sink point which splits an inconsistent branch (with a *V-shaped* elevation profile) into two consistent branches.

4. A sink node is forward merged to a source point which splits an inconsistent branch (with a *A-shaped* elevation profile) into two consistent branches.

5. A sink node is located at the boundary of the map and remains open.

Each subset of Θ corresponds to a hypothesis. Frame of discernment Θ delimits a sample space which contains all possible linking hypotheses, only one of which is true at any one time. The evidence provided by each test τ_i is represented as a basic probability assignment (*bpa*) over the hypotheses discerned by a frame Θ. The *bpa* represents the impact of each distinct piece of evidence on the subsets of a frame Θ. Applying Dempster's rule, DNESYS pools the multiple bodies of evidence obtained from the different tests to get new belief functions. Finally, a single hypothesis among others is selected, if and only if the belief function has a maximum value which is greater than a threshold specified in the knowledge base. When there are no remaining inconsistencies, the complete *PDG* contains the results of the reasoning process. Global consistency is achieved for two reasons: individual segments are consistent with one another and with ridge segments, and the *PDG* structure itself precludes the formation of global loops among the valley segments.

7.4.4 Data Structure and Control Mechanism

The high-level expert of DNESYS consists three major parts: the database, the knowledge base, and the inference procedures. The data structures and control mechanisms similar to those described in Chapter V are used in DNESYS.

In the DNESYS database, all valley segments and ridge segments are structured as hierarchical lists. These lists are treated as PROLOG fact rules asserted in the system database. The DEM coordinate space is partitioned into blocks, and for each block an endpoint index is constructed for those segments having endpoints in that block. A space spanning index is used to indicate which blocks are spanned by each segment. The space spanning index is useful for the ridge intersection test and for the merging that may occur when an inconsistent segment is split. Each element in the index table has the form

```
((INDEX
  *NODE-TYPE {SR: source node, SN: sink node, IN: inconsistent node}
  *INDEX-TYPE {1: spanning index, 0: open-end index}
  *SEARCH-FLAG {1: done, 0: to be tested}
  *END-TYPE {1: start point, 0: end point}
  *HORIZONTAL-INDEX
  *VERTICAL-INDEX
  *SEGMENT-NUMBER-IN-TREE
  *TREE-NUMBER))
```

The data structure for drainage primitives includes four parts: a unique segment label, a segment parameter list, a list of sample points at each end, and an extended segment chain code which includes the elevation and direction. The data structure has the form

```
( ( SEGMENT-NUMBER *s
    TREE-NUMBER *t
   ( (START-TYPE *sp)
    (STOP-TYPE *tp)
    (SEGMENT-LENGTH *sl)
    (START-MEAN-ELEVATION *sme)
```

```
 (STOP-MEAN-ELEVATION *tme)
 (START-ELEVATION *se)
 (STOP-ELEVATION *te)
 (START-MEAN-CURVATURE *smc)
 (STOP-MEAN-CURVATURE *tmc)
 (START-ELEVATION-SLOPE *ses)
 (STOP-ELEVATION-SLOPE *tes)
 (START-DIRECTION *sd)
 (STOP-DIRECTION *td)
)
((START-SAMPLING-POINTS *x11 *y11 *x12 *y12)
 (STOP-SAMPLING-POINTS *x21 *y21 *x22 *y22)
)
((COMPOSED-SEGMENT-TRAIN-CODE
  (POINT 1    *x
              *y
              *elevation
              *direction
  POINT 2    *x
              *y
              *elevation
              *direction
              .
              .
              .
  POINT N    *x
```

```
                    *y
                    *elevation
                    *direction)
          ))
         ))
```

This database contains all the information required to reconstruct a raster image after reasoning has been completed. The data structure for drainage primitives including valley lines and ridge lines is a two-level hierarchical list. As soon as the segment number and the tree number have been chosen in the deduction process, any one of the three elements in the first-level list can be moved to a global area called the blackboard (Lesser and Erman, 1977). Then according to the END-TYPE in the index, corresponding elements in the second-level list can be asserted as the independent fact rules with their own predicates in the blackboard. Thus these data are temporarily shared by the hypothesis generation and verification processes. They are erased immediately from the blackboard before the data for another segment are moved in.

The knowledge base consists of a test name list, a threshold table, a table of evidence strengths (*bpa*) for confirmed or unconfirmed $\tau_i's$, a spatial search distance, and inference procedures. The test name list contains the symbolic names for $\tau_1, ..., \tau_N$. Through this list, DNESYS specifies how many tests and which tests should be carried out in the system. The threshold table contains all threshold values $T_1, ..., T_N$ for their corresponding tests. According to this table, DNESYS maps the evidence that is extracted from the observed data into a Boolean logic set. The table of *bpa*s contains the evidential strengths, $p(h|\tau_i)$. The spatial search distance provides an upper limit to the spatial neighborhood involved when the algorithm focuses attention on a particular segment. This distance specifies how many neighboring blocks should be searched to generate hypotheses. Finally, the inference procedures are used to implement the evidential reasoning using the D-S model. The basic data structure in the knowledge base is the frame. Each frame has a set of associated slots, which are empty buffers that will hold particular types of information. Some of these are filled with the corresponding values assigned by experts or obtained from training data, and the rest are filled as the reasoning process takes place. Through the knowledge base frames, the user can specify or modify the knowledge base freely except for the inference network, which is relatively fixed.

The main controller in the high level expert of DNESYS is based upon the database and knowledge base structures just described. It uses the spatial relevancy finder to select an open node (either a source node or a sink node) and to calculate all possible search indices which fall in a search radius. Then the hypothesis generator generates hypotheses based upon the search index list. By consulting the test name list in the knowledge base, the tests described previously are computed for hypotheses. All the parameters concerned with a test are accessed. The evidence extractor determines the Boolean values of $\tau_1, \tau_2, ..., \tau_8$ according to the threshold values imbedded in the knowledge base for each test. Then the initial *bpa*s, $p(h|\tau_i)$, are extracted from the knowledge base by table look-up. These values are treated as the evidence strengths and are propagated into the built-in inference network. This is done by filling an evidence frame which contains the evidence names and the corresponding slots for the evidence strengths. For each hypothesis, a belief function which accumulates all the evidence can be computed recursively from the evidence frame following the inference network. Finally, N belief functions are calculated for N hypotheses generated according to the N pairs of indices. These are put into a hypothesis frame for hypothesis selection. The decision maker or the hypothesis selector examines that frame to find the maximum belief function value and compares it to a threshold specified in the knowledge base. A hypothesis is accepted if and only if its belief function is above the threshold and if its value is maximal among the alternatives. The other hypotheses are rejected. The selected hypothesis invokes a curve fitting procedure to generate the digitization of a link between the endpoints of two stream segments. After that, the relevancy finder will pick up the next open node by searching the inverted index downward from the current position. Then the controller will repeat the above processes, until the inverted index is exhausted. A complete drainage *PDG* and its corresponding drainage networks can now be produced.

7.4.5 Curve Interpolation

If a hypothesis is selected that requires reconnections among the valley segments, then the next problem is to determine how the connections should be made at the pixel level. One way is to extend the end of one segment toward the other pixel by pixel. However, it is easy to become lost in the middle if the gap is wide, if the information in the gap is very ambiguous, or if there is missing information in the gap due to occlusion. To overcome this difficulty, the DNESYS system uses a parametric cubic spline to connect segments smoothly by maintaining continuous curvature at segment endpoints.

The expression of a single parametric cubic spline segment $\boldsymbol{\alpha}$, in terms of arc length S, is given by

$$\begin{aligned}\boldsymbol{\alpha}(S) &= \sum_{i=1}^{4} \mathbf{B}_i S^{i-1} \\ &= \mathbf{B}_1 S^0 + \mathbf{B}_2 S + \mathbf{B}_3 S^2 + \mathbf{B}_4 S^3\end{aligned} \tag{7.27}$$

where $0 \leq S \leq S_{max}$.

Let $\boldsymbol{\alpha}(S_k) = \boldsymbol{\alpha}_k$ and $\boldsymbol{\alpha}(S_{k+1}) = \boldsymbol{\alpha}_{k+1}$, where S_k and S_{k+1} are two adjacent point values within the range of parameter S. Then $\boldsymbol{\alpha}'_k$ is the value of the first derivative of $\boldsymbol{\alpha}(S)$ with respect to S at $S = S_k$, etc.. Four equations can be obtained by taking derivatives of Equation 7.27 with respect to S and setting $S = S_k$ and $S = S_{k+1}$ respectively.

$$\boldsymbol{\alpha}(S = S_k = 0) = \boldsymbol{\alpha}_k = \mathbf{B}_1 \tag{7.28.a}$$

$$\boldsymbol{\alpha}(S = S_{k+1}) = \boldsymbol{\alpha}_{k+1} = \mathbf{B}_1 + \mathbf{B}_2 S_{k+1} + \mathbf{B}_3 S_{k+1}^2 + \mathbf{B}_4 S_{k+1}^3 \tag{7.28.b}$$

$$\boldsymbol{\alpha}'(S = S_k = 0) = \boldsymbol{\alpha}'_k = \mathbf{B}_2 \tag{7.28.c}$$

$$\boldsymbol{\alpha}'(S = S_{k+1}) = \boldsymbol{\alpha}'_{k+1} = \mathbf{B}_2 + 2\mathbf{B}_3 S_{k+1} + 3\mathbf{B}_4 S_{k+1}^2 \tag{7.28.d}$$

The coefficient vectors $\mathbf{B}_i$ are then solved from above equations:

$$\mathbf{B}_1 = \boldsymbol{\alpha}_k \tag{7.29.a}$$

$$\mathbf{B}_2 = \boldsymbol{\alpha}'_k \tag{7.29.b}$$

$$\mathbf{B}_3 = \frac{3(\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k)}{S_{k+1}^2} - \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}} - \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}} \qquad (7.29.c)$$

$$\mathbf{B}_4 = \frac{2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})}{S_{k+1}^3} + \frac{\boldsymbol{\alpha}'_k}{S_{k+1}^2} + \frac{\boldsymbol{\alpha}'_{k+1}}{S_{k+1}^2}. \qquad (7.29.d)$$

Given N data points through which the curve must pass, to generate the curve position vector $\boldsymbol{\alpha}_i(S)$, $1 \leq i \leq N$, we need to know the tangent vector $\boldsymbol{\alpha}'_i$. In DNESYS, two points at the end of each segment are chosen for fitting the interpolating cubic as shown in Figure 46.

By using the natural end condition, $d^2|\boldsymbol{\alpha}|/dS^2 = 0$ (Rogers and Adams, 1976), we have

$$\boldsymbol{\alpha}'_1 + \frac{1}{2}\boldsymbol{\alpha}'_2 = \frac{3}{2}\left(\frac{\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1}{S_2}\right) \qquad (7.30.a)$$

$$S_3\boldsymbol{\alpha}'_1 + 2(S_3 + S_2)\boldsymbol{\alpha}'_2 + S_2\boldsymbol{\alpha}'_3 = \frac{3}{S_2 S_3}(S_2^2(\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2) + S_3^2(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1)) \qquad (7.30.b)$$

$$S_4\boldsymbol{\alpha}'_2 + 2(S_3 + S_4)\boldsymbol{\alpha}'_3 + S_3\boldsymbol{\alpha}'_4 = \frac{3}{S_3 S_4}(S_3^2(\boldsymbol{\alpha}_4 - \boldsymbol{\alpha}_3) + S_4^2(\boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2)) \qquad (7.30.c)$$

$$2\boldsymbol{\alpha}'_3 + 4\boldsymbol{\alpha}'_4 = 7.overS_4(\boldsymbol{\alpha}_4 - \boldsymbol{\alpha}_3). \qquad (7.30.d)$$

The above system can be written in matrix form as

$$\mathbf{C}\boldsymbol{\alpha}' = \mathbf{B}, \qquad (7.31)$$

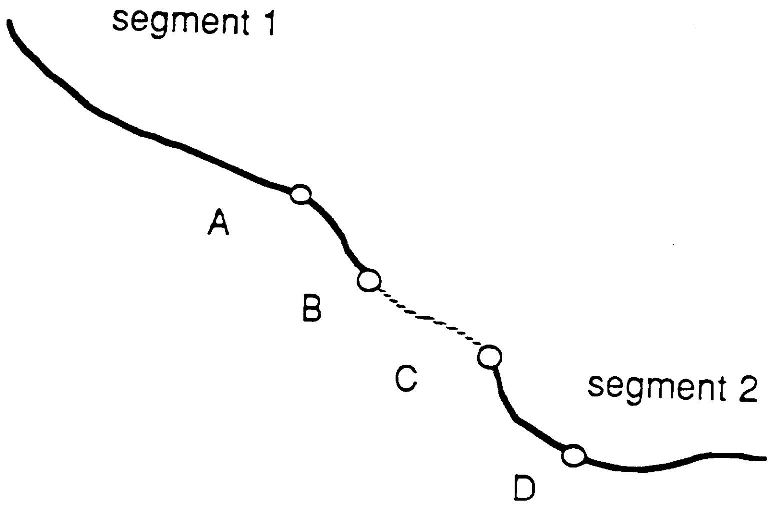


Figure 46. Control points in the curve fitting.

where $\mathbf{C}$ is a 4×4 matrix, $\boldsymbol{\alpha}'$ is a 4×1 variable matrix, and $\mathbf{B}$ is a 4×1 matrix. Then the matrix equation can be solved for $\boldsymbol{\alpha}'$ by matrix inversion. Substituting $\boldsymbol{\alpha}'$ into Equation 7.27, one obtains the coefficient matrix $\mathbf{B}$. Thus, the cubic spline segment is specified by

$$\begin{aligned}\boldsymbol{\alpha}(S) = \boldsymbol{\alpha}_k + \boldsymbol{\alpha}'_k S + (\frac{3(\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k)}{S^2_{k+1}} - \frac{2\boldsymbol{\alpha}'_1}{S_{k+1}} - \frac{\boldsymbol{\alpha}'_2}{S_{k+1}})S^2 \\ + (\frac{2(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})}{S^3_{k+1}} + \frac{\boldsymbol{\alpha}'_1}{S^2_{k+1}} + \frac{\boldsymbol{\alpha}'_{k+1}}{S^2_{k+1}})S^3\end{aligned} \tag{7.32}$$

where $0 \leq S \leq S_{k+1}$ and $1 \leq k \leq 3$.

7.4.6 Verification

Curve $\boldsymbol{\alpha}$ is a continuous curve superimposed on a discrete sampling grid. By tracing this curve, grid points near the curve are marked as component pixels for a new discrete segment connecting the endpoints of the original segments. Since most drainage networks are locally smooth, the fitted curve tends to be close to the shape of the true stream segment. A problem that arises from the computation of the spline independent of external constraints is that the generated curve may overlap with a few ridge points. This may occur if the resolution of the original DEM data is too coarse, if the generated curve bridges segments near the flow sources, or if the gap between the two segments is too wide. In fact, the real drainage network patterns are often more variable than those of the abstract model in our knowledge base. This is not surprising since natural streams are not so smooth and well behaved as the model.

The overlap problem can be solved by adding a verification stage to the reasoning process. Each hypothesis that has been selected by high level reasoning is verified by evidential reasoning at the pixel level under the guidance of the high level results. A belief function is computed for each pixel near those that are inserted as well as for those on all the selected valley segments.

The belief function selects between exactly two hypotheses for each selected pixel–stream or no stream, and the stream paths are corrected based on the results. If a ridge line intersects a valley segment, the entire segment may be removed. The evidence for the verification stage consists of:

1. The high-level reasoning results.
2. The V-shaped map.
3. The A-shaped map.
4. The flow-direction map.
5. The elevation data.

A pixel is verified as a component of a drainage network if its belief value is higher than a preset threshold. The hypothesis verification at the low level provides an opportunity to correct errors made at higher levels, and final decisions are based on reasoning both at high levels and at low levels. The reasoning process in the verification stage is similar to the high level reasoning process, and the details are omitted here due to space considerations.

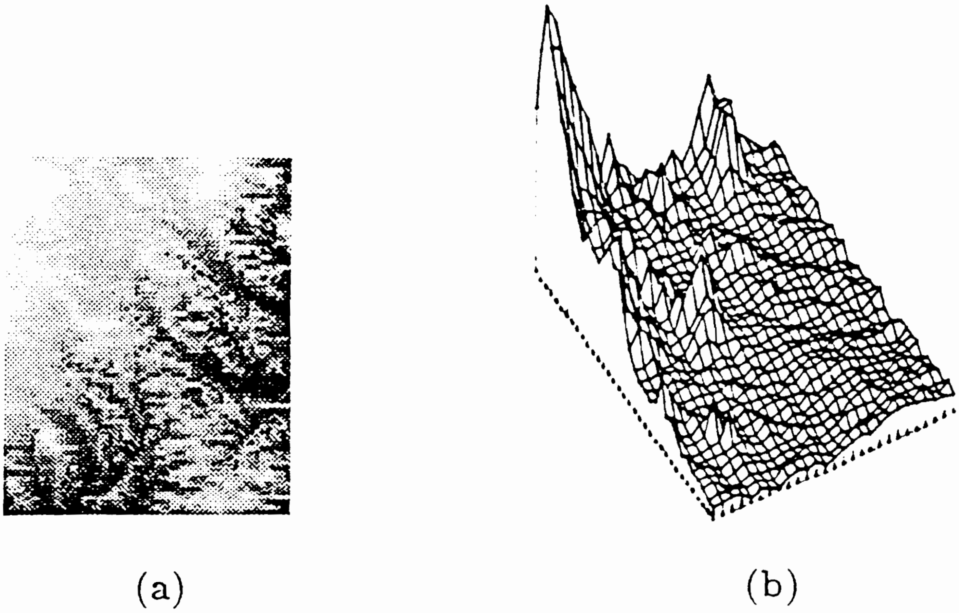
7.4.7 Experimental Results

Two sets of DEM data with different resolutions from the Stanardsville, Virginia and the Hamburg, Virginia USGS 7.5 minute topographic quadrangles were used in our experiments. The data for the Stanardsville area were derived by manual digitization of elevations at the centers of 200 meter grid cells, referenced to the UTM grid system, as shown on USGS 7.5 minute quadrangles. The data for the Hamburg area were collected as a digital elevation model (DEM) by the U.S. Geological Survey (Elassal and Caruso, 1983). The 7.5 minute DEMs represent digital elevations for areas corresponding to USGS 7.5 minute quadrangles published at 1:24,000. Each DEM represents the earth's topography as a regular array of data referenced to the UTM grid system. Elevations for this grid are derived either by manual or automated compilation of stereo aerial photography at a scale of about $1 : 80,000$. Values derived from the stereo models are then processed to represent elevations on the terrain spaced at intervals of about 30 meters. Each value is an integer representing topographic elevation in meters.

7.4.7.1 High Level Reasoning Results

In the first data set, peaks of the Blue Ridge mountains occupy the northwestern corner of the Stanardsville quadrangle, with drainage to the southeast to the upper piedmont surface through South River and Conway River. This drainage forms a dendritic pattern, with relatively steep channel gradients in the mountains, and relatively low gradients once the streams reach the dissected piedmont surface.

The original DEM data of size 70 x 55 pixels are shown in Figure 47(a). The topographic features plotted from the DEM data are shown in Figure 47(b). The V-map which contains the possible valley points marked by the V-shaped operator is shown in Figure 48(a). The A-map which contains possible ridge points marked by A-shaped operator is shown in Figure 48(b). From these two maps, it can be seen that the topography contains pits, saddle points (where valleys and ridges



(a)

(b)

Figure 47. The original DEM map and its surface plot. (a)The original DEM map (size: 70 × 55). (b)The surface plotted from the DEM map of (a).

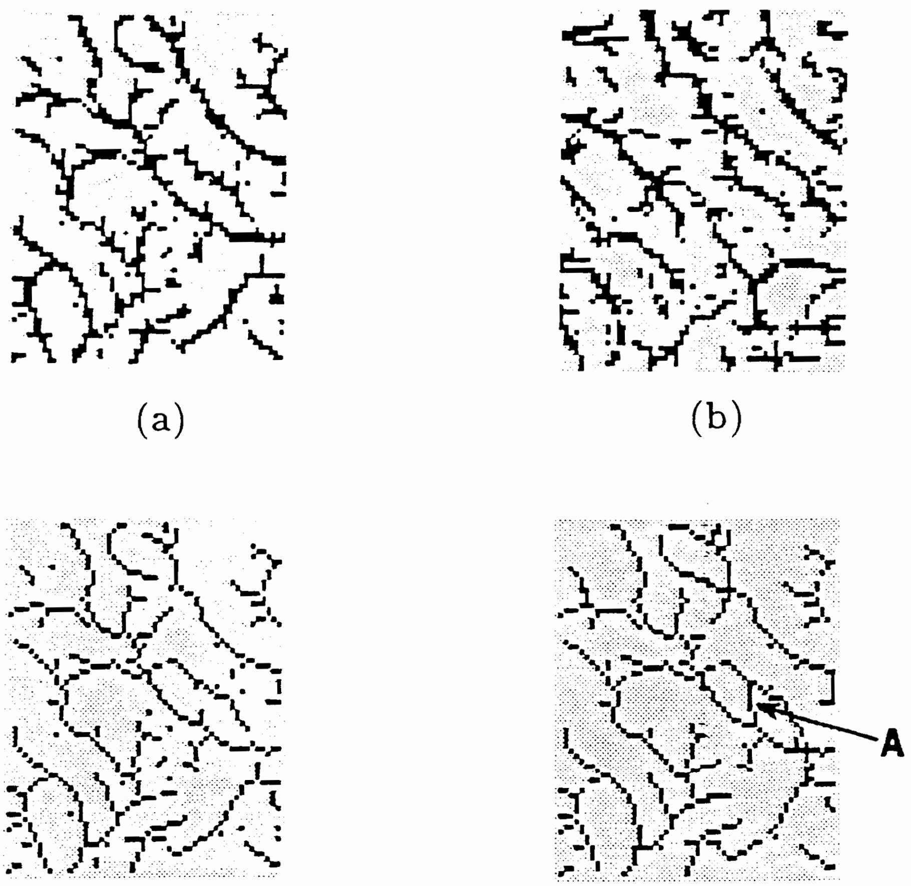


Figure 48. (a)The V-shaped map for Figure 47(a). (b)The A-shaped map for Figure 47(a). (c) The Valley segment map for Figure 47(a). (d) The drainage network after reasoning..

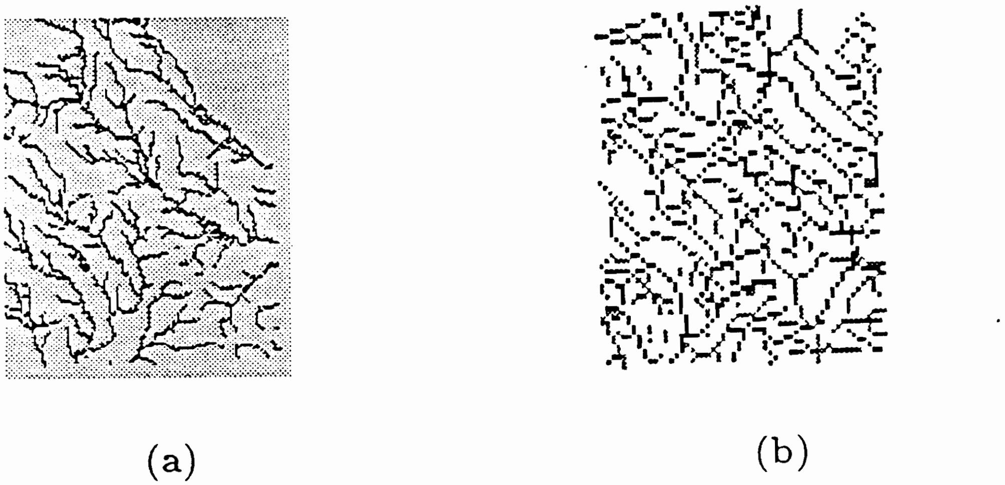


Figure 49. The comparison of extracted networks with the ground truth data: (a) The ground truth data for Figure 47(a). (b) The extracted drainage network (dark) with ridge lines (light).

intersect), isolated stream segments, and other ambiguities. Figure 48(c) shows the valley segment map after applying the A-map masked thinning algorithm. All the segments in this map satisfy the conditions:

1) All segments have a width of one pixel.

2) All segments mark V-shaped topographic features.

3) No segments cross ridge lines.

However, this map also contains inconsistent segments. An incomplete drainage parameterized directed graph is constructed from the this map by the network tracing and property measuring algorithm described in Section 7.3.1.

The statistics of this incomplete PDG are as following:

Number of drainage trees: 26
Number of drainage trees not terminated on the boundaries: 11
Number of stream branches: 106
Number of inconsistent branches: 3
Number of inconsistent nodes: 92 .

DNESYS then performed the spatial reasoning on the incomplete drainage graph. All inconsistent nodes and branches were examined and removed, and a complete drainage *PDG* was obtained. The resulting map produced by the high-level expert is shown in Figure 48(d).

The statistics for the resulting PDG are

Number of drainage trees: 10

Number of trees not terminated on boundaries: 0

Number of branches: 100

Number of inconsistent nodes: 0

Number of inconsistent branches: 0

Number of curves generated: 15 .

This result shows considerable improvement compared with the result obtained from low-level labeling shown in Figure 48(a). The only difficulty is a segment A that is introduced by a noise point. It will be shown later that this is removed when ridges are taken into account. The number of drainage trees which do not terminate on the boundaries are reduced from 11 to 0, and all nodes are consistent. The drainage network that is generated is consistent with the model of drainage networks described in Section 7.4.2 and therefore has the structure of a real drainage network. In particular, it is also comparable to the ground truth data in Figure 49(a). This ground truth was interpreted manually from the blue line drainage network of a USGS topographic map with a much finer resolution. The extracted drainage networks with the ridge lines are shown together in Figure 49(b).

The same improvements are obtained from the second set of test data. A 175 x 186 pixel region from the northeast quarter of the Hamburg, VA. quadrangle was tested using DNESYS (Figure 50(a)). Topographic features plotted from this DEM data are shown in Figure 50(b).

The same low-level processing operations and the same inference procedures were applied to these test data. The parameters in the knowledge base, however, were slightly modified according to the geographic features in this area. The valley segment map that resulted from the low-level processing is shown in Figure 51(a). An initial PDG was deduced from this map by DNESYS.

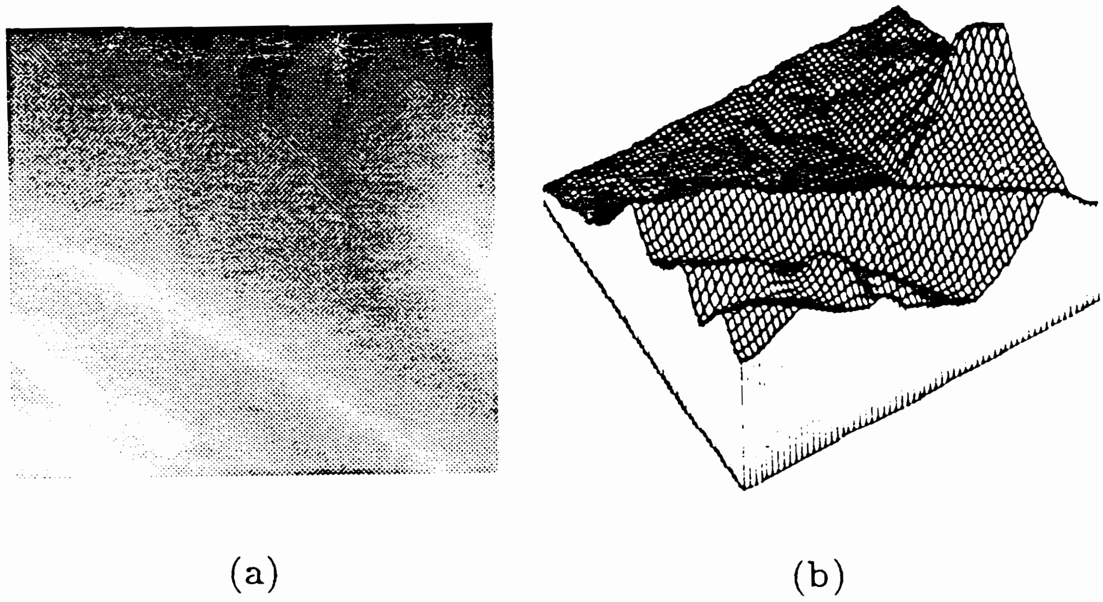
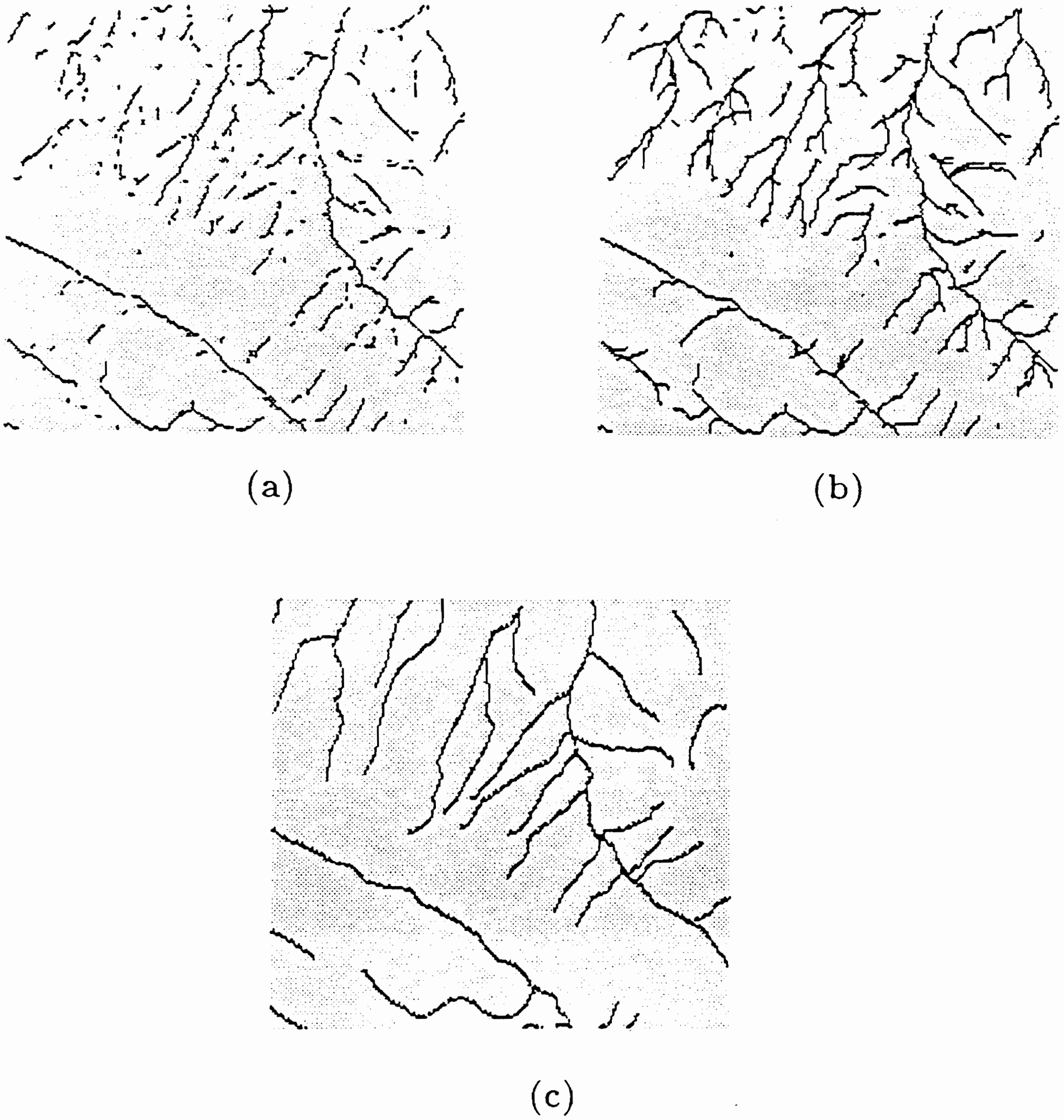


Figure 50. (a) The DEM of the second test area (size: 175 × 186). (b) The surface plotted from the DEM in (a).



(a)

(b)

(c)

Figure 51. (a) The valley segment map of the second test area. (b) The corresponding reasoning result. (c) The ground truth data for Figure 50(a)).

The statistics for the initial PDG are:

Number of drainage trees: 198

Number of drainage trees not terminated on boundaries: 180

Number of source nodes: 233

Number of inconsistent nodes: 168

Number of inconsistent branches: 21.

The statistics for the resulting PDG are:

Number of drainage trees: 24

Number of drainage trees not terminated on boundaries: 8

Number of source nodes: 124

Number of inconsistent nodes: 8

Number of inconsistent branches: 3.

The map corresponding to the resulting PDG is shown in Figure 51(b). From the comparison of the statistics of PDGs and the comparison of the extracted drainage map with the ground truth data (Figure 51(c)), we see that DNESYS also works well for the finer resolution data. The number of drainage trees which do not terminate on the boundaries are reduced from 180 to 8. The number of inconsistent nodes are reduced from 168 to 8. The number of inconsistent branches are reduced from 21 to 3. The interpreted network shows more detail than the ground truth network derived from the USGS blue lines. Although these interpretations differ from the ground truth data, we do not consider them to be "errors" because they show more detail than the ground truth data, and the additional detail appears to depict true drainage.

A third area of 160 × 150 pixels from the northwest quarter of the Hamburg quadrangle was tested on DNESYS (Figure 52(a)). The corresponding topographic features are shown in Figure 52(b). The same knowledge base as in the previous test was used. The experimental results are comparable to the results for the first DEM.

The statistics of the initial PDG are:

Number of drainage trees: 168

Number of drainage trees not terminated on boundaries: 153

Number of source nodes: 189

Number of inconsistent nodes: 148

Number of inconsistent branches: 16.

The valley segment map that corresponds to the initial PDG is shown in Figure 53(a).

The statistics of the resulting PDG are:

Number of drainage trees: 19

Number of drainage trees not terminated on boundaries: 6

Number of source nodes: 84

Number of inconsistent nodes: 6

Number of inconsistent branches: 1.

The drainage network after reasoning by DNESYS is shown in Figure 53(b). The ground truth data for this map is shown in Figure 53(c). The experimental results again demonstrate substantial improvement. The number of drainage trees which do not terminate on boundaries are reduced from 153 to 6. The number of

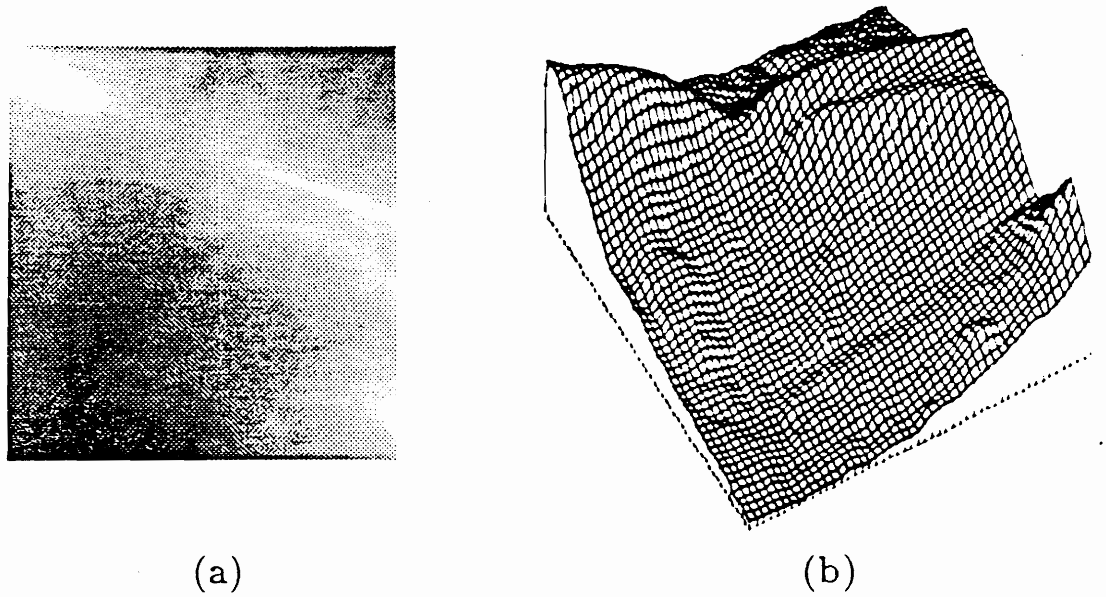


Figure 52. (a) The DEM of the third test area (size 160 × 150). (b)The surface plotted from the DEM in (a).

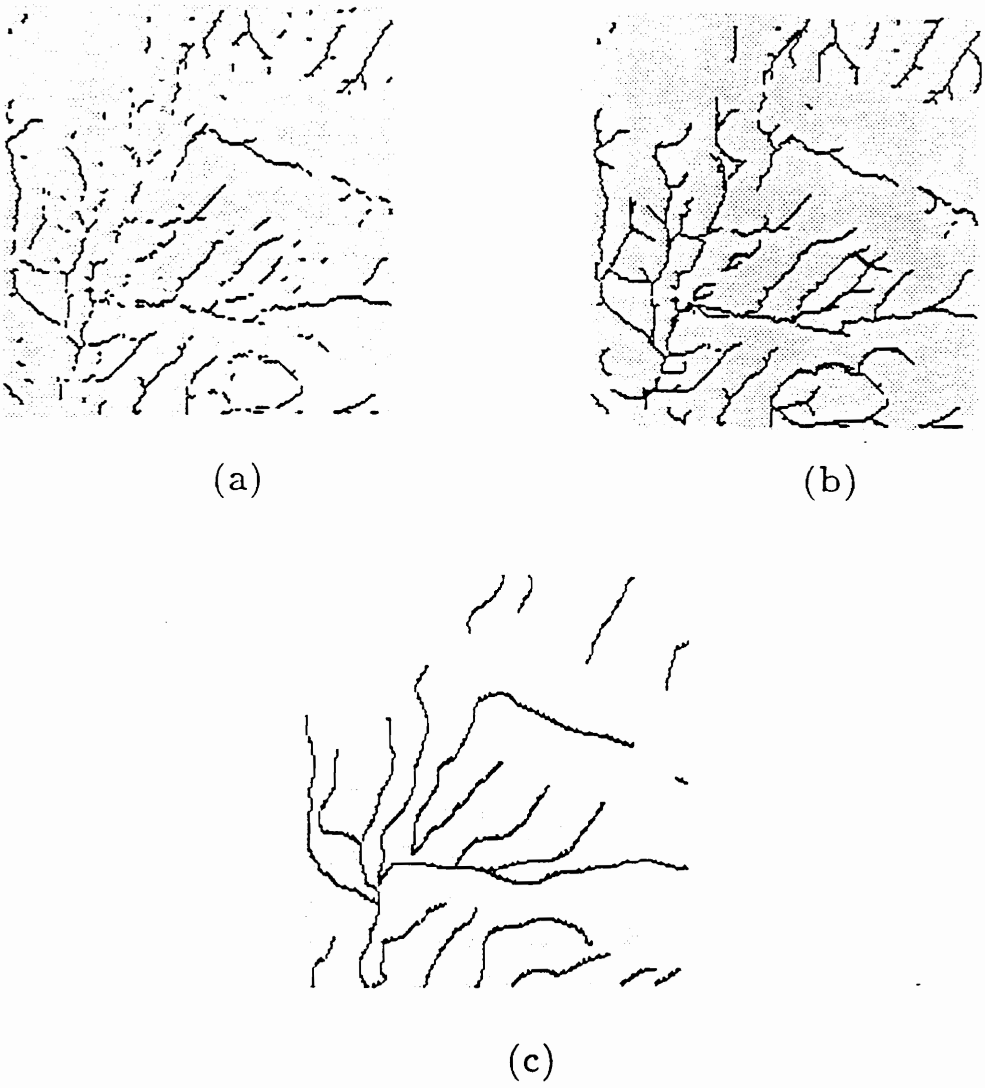


Figure 53. (a) The valley segment map of the third test area. (b) The corresponding reasoning result. (c) The ground truth data for Figure 52(a).

inconsistent nodes are reduced from 148 to 6. The number of inconsistent branches are reduced from 16 to 1, a very significant improvement.

7.4.7.2 Comparison with Earlier Methods

In order to compare these results with other local operators, we have run Jenson's operator on the elevation data for the second test area shown in Figure 50(a). The results are shown in Figures 54(a) and 54(b). Our experience is that it is difficult to extract the drainage networks from the result of Jenson's operator. In both figures, most drainage pixels labeled by the operator are connected together to form dense regions within which the stream segments are difficult to identify. Possible reasons for that are

1. Jenson's operator is sensitive to the data resolution.
2. Due to noise, quantization errors, and insufficient accuracy in the raw data, some drainage features are not directly supported by the local features of the raw data.

7.4.7.3 Results after Verification and Reinterpretation

It has been shown that high level reasoning significantly reduces the errors inherent in low level operators. The errors remaining after high level reasoning are relatively minor and may be categorized by comparing the extracted drainage networks with ground truth data. These errors include:

1. A few isolated segments, which are counted as trees which do not terminate on boundaries, are not connected to the overall system. These errors often occur

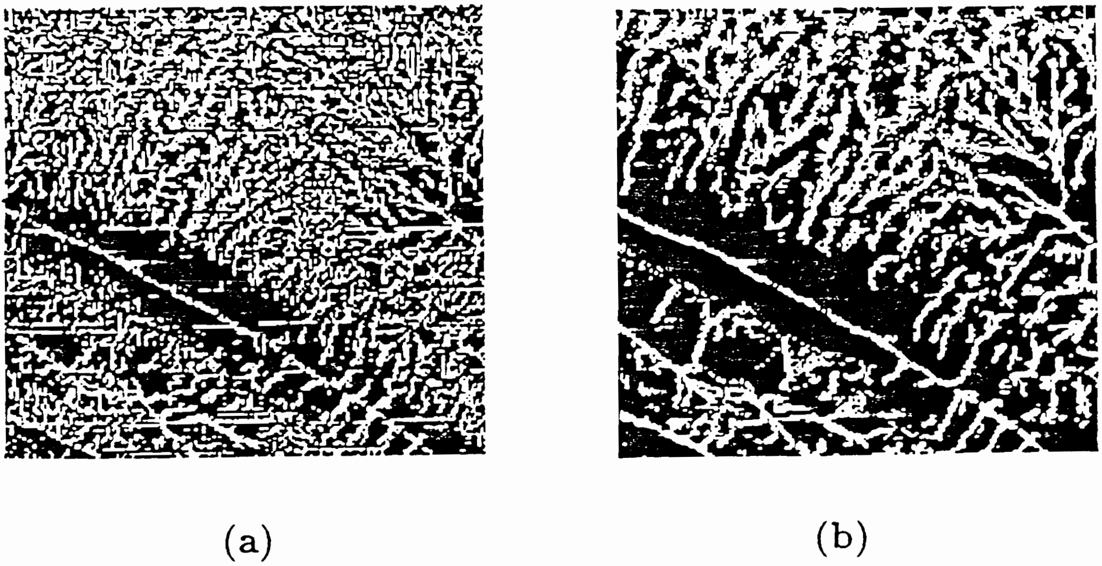


Figure 54. (a)The drainage cells labeled by Jenson's operator without filtering. Black: drainage cells; white: others. (b) The drainage cells labeled by Jenson's operator with Gaussian filter. Black: drainage cells; white: others.

in regions of low topographic slope where it is difficult to determine the correct drainage direction due to the lack of sufficient evidence.

2. A few true stream segments incorrectly connected to one another. These are counted as inconsistent branches.

3. A few interpreted segments that intersect ridge points. This is caused by the curve fitting algorithm explained in Section 7.4.5. However, the ridge points from the low-level labeling, especially the isolated one, are not always reliable.

The results can be further improved by adding the verification stage discussed in Section 7.4.6 and then making a second pass through the high level reasoning stage of DNESYS. The verification stage produces a new set of valley segments that replaces those from the V-shaped map that were used in the first pass.

To demonstrate the effectiveness of verification, the upper left portion of the second test area of Figure 50(a) was processed further. The results are shown in Figure 55. The reinterpreted drainage networks are more accurate than the hypothesized ones and reveal more detail than the digital ground truth data in Figure 55(c). However, by detailed examination of corresponding topographic maps, almost all the additional details are confirmed to be extensions of stream segments into the valleys at their headwaters. To check this, Figure 55(d) shows the extensions of the blue line stream map into the valleys, and the match is extremely good.

7.4.7.4 Drainage Basin Labeling

Earlier it was asserted that the drainage network was fundamental to the determination of the drainage basins. To confirm this a simple algorithm was tested for

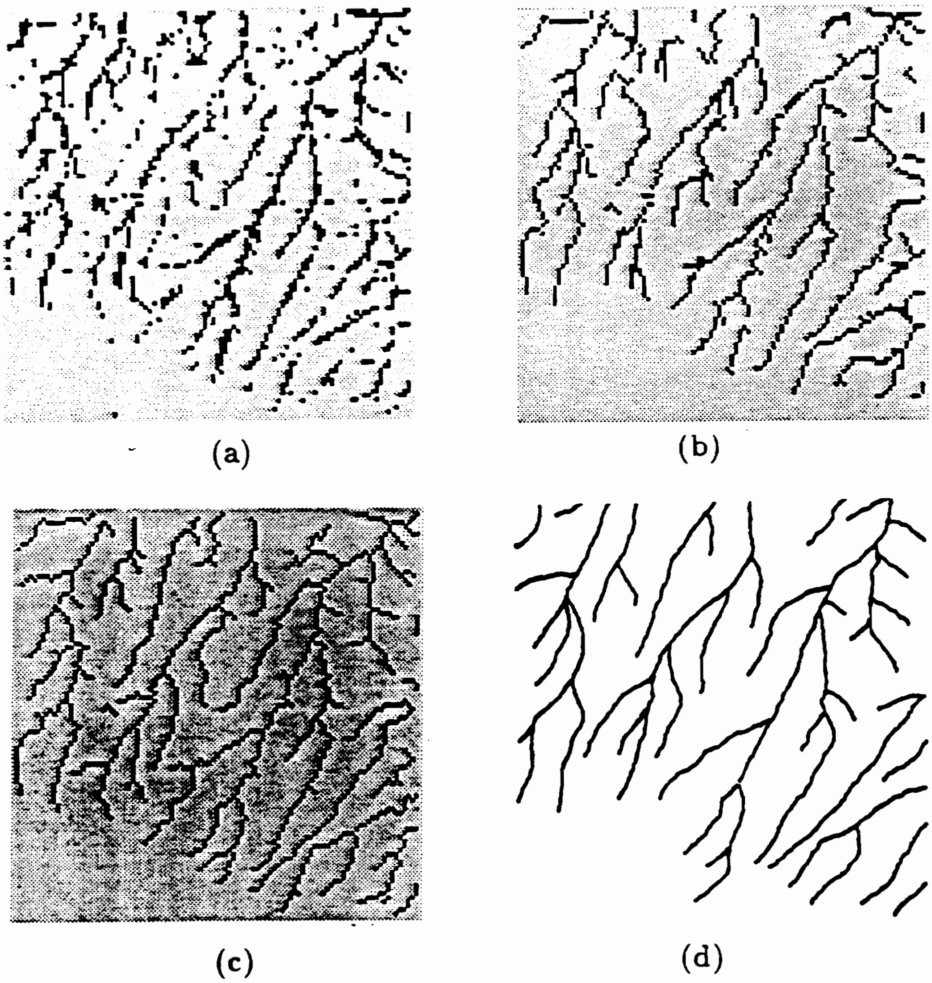


Figure 55. (a) The low-level labeling. (b) The high-level hypotheses. (c) The networks after verification and reinterpretation. (d) The ground truth map.

labeling the drainage basins using the network from Figure 55(c). The algorithm is a drainage tree thickening algorithm that is used to grow the drainage basins outward from the streams while maintaining the maximum possible consistency with the ridge line fragments that are located between the branches of the drainage trees. Basin labels are propagated from labeled pixels to adjacent pixels with higher elevations. The basins for the streams leaving the image area are shown in different shades in Figure 56. Also, the system is able to label the extracted drainage netwroks according to the hierarchical order model. This result is shown in Figure 57.

7.5 Object Recognition on a Printed Circuit Board

The task of recognizing electrical parts on a printed circuit board (PCB) is an important problem in automatic industrial inspection. The experiments presented in this section do not attempt to solve the entire problem of automatic PCB inspection. By means of this experiment, we intend to illustrate the sources of uncertain information, show how a hierarchical visual event space is constructed, show how visual evidence is collected and mapped from one level to the other level, and show how our reasoning algorithm can be used to cope with uncertain visual information. This experiment also shows that an image which looks so clear and so simple for human vision system still poses quite a complicated task for a computer vision system.

A test PCB image is shown in Figure 58. There are IC chips, resistors, capacitors, and other electronic parts on the board. From Figure 58 we can clearly see that all electronic parts have different appearances because of illumination, surface,

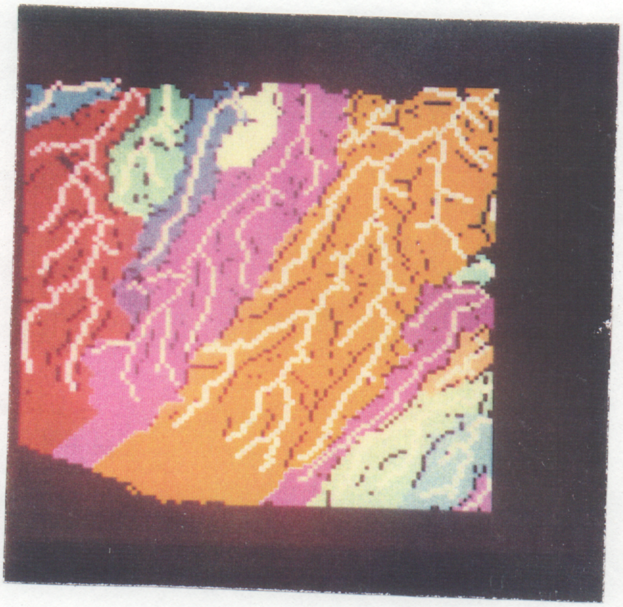


Figure 56. Drainage basins with drainage network (white) and ridge line fragments (black).

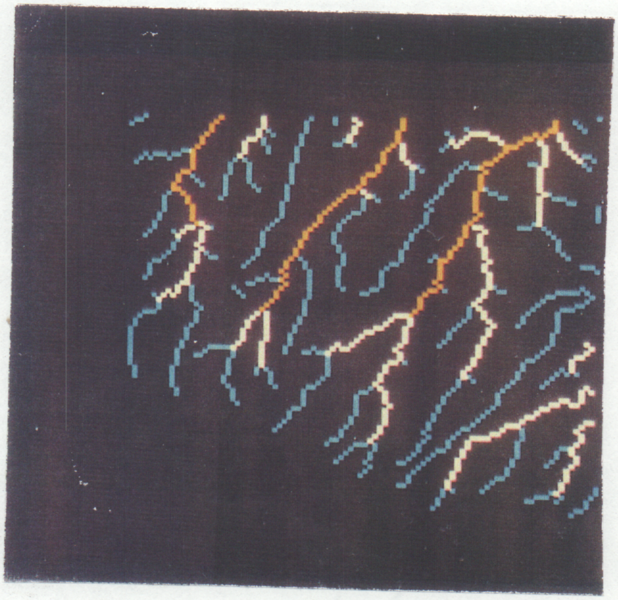


Figure 57. The drainage network labeled according to the hierarchical order. Blue: the first order. White: the second order. Yellow: the third order.

and shape variations. For example, the dark marks on the resistors break the resistor bodies into separate parts. Without *a priori* knowledge about a resistor, one can not even recognize that these separate parts are connected each other. Also, the highlights on surfaces of capacitors and surfaces of resistors make the same type of part look different at different locations. This raw data ensures the low-level processing modules provide uncertain information to the higher processing levels, whatever low level processing modules are chosen.

The hierarchical visual event space for this problem may consist of the significant points, edges or lines, contours or regions, primary components, electronic parts, and the PCB board. Since we have already shown how to reason from raw data to edges and from edges to significant lines in Section 7.2 and Section 7.4 of this chapter, we will focus on the reasoning process from region level to the object level in this section.

The results from the previous levels are shown in Figure 59. Now we are at the level of reasoning about the primary components of objects based on region-level evidence. For simplicity, a subimage of Figure 58, which contains an array of resistors, is used for this experiment (Figure 60). The corresponding lower level processing results are shown in Figure 61. To label each region with a unique label is a necessary step for the evidence measuring and reasoning at the high level. The symbolic image which is labeled by each region with a unique label is shown in Figure 62. Notice that the regions in Figure 61 do not correspond to the shapes of objects to be recognized. The visual events about which reasoning takes place at this level involve components of resistors. A frame of discernment for components of resistors is shown in Figure 63, and this corresponds to a single reasoning tree

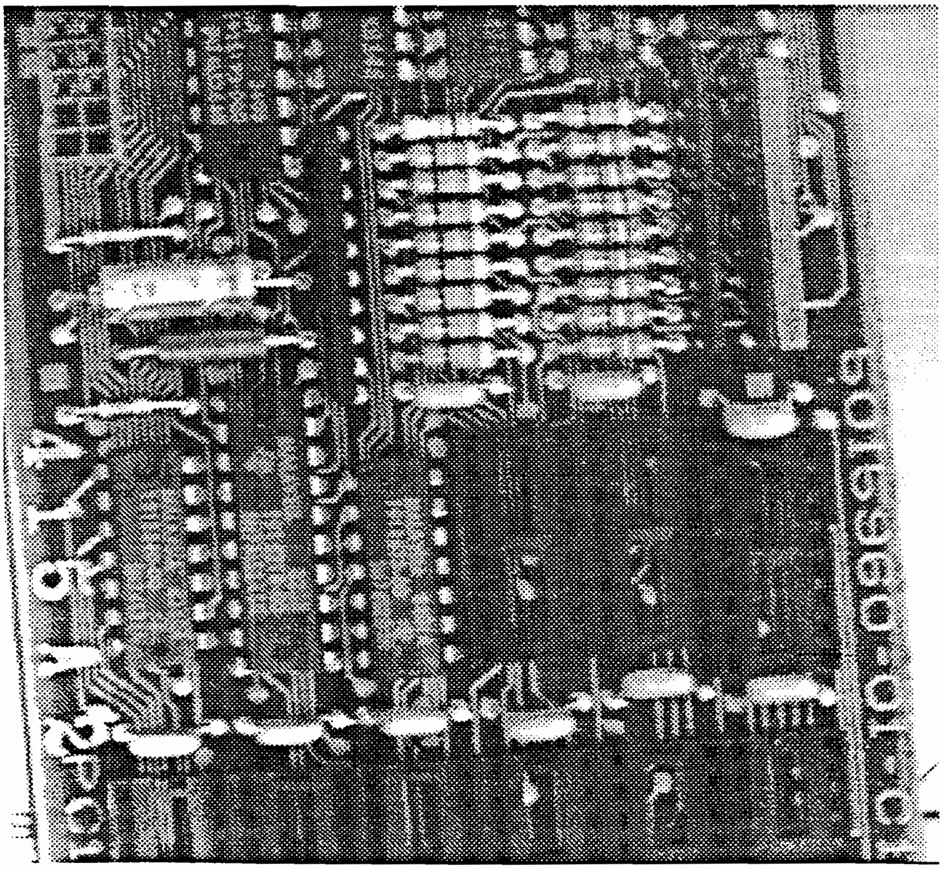


Figure 58. The PCB test image, size: 512 × 480.

in Figure 25. There may be more reasoning trees as in Figure 25 if there are other object types besides resistors to be recognized in the image.

Given a visual event at the current reasoning level, the focus of attention function returns the names of the levels to be analyzed: the level of components of objects, the region level, and the raw data level. The focus of attention function also returns the search distance: 1.5 times the maximum component size. Then, the focus of attention function selects an index element of a visual event from the invertible index table at the level the region level. The control mechanism described in Chapter VI is used to generate hypotheses about the event.

Visual evidence at the region level and the raw data level which is available for supporting visual events at the component level in this example are the region size, mean gray value, variance of graytone value, row center of mass, column center of mass, angle, mean radius, variance of radius, length of the region boundary, measure of circularity, etc. The measurements of these pieces of evidence are shown in Figure 64-75. From these figures we can see that no piece of evidence by itself is sufficient for determining the component type because of the nature of uncertainty information. However, each piece of evidence provides a grain of support for determining the type of component, which can be combined into more certain support for hypotheses about visual events at this level. Note that some evidence should be reextracted from the original raw data under the guidance of the events at the region level. The visual evidence extractor computes these visual evidence measurements. Among these pieces of visual evidence, only six are chosen in this experiment. They are region size, boundary length, circularity, center of mass, variance of radius, and mean graytone. Evidence mapping functions map these measurements into evidence strengths. The mapping functions in this experiment are discrete table look-up

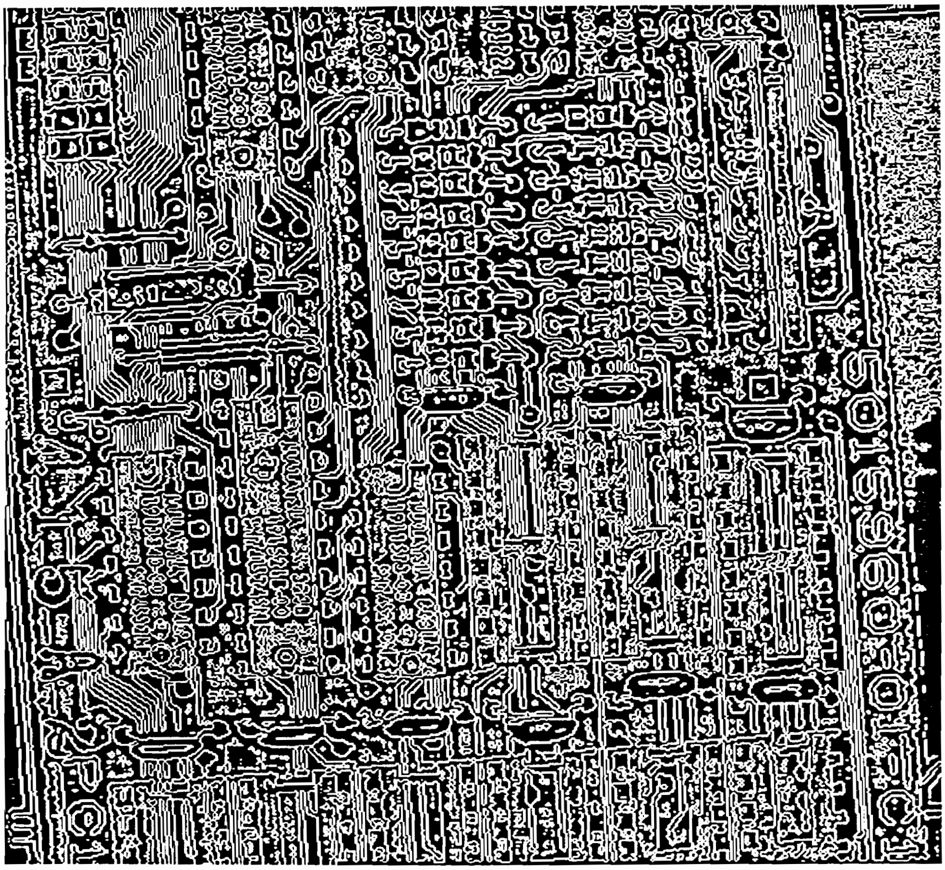


Figure 59. The reasoning results for Figure 58 at lower levels: significant points and significant lines.

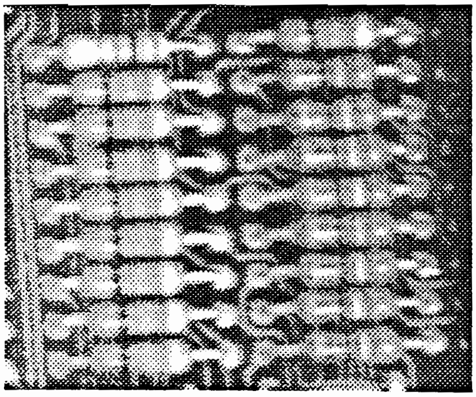


Figure 60. The subimage of Figure 58. Size: 191 × 161.

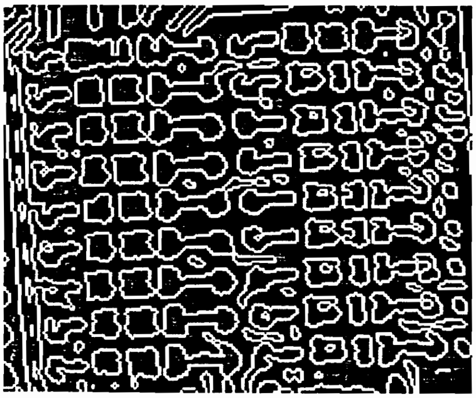


Figure 61. The reasoning results from lower levels at the region level.

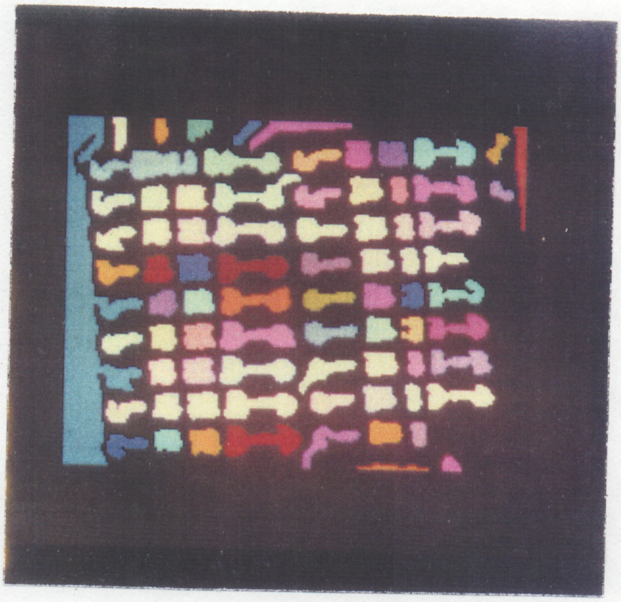


Figure 62. The symbolic image of Figure 61: each region with a unique label.

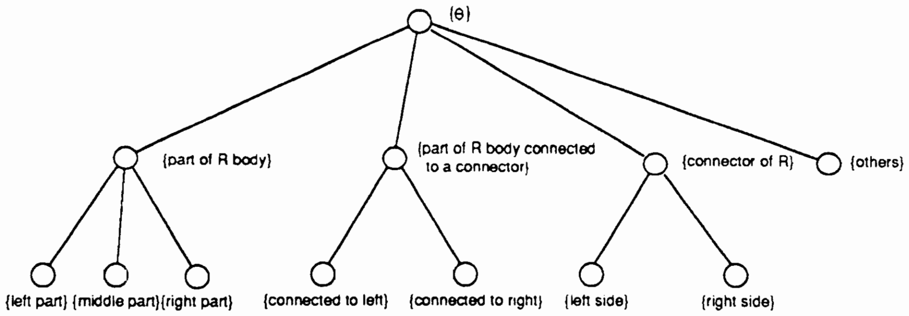


Figure 63. The frame of discernment for components of resistors.

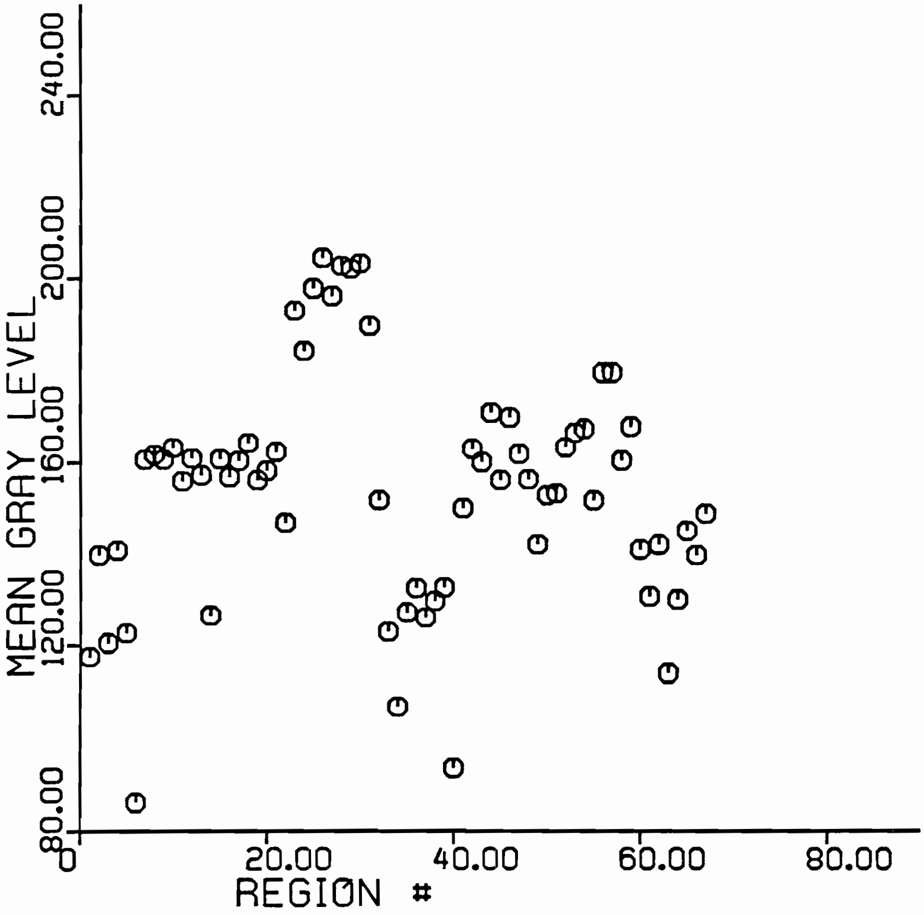


Figure 64. Visual evidence 1: mean graytone.

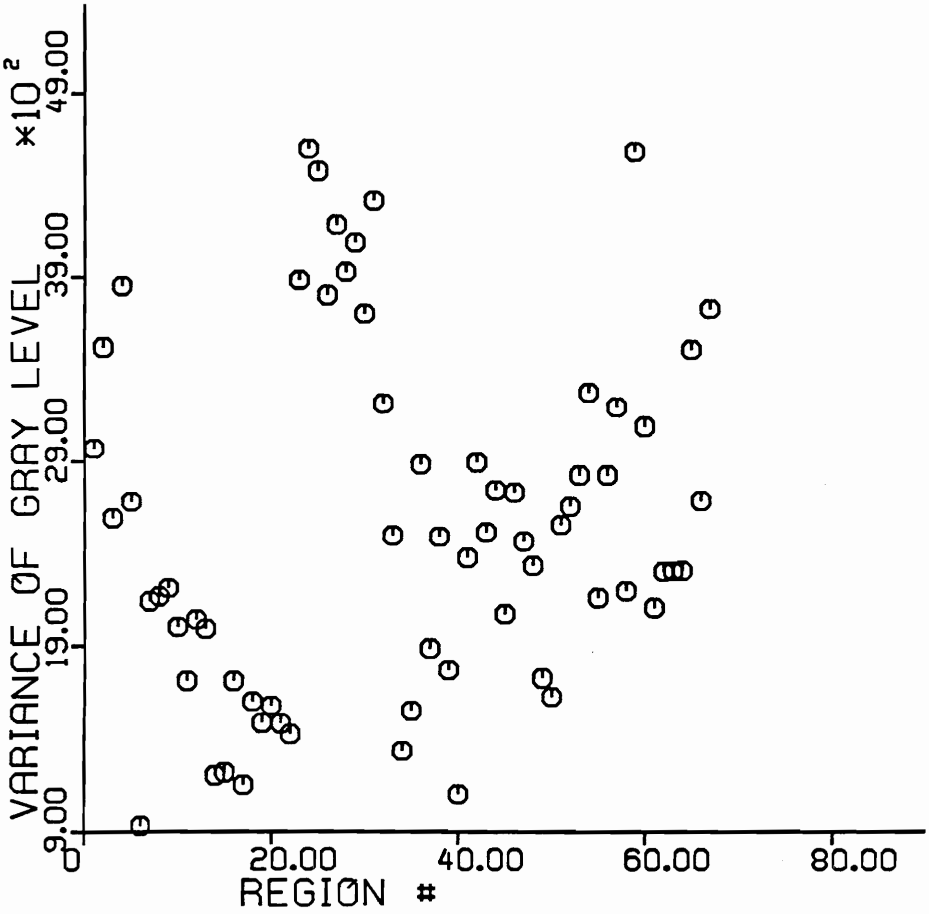


Figure 65. Visual evidence 2: variance of graytone.

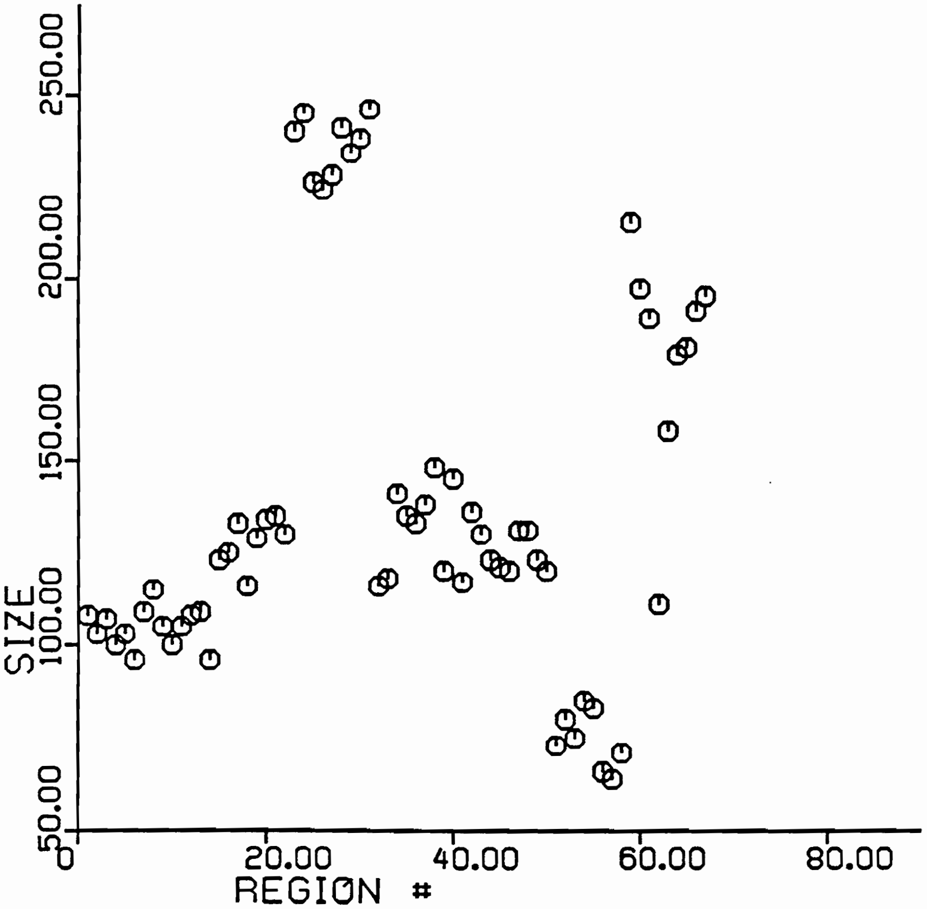


Figure 66. Visual evidence 3: region size.

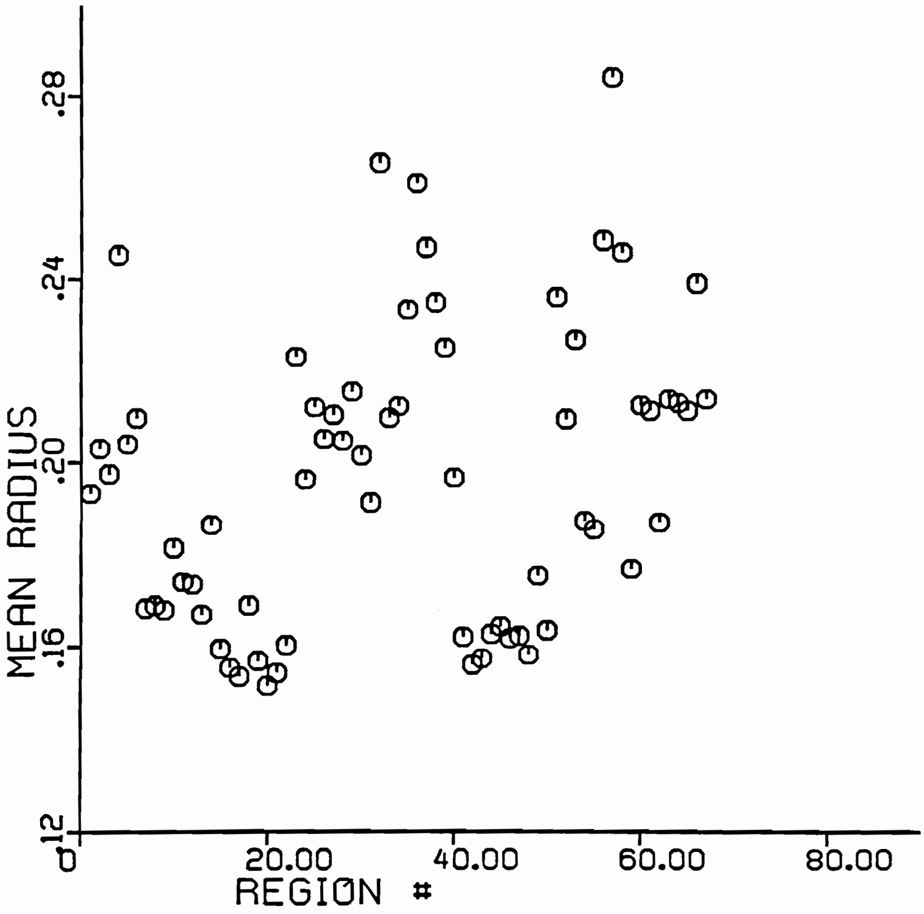


Figure 67. Visual evidence 4: mean radius.

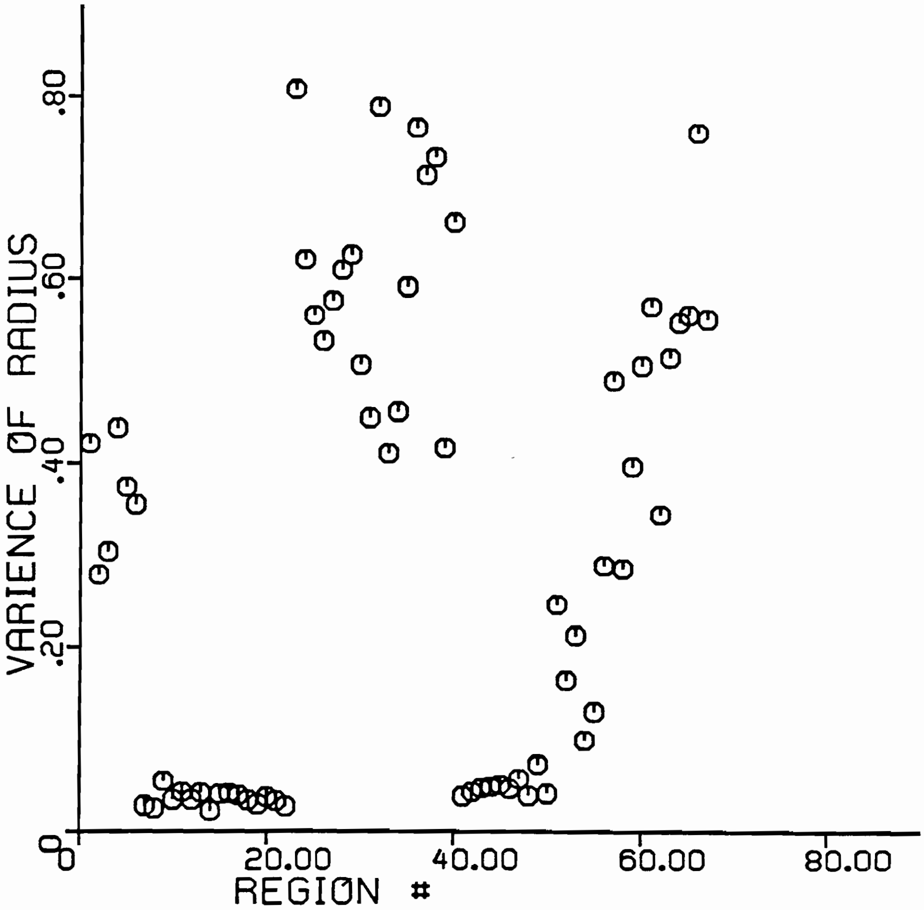


Figure 68. Visual evidence 5: variance of radius.

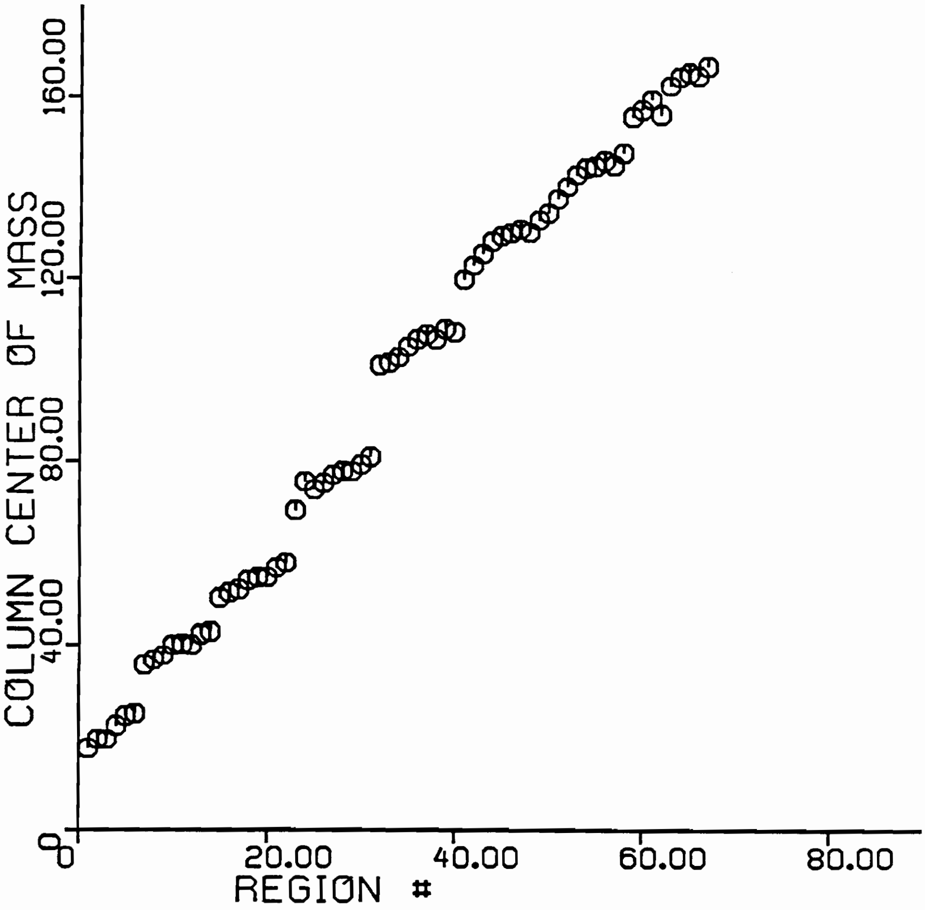


Figure 69. Visual evidence 6: column center of mass.

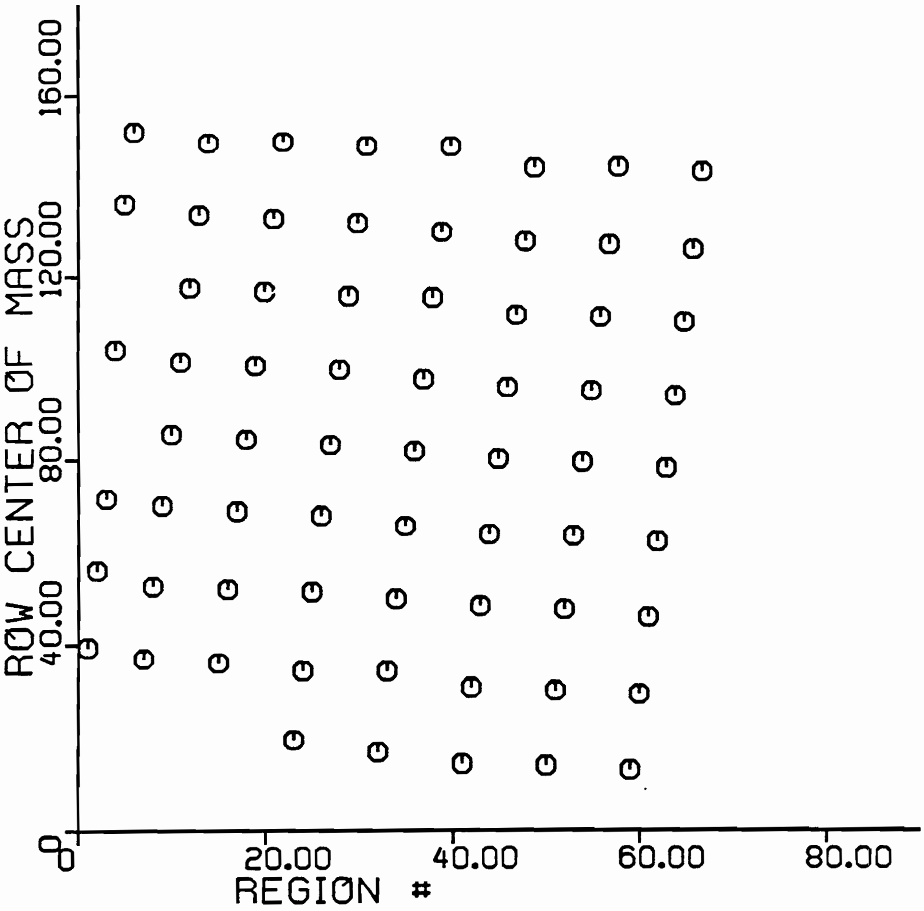


Figure 70. Visual evidence 7: row center of mass.

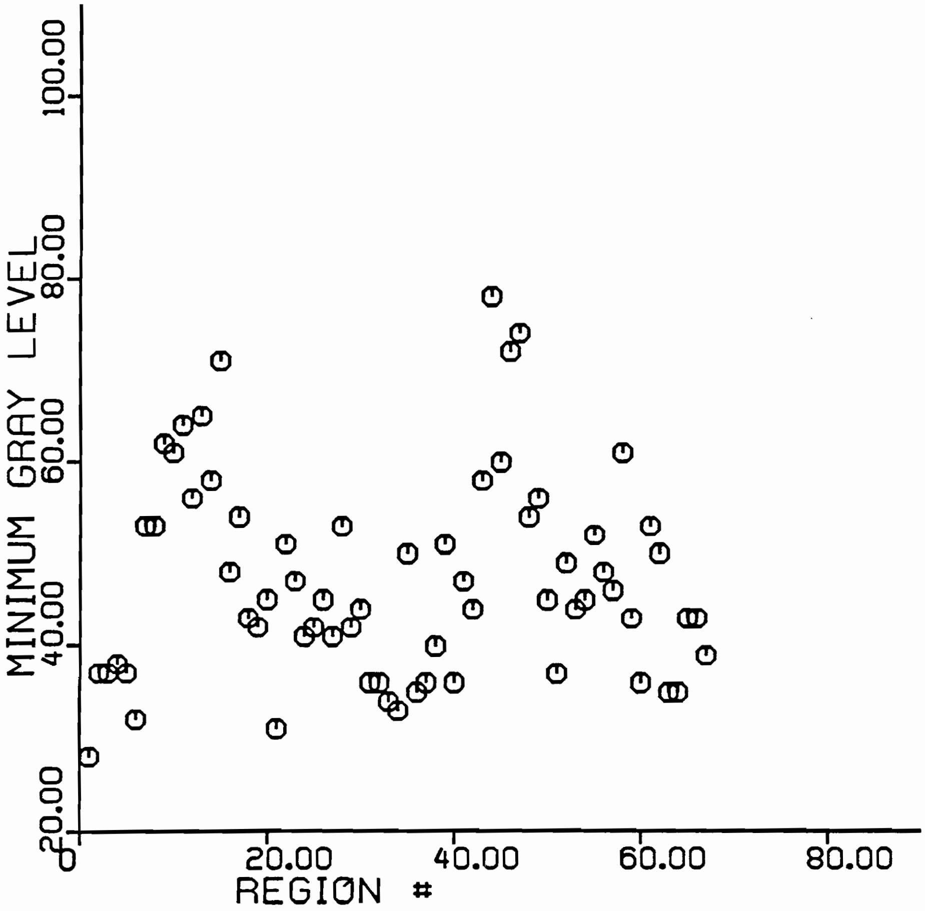


Figure 71. Visual evidence 8: minimum graytone.

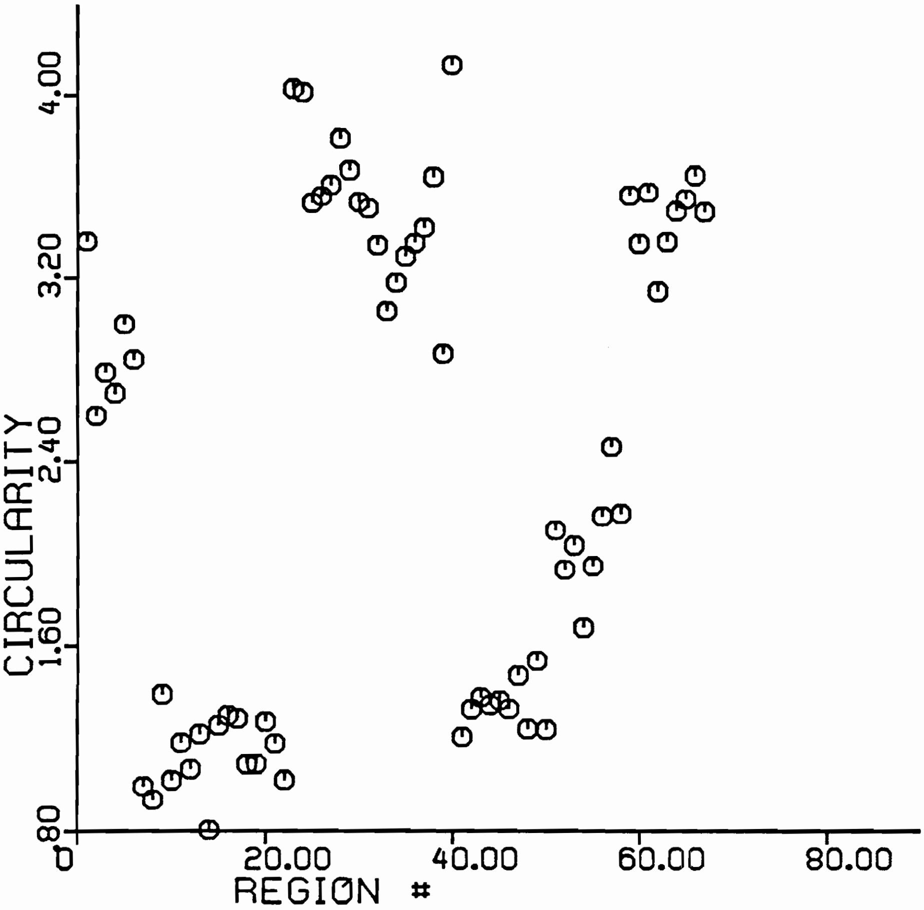


Figure 72. Visual evidence 9: circularity.

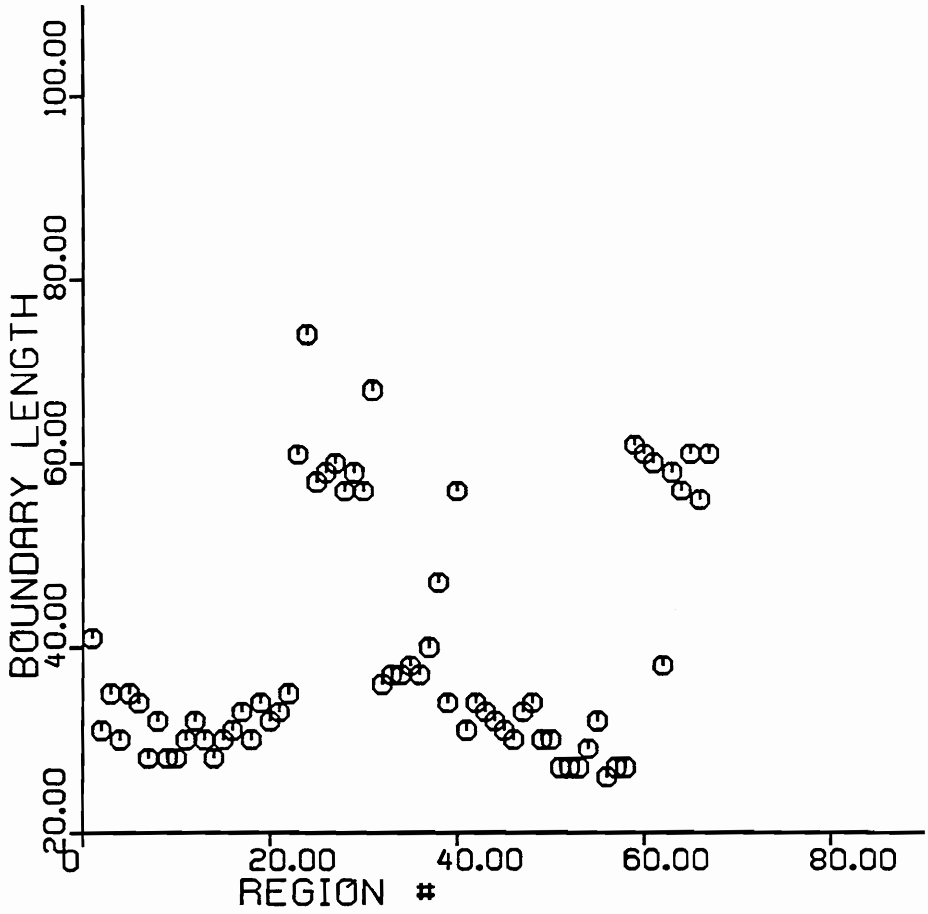


Figure 73. Visual evidence 10: boundary length.

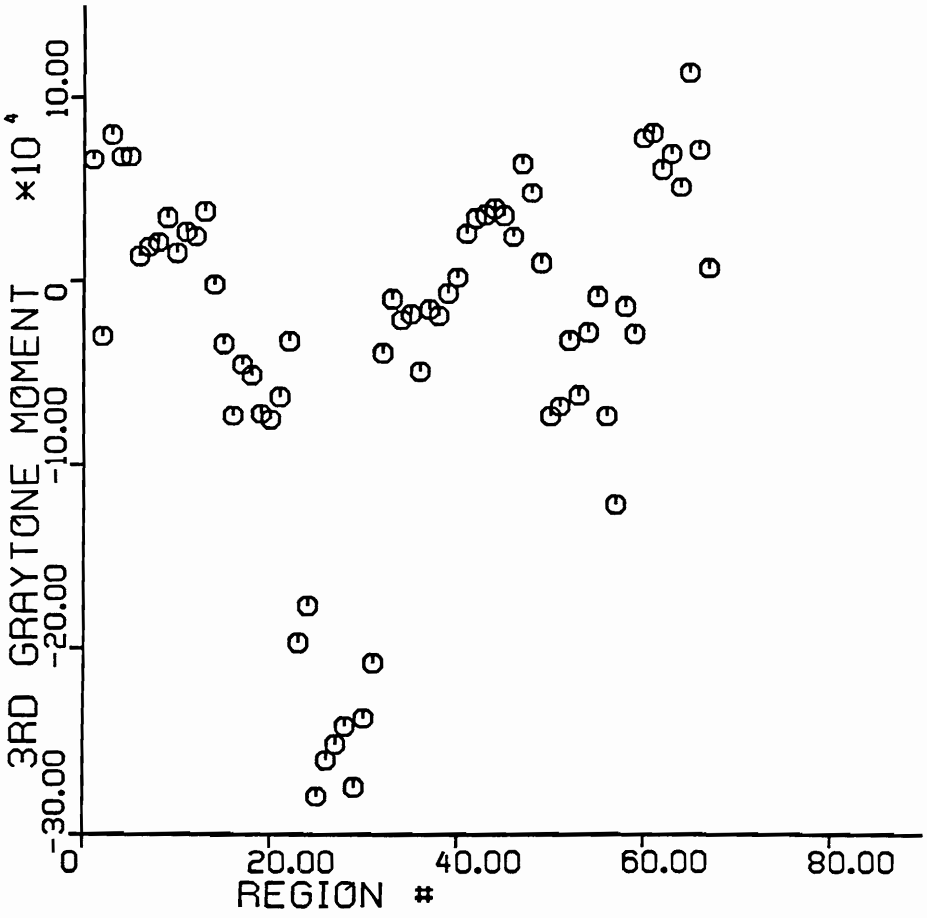


Figure 74. Visual evidence 11: the third graytone moment.

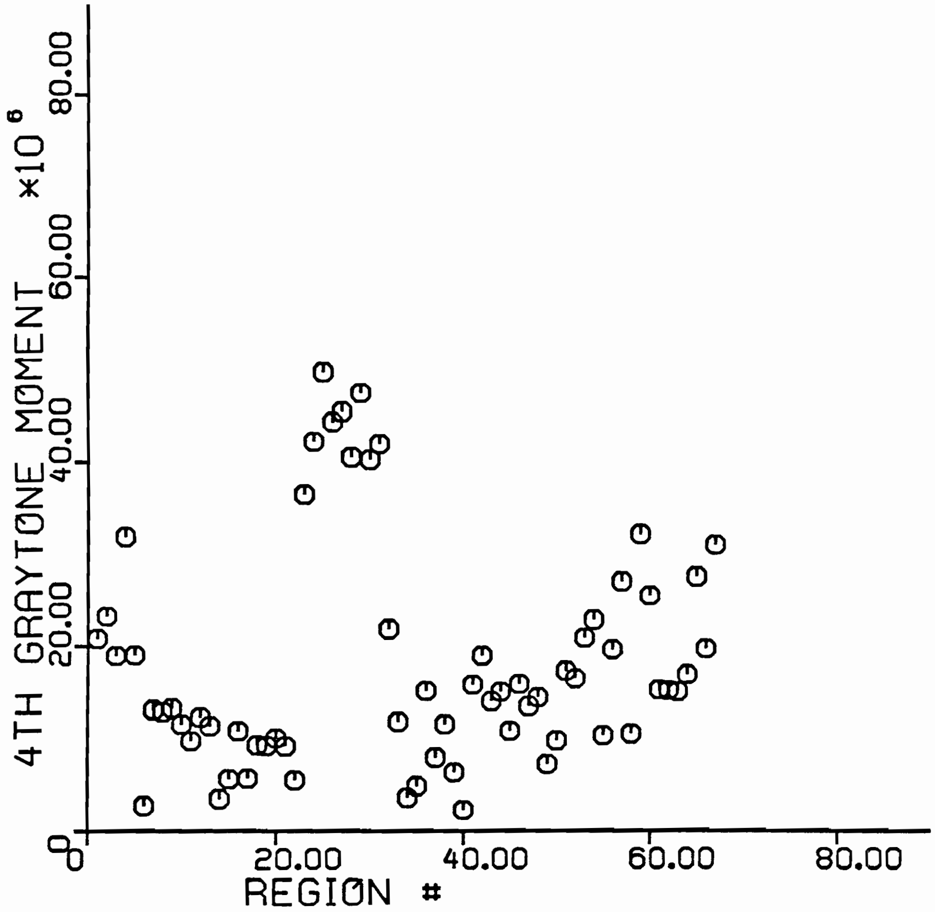


Figure 75. Visual evidence 12: the fourth graytone moment.

functions. The initial beliefs which are exactly committed to each hypothesis are computed. These initial beliefs are propagated into the built-in reasoning tree in Figure 63. Then, the interactions among pieces of hierarchical evidence can be determined to deduce the type of component. For this step, the algorithm is described in Section 5.2 of Chapter V, and the implementation method is given in Section 6.4 of Chapter VI. The some of the dichotomy belief values computed at this level are shown in Figure 76. The reasoning results at this level are shown in Figure 77. The labeling accuracy is 91.1 percent in a total of 67 regions.

After the completion of the reasoning process at the region level, all asserted components of resistors and the spatial relations among them are mapped into evidence strengths at the object level for deducing electronic parts at that level. The same reasoning steps as at the region level may be applied at this level. However, the visual evidence in this level is mainly from spatial relation measurements. The mapping procedures and reasoning steps for this stage are similar to the algorithm described in Section 5.3 of Chapter V. As soon as a visual event is determined to be a resistor, the detailed description of that resistor will be propagated downward to check the consistency of related visual events at the level of components of resistor. Meanwhile, missing information (easily seen in Figure 61) which results from low-level processing alone is filled in. This completes the verification process for the hypothesis about a resistor at the object level.

Region number	Hypotheses Hi	Beliefs for Hi	Beliefs against Hi	Uncommitted Beliefs	Labeling result
12	1	0.8074811	0.1599170	3.2601926E-02	correct
	2	0.2379085	0.6303473	0.1317443	
	3	0.2070734	0.6710824	0.1218443	
	4	0.8720990	0.1046946	2.3206346E-02	
13	1	7.6894283E-02	0.7989886	0.1241171	correct
	2	0.1874637	0.6896297	0.1229067	
	3	9.4748810E-02	0.7987493	0.1065019	
	4	0.9999184	5.1616193E-05	2.9979588E-05	
21	1	8.4777720E-02	0.8097447	0.1054775	correct
	2	0.9331138	4.2311180E-02	2.4575077E-02	
	3	0.9682848	2.1357317E-02	1.0357952E-02	
	4	1.7272513E-02	0.9345738	4.8153672E-02	
22	1	0.1495024	0.7524795	9.8018125E-02	correct
	2	0.9948712	2.9751726E-03	2.1535861E-03	
	3	0.5609675	0.3414514	9.7581111E-02	
	4	2.3059983E-02	0.9099219	6.7018084E-02	
30	1	8.4777720E-02	0.8097447	0.1054775	correct
	2	0.9945288	2.8866432E-03	2.5845449E-03	
	3	0.6870073	0.2334875	7.9505138E-02	
	4	1.6580667E-02	0.9352317	4.8187569E-02	
31	1	0.9956045	3.1194342E-03	1.2760167E-03	correct
	2	0.3245045	0.5587213	0.1167743	
	3	0.1118111	0.7286139	0.1595751	
	4	0.1879095	0.7102218	0.1018686	
32	1	0.9452386	4.4163227E-02	1.0598332E-02	correct
	2	0.2407435	0.6280024	0.1312542	
	3	0.2070734	0.6710824	0.1218443	
	4	0.7994734	0.1681857	3.2340933E-02	
34	1	8.4777720E-02	0.8097447	0.1054775	correct
	2	0.9331138	4.2311180E-02	2.4575077E-02	
	3	0.6870073	0.2334875	7.9505138E-02	
	4	0.4201215	0.5433462	3.6532320E-02	
35	1	0.9415534	4.3600257E-02	1.4846380E-02	correct
	2	0.1163604	0.7499776	0.1336620	
	3	6.6658020E-02	0.8257762	0.1075658	
	4	0.8866028	9.2822351E-02	2.0574773E-02	
54	1	0.9836831	1.3158903E-02	3.1578857E-03	correct
	2	0.4797946	0.4174294	0.1027759	
	3	0.3052896	0.5282917	0.1664187	
	4	0.1573883	0.7369146	0.1056972	
58	1	0.9009570	7.3884413E-02	2.5158476E-02	incorrect
	2	0.1163604	0.7499776	0.1336620	
	3	6.6658020E-02	0.8257762	0.1075658	
	4	0.9405419	4.8670135E-02	1.0788103E-02	

Figure 76. Belief values calculated for hypotheses at the region level. Hypotheses: 1: left connector; 2: left part of the body; 3: middle part of the body; 4: right part of the body with the righ connector.

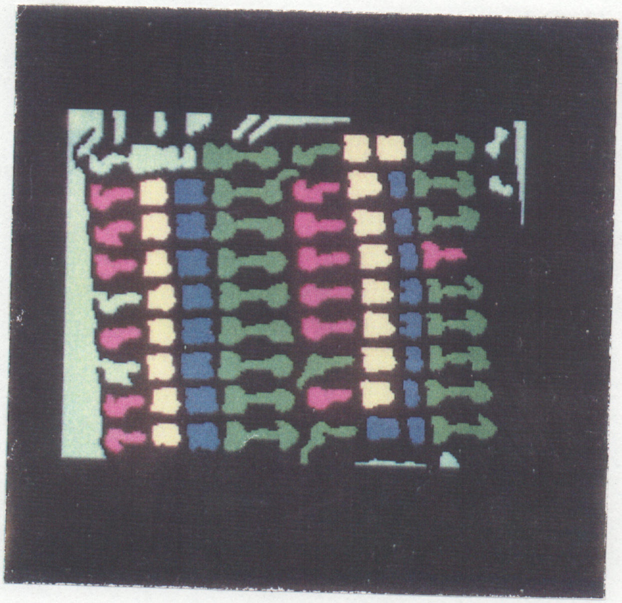


Figure 77. The reasoning results at the component level. Red: the left connector. White: the left part of the resistor body. Purple: the middle part of the resistor body. Green: the right part of the resistor body with the right connector. Light blue: others.

Chapter VIII
SOME ASPECTS OF DESIGNING THE SYSTEM

The principles and methods for designing an uncertainty reasoning system within our framework have been already described in previous chapters. However, some of the practical aspects of designing such a system should be emphasized further.

General speaking, designing a system depends upon the type of task to be solved, the kind of knowledge about images to be used, and the types of processing modules available for extracting visual evidence. Designing such an uncertainty reasoning system, in fact, involves mainly designing three visual hierarchies: the visual event hierarchy, the visual hypothesis hierarchy, and the visual evidence hierarchy with an associated set of mapping functions.

A hierarchical visual event space should be constructed first. Then a visual hypothesis space is constructed corresponding to the visual event hierarchy. Based upon above two hierarchical spaces, the visual evidence hierarchy is constructed together with the mapping functions that determine the *bpa's*.

8.1 Designing the Visual Event Hierarchy

The visual event space should be hierarchically constructed from the level of image primitives to a high-level interpretation. The number of levels in the event hierarchy depends upon the final processing goal. For the resistor problem, if the final processing goal is to recognize different PCB pictures, then six levels of visual event hierarchy may be needed (see Figure 78). If the final processing goal is to recognize electronic parts, then only five levels of visual event hierarchy may be required, and the top level will consist of the visual events about different types of electronic parts. Since each additional level of visual hierarchy requires considerable computation including evidence mapping, evidence combining, evidence interaction, hypothesis selection etc., too many levels of representation would considerably increase the computational complexity. However, too few levels of representation would make it difficult to make a reasonable model of a structured world. For example, if we try to eliminate level 1 through 4 in Figure 78, then much useful structural information would not be represented, and this certainly reduces the system ability. A compromise can be made by balancing the computational complexity and the efficiency of representation. In any case, the deep structure of the event hierarchy sould correspond to meaningful structures in the visual image.

A visual event at a particular level of abstraction should provide as complete a description as possible of the visual events at preceding levels. It should provide as much information as possible for the next higher level representation. Usually the initial levels are constrained by the types of primitives that can be computed directly from a sensory image. At the lowest level, the size of a visual event is the pixel size. The number of events may be very large, and the number of classes of events is usually relatively small. The intermediate levels are constrained by what is available from preceding levels and what is required by succeeding levels. The visual events at these levels are transformed from the low level by grouping low-level events into more complex entities or by describing the relationships between events of different types. These events are more meaningful and describe the low level events.

The number of events are decreased as one moves up in the hierarchy. The top levels are directly constrained by the final processing goal. The visual events defined at this level depends upon the goals and the knowledge about the image. For example, if you only need to discriminate two types of electronic parts: resistor and capacitor, then only three types of visual events should be defined, that is, resistor, capacitor, and others. However, if you need to recognize an object among fifty possible objects, then all the fifty types of visual events need to be defined. If you have more knowledge about the image so that only ten types of visual events are possible, then the number of visual events defined can be reduced. The fewer the visual events defined, the faster the reasoning speed will be.

8.2 Designing the Visual Hypothesis Hierarchy

A hypothesis that explains a visual event may also be supported or refuted by evidence from other visual events or by other confirmed hypotheses. A confirmed hypothesis at some level of abstraction is a visual event that provides evidence for reasoning about hypotheses at a higher level of abstraction. Therefore, for each level of the visual event hierarchy, there is a corresponding level of the hypothesis hierarchy. Figure 79 shows level 4 and level 5 of the hypothesis hierarchy which correspond to level 4 and level 5 of the event hierarchy.

A finite set of hypotheses $H_i \in H$ at the level i explain a finite set of visual events at that level in the visual event hierarchy. At level 4 in Figure 79, for example, H_i can be all primary components of electronic parts on the printed circuit board: the components of resistor, the components of capacitor, components of IC chip, etc. The finite set of hypotheses H_i at the level i can be decomposed into groups of mutually exclusive subsets, and each group can be considered to be a separate frame of discernment. This is shown in Figure 79 by Θ_1, Θ_2, and Θ_3. For simplicity, only the frame of discernment for the components of resistor is shown in the figure.

Each such separate frame of discernment corresponds to a class of possibly valid visual events at a particular level of the visual event hierarchy. These hierarchically related hypotheses at each level can be constructed as reasoning trees that themselves are conceptual hierarchies. In the tree, all the terminal nodes correspond to a set of possible visual events . Higher nodes correspond to a cluster of visual events among that set. Each cluster corresponds to the union of the terminal nodes below it. For example, in Figure 79, the node "part of a resistor body" is not an elements of Θ_1. This node represents the union of the terminal nodes: left part of the resistor, right part of the resistor, middle part of the resistor. These reasoning trees are used for evidence interaction to deduce a possible visual event which may not be directly supported by available visual evidence.

Designing a visual hypothesis hierarchy, in fact, requires designing a set of suitable reasoning trees at different levels. First, according to the visual event hierarchy, we need to group all possible predefined visual events into a set of clusters at each level. For example, all components of electronic parts in level 4 of the visual event hierarchy are grouped into the cluster of components of resistor, the cluster of components of capacitor, and the cluster of components of IC chip. Then each cluster is organized into a reasoning tree by the method described above. Reasoning about evidence interaction usually needs both bottom-up and top-down processes as described in Section 2 of Chapter V. The computational complexity for reasoning about such interaction is linear in the number of nodes in the reasoning tree (Shafer and Logan, 1987). Given the number of terminal nodes, the deepest reasoning tree would be a binary tree in which each non-terminal node has at most two child nodes. This reasoning tree would require the worst computations. In this case, however, the maximum number of nodes in the tree is only about twice of the number of terminal nodes. For example, if there are 100 terminal nodes, the number of nodes at the higher levels of this tree may be 50, 25, 13, 7, 4, 2. On one hand, we need to use as many indirect evidence sources as possible; that requires a deeper reasoning tree which contains more nodes that represent the super sets of the terminal hypotheses. On the other hand, to speed up the reasoning process, the fewer levels of tree depth the better. Shafer's partitioning method can be used to decompose a large reasoning tree into several small trees (see the spatial relationship example in Section 3 of Chapter V) to achieve a balanced result.

8.3 Designing Visual Evidence Hierarchy and Mapping Functions

A visual event can provide visual evidence for confirming or disconfirming hypotheses about other visual events. However, a piece of visual evidence may not necessarily be a valid visual event defined in the visual event hierarchy. Visual evidence is the information required for asserting other visual events. For example, a spatial relationship between two visual events may provide visual evidence although it is not a visual event defined in the hierarchy. Visual evidence comes from three basic sources: the visual events confirmed from both lower levels or higher levels, the spatial relationship determined among these conformed visual events at the next lower level, and the intensity or geometric measurements from lower levels. The general algorithm for mapping evidence to strength of evidence is provided in Chapter V.

Designing a visual evidence hierarchy is a task of collecting all available information including inaccurate measurements and incomplete samples for the hypotheses at each level of hierarchy. The visual evidence hierarchy for the resistor problem is shown in Figure 80. Note that the design of a visual evidence hierarchy is the third step after the designs of the visual event hierarchy and the visual hypothesis hierarchy. To design a visual evidence hierarchy we need to refer to the visual event hierarchy and visual hypothesis hierarchy first. The procedure is the following:

1. Pick up one level of visual events from the visual event hierarchy,

2. Examine the corresponding level of visual hypothesis hierarchy to see what types of possible evidence is needed to support or to refute these hypotheses.

3. Examine the available information from raw data or other asserted visual events to determine visual evidence extraction methods.

The visual evidence hierarchy shown in Figure 80 has the following property: the visual evidence at each level consists of the statistical and structural measurements of the visual events at the levels below that level. For example, the visual evidence at level 2 consists of edge points, fork points, corner points, edge length, edge curvature and direction etc. which come from the statistical and structural measurements of intensity and spatial relations from the significant points at level 1. The visual evidence at level 4 consists of region size, center of mass, boundary length, mean and variance of radius, mean and variance of intensity, texture etc. which are extracted from the visual events at level 3 and the raw data.

It is important to note that not every piece of visual evidence has the same importance for a hypothesis. The key evidence or trigger evidence provides the information needed for selecting a hypothesis; this is important for designing the visual evidence hierarchy. However, the selection and definition of trigger evidence depends upon available visual knowledge. At level 4 of the visual evidence hierarchy in Figure 80 we may have more evidence sources than we used, for example, the minimum graytone value, the maximum graytone value, the third graytone moment, the fourth graytone value, etc. After analyzing the visual evidence data from a training set, we noticed that they were not so important as other evidence sources. Therefore, only the region size, center of mass, boundary length, circularity, variance of radius, and mean graytone were chosen as the visual evidence for reasoning about visual events at level 5.

Based upon the collection of multiple sources, subjective mapping functions are designed according to image knowledge and training data. Mapping function design is an important step in the design of an uncertainty reasoning system within our framework. Through mapping functions and the D-S model, the uncertainty information can be mapped from one visual hierarchical space onto another and can be transformed from one level to another. Incorrectly designing the mapping functions will mislead the uncertainty reasoning. The lack of knowledge about using proper evidence to support a proper hypothesis will cause difficulty in designing mapping functions. For example, in the available training data, we don't have enough knowledge about the relations between the fourth graytone moment of a region and the primary components of a resistor. Blindly designing a mapping function would introduce considerable bias and uncertainty into the reasoning process. However, the mapping function design can be considered as a learning process for the uncertainty reasoning system. Mapping functions can be modified and tuned by more training data.

So far we have discussed some important aspects in designing an uncertainty reasoning system within the framework of this dissertation. These aspects should be taken into consideration in the design process. To illustrate the design of three visual hierarchies, an example for the problem of resistor recognition is shown in Figure 78 through 80. (In Figure 79, only level 4 to 5 are shown, other levels have the similar structure.)

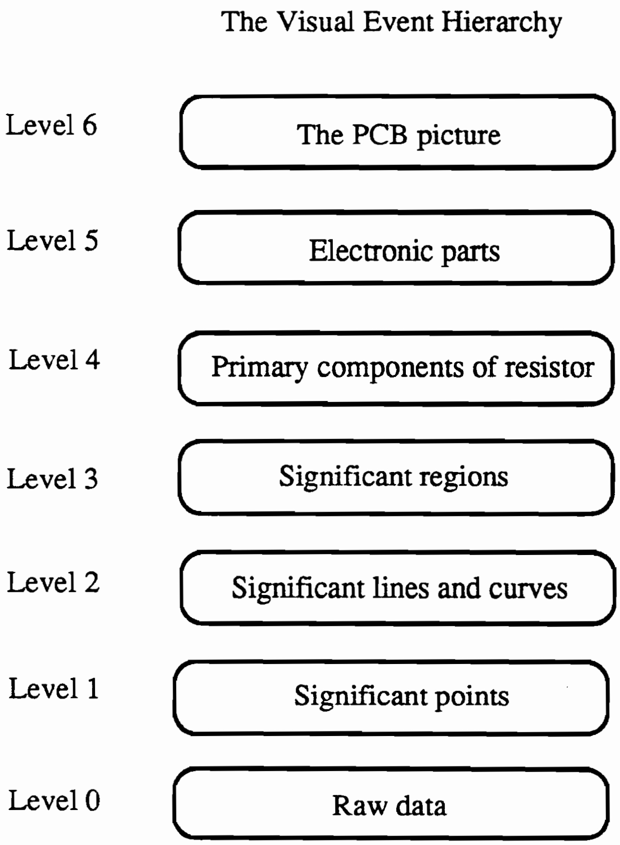


Figure 78. The visual event hierarchy for the resistor problem.

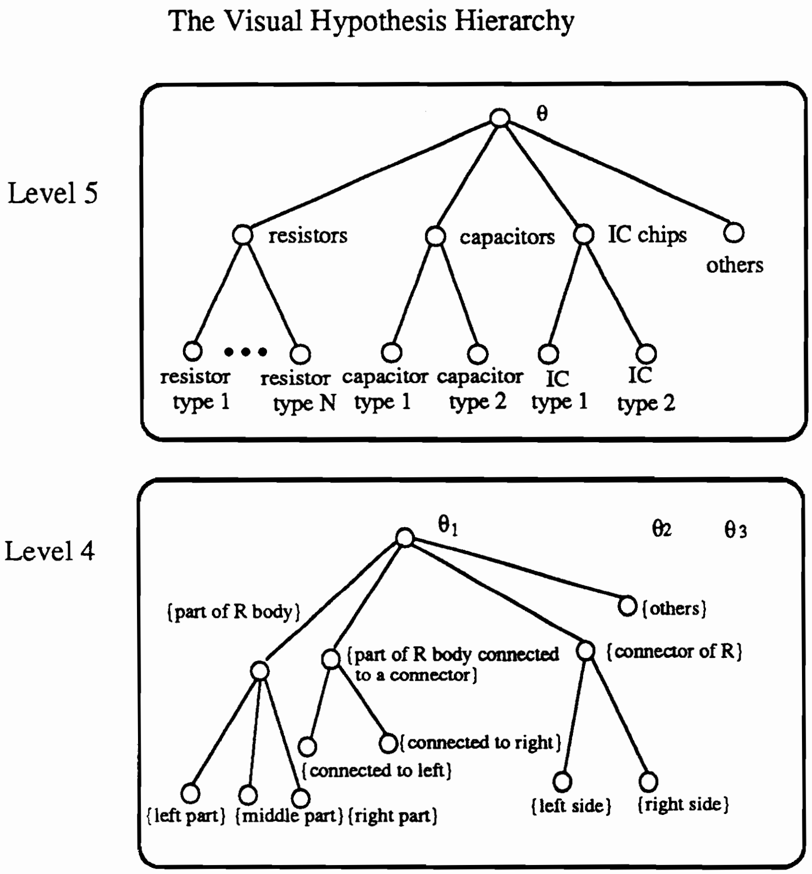


Figure 79. The visual hypothesis hierarchy for the resistor problem.

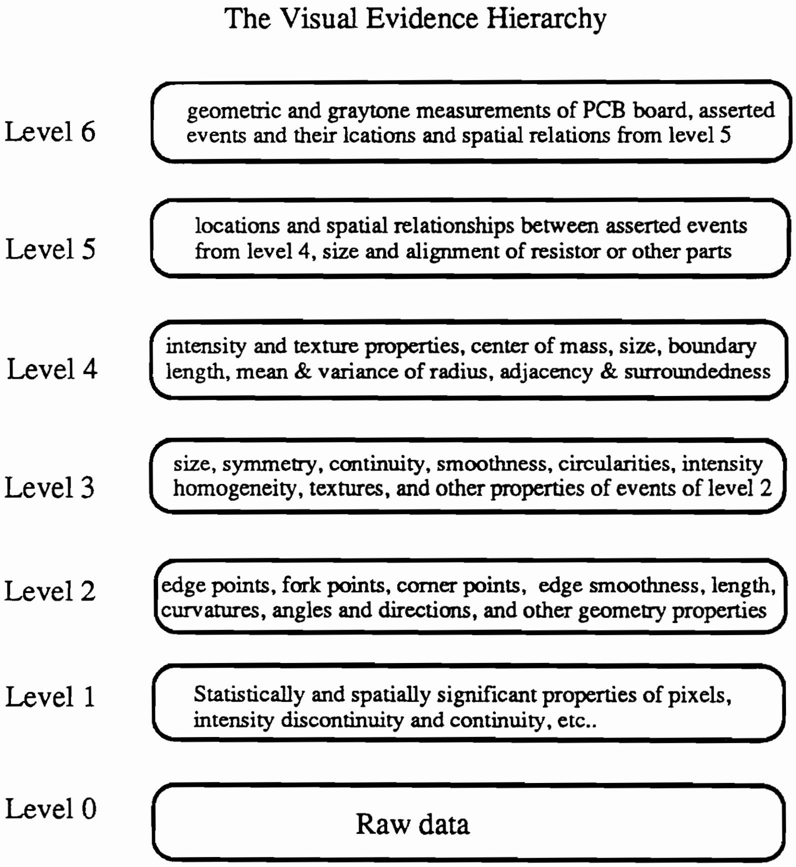

The Visual Evidence Hierarchy
Level 6
geometric and graytone measurements of PCB board, asserted events and their lcations and spatial relations from level 5
Level 5
locations and spatial relationships between asserted events from level 4, size and alignment of resistor or other parts
Level 4
intensity and texture properties, center of mass, size, boundary length, mean & variance of radius, adjacency & surroundedness
Level 3
size, symmetry, continuity, smoothness, circularities, intensity homogeneity, textures, and other properties of events of level 2
Level 2
edge points, fork points, corner points, edge smoothness, length, curvatures, angles and directions, and other geometry properties
Level 1
Statistically and spatially significant properties of pixels, intensity discontinuity and continuity, etc..
Level 0
Raw data

Figure 80. The visual evidence hierarchy for the resister problem.

Chapter IX
CONCLUSION

The problem of reasoning about uncertainty visual information in computer vision systems is addressed in this dissertation. We define this problem on the basis of three hierarchical spaces: the hierarchical visual event space, the hierarchical visual evidence space, and the hierarchically related visual hypothesis space. Then the uncertainty reasoning is carried out by means of mapping information from one space to another and transforming uncertainty information from one level to another using *a priori* knowledge.

We have shown a computationally efficient way to apply D-S theory for reasoning in such hierarchically structured spaces. We also have resolved the problems which arise from adapting D-S theory to such a computer vision system. That includes problems of uncertainty visual information representation, uncertainty transfer and evidence discount, dependent measurements, and applying Dempster's rule under conflicting information.

Because a frame of discernment consists of a set of mutually exclusive visual events, the reasoning process is visual evidence driven, and the goal is to assign labels to visual events. Meanwhile, the interaction of hierarchical evidence can provide a way to deduce a visual event indirectly from other relevant hypothesis. However, at each level of the event hierarchy we may pose additional D-S problems involving spatial relationships among hypotheses at that level and other evidence. The results of these subsidiary D-S problems contribute to the belief functions for the original hypotheses at that level. Therefore, we now have a framework for incorporating relational model into an event driven reasoning process.

At the system level, the Dempster-Shafer engine propagates interpretations upward from the raw data to the highest level interpretations. On the other hand, at any particular level the verification process propagates interpretations downward. The verification process can also be based upon the Dempster-Shafer engine if desired. This provides a bidirectional active reasoning through both bottom-up and top-down processes.

We also described how to implement this reasoning system. The reasoning paradigm, the data structure, and the control mechanism are briefly presented. Finally, we have demonstrated some applications of such a framework in preprocessing, lower level processing, and higher level processing. DNESYS–an expert system also demonstrated its merit to support this framework.

On the minus side, establishing reasonable event and relational hierarchies and implementing the mapping functions to compute the initial $bpa's$ is an arduous process that may not be as intuitive as one would like. However, our structure does enable us to solve very complex reasoning problems in computer vision on a more axiomatic basis.

This framework of uncertainty reasoning algorithms for the hierarchical visual recognition at multiple levels of abstraction can be used for more applications than we have demonstrated. Consequently, many improvements and new methods could be developed, low level processing could be more intelligent and more reliable, and high level processing could be more powerful and more flexible in computer vision systems.

BIBLIOGRAPHY

Aloimonos, J., I. Weiss and A. Bandyopuahyay, "Active Vision," *Proceedings of IEEE First Int. Conf. on Computer Vision*, pp. 35–54, 1987.

Andress, K. and A. Kak, "A Production System Environment for Integrating Knowledge with Vision Data," *Proceedings of Workshop on Spatial Reasoning*, pp. 1–12, 1987.

Andress, K. and A. Kak, "Evidence Accumulation and Flow of Control in a Hierarchical Spatial Reasoning System," *AI Magazine*, pp. 75–94, Summer 1988.

Ballard, D., C. Brown, and J. Feldman, "An Approach to Knowledge-Directed Image Analysis," in *Computer Vision Systems*, A. Hanson and E. Riseman (Eds.), New York: Academic Press, 1978.

Bajcsy, R., "Active Perception vs. Passive Perception," *Proceedings of the 3rd IEEE Workshop on Computer Vision: Representation and Control*, pp. 55–59, 1985.

Barnett, J. A., "Computational Methods for a Mathematical Theory of Evidence," in *Proceedings IJCAI-81,* Vancouver, BC, pp. 868–875, 1981.

Barrow, H. and J. Tenenbaum, "Computational Vision," *Proceedings IEEE* (69) No. 5, pp. 572–595, May 1981.

Ben-Arie, J. and A. Meiri, "3-D Object Recognition by Optimal Matching Search of Multinary Relations Graphs," *Computer Vision, Graphics, and Image Processing* (37), pp. 345–361, 1987.

Besl, P. and Ramesh Jain, "An Overview of Three-Dimensional Object Recognition," RSD-TR-19-84, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan, 1984.

Bhatnagov, R. and L. N. Kanal, "Handling Uncertain Information: A Review of Numeric and Non-Numeric Methods ," in *Uncertainty in AI* , L. N. Kanal and J. F. Lemmer (Eds.), Elsevier Science Publishers B. V. pp. 3–26, 1986.

Binford, T., "Survey of Model-Based Image Analysis Systems," *The International Journal of Robotics Research* (1), No. 1, pp. 18–63, 1982.

Brady, M., "Computational Approaches to Image Understanding," *Computing Surveys* (14) No. 1, pp. 3–71, March 1982.

Brady, M., "Intelligent Vision," in *AI in the 1980's and Beyond*, Grimson and Patil (Eds.), The MIT Press, pp. 201–241, 1987.

Brooks, R., "Symbolic Reasoning Among 3-D Models and 2-D images," *Artificial Intelligence* (17), pp. 285–348, August 1981.

Buchanan, B. and E. Shortliffe, "Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project," Addison-Wesley, Reading, MA, 1984.

Chaniak, E., "The Bayesian Basis of Common Sense Medical Diagnosis," *Proceedings AAAI*, pp. 70–73, 1983.

Carnap, R., "Logical Foundations of Probability," The University of Chicago Press, 1950.

Carnap, R., "Meaning and Necessity, A Study in Semantic and Model Logic," Phoenix Books, University of Chicago, 1960.

Cheng Y. and Kashyap, "Study of the Different Methods for Combining Evidence," *SPIE (635) Applications of Artificial Intelligence III*, pp. 384–392, 1986.

Davis, L. and Hwang, "The SIGMA Image Understanding system," *Proceedings of the Third Workshop on Computer Vision*, pp. 19–26, October 13–16, 1985.

Dempster, A., "Upper and Lower Probability Inferences Based on a Sample From a Finite Univariate Population," *Biometrika* (54), pp. 515–528, December 1967.

Dempster, A., Ann. Math. Statistics, 38. pp. 325, 1967.

Dempster, A., "A generalization of Baysian Inference," J. R. Stat. Soc. Series B, No. 30, 1968.

Dubois, D. and H. Prade, "Combination and Propagation of Uncertainty with Belief Functions," *Proceedings of Ninth Int. Joint Conf. Artificial Intelligence*, Los Angeles, CA, pp. 18–23, August 1985.

Dubois, D. and H. Prade,"Representation and combination of uncertainty with belief functions and possibility measures," *Computing Intelligence* No. 4, pp. 244–264, 1988.

Duda, R., P. Hart, and N. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," *Proceedings of 1976 National Computer Conference*, pp. 1075–1082, 1976.

Duda, R., P. Hart, K. Konolige, and R. Reboh, "A Computer-Based Consultant for Mineral Exploration," Tech. Rep. SRI International, 1979.

Elassal, A. A. and V. M. Caruso, "Digital Elevation Models," *U. S. Geological Survey Circular 895-B*, 40 p., 1983.

Faugeras, O. and M. Hebert, "The Representation, Recognition, and Locating of 3-D Objects," *The International Journal of Robotics Research* (5), No. 3, pp. 27–52, 1986.

Frisby, J., "Seeing: Illusion, Brain, and Mind," Oxford University Press, 1980.

Garland, E. and R. W. Ehrich, "A GIPSY Primer," SDA Laboratory, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, December 1987.

Garvey, T., "Evidential Reasoning for Helicopter Route Planning," *IEEE Trans. on Geoscience and Remote Sensing* (GE-25), No. 3, pp. 294–304, May 1987.

Gordon, J. and E. H. Shortliffe, "A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space," *Artificial Intelligence* (26), pp. 323–357, July 1985.

Gregory, R. L., "Perceptions as Hypotheses," *Phil. Trans. R. Soc. Lond. B290*, pp. 181–197, 1980.

Grimson, W. E. L., "Making Robots See and Feel," in *AI in the 1980's and Beyond*, Grimson and Patil (Eds.), The MIT Press, pp. 243–280, 1987.

Haralick, R., "Computer Vision Theory: The Lack Thereof," *Proceedings of the 3rd IEEE Workshop on Computer Vision: Representation and Control,* pp. 113–121, 1985.

Haralick, R. M., J. B. Campbell, and S. Wang, "Automatic Inference of Elevation and Drainage Models from a Satellite Image," *Proceedings of the IEEE* (73), No. 6, pp. 1040–1053, June 1985.

Haralick, R., "Documentation of GIPSY Command ZEROCR," Version A.01, November 1981.

Hanson, A. and E. Riseman, "VISIONS: A Computer System for Interpreting Scenes," in *Computer Vision Systems*, A. Hanson and E. Riseman (Eds.), Academic Press, NY, 1978.

Hanson, A. and E. Riseman, "A Methodology for the Development of General Knowledge-Based Vision Systems," *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, pp. 159–170, December 1984.

Henkind, A. B. and M. C. Harrison, "An Analysis of Four Uncertainty Calculi," *IEEE Trans. on Syst. Man. Cybern.,* (18), No. 5, pp. 700–714, September/October 1988.

Heyting, A., "Intuitionism: An Introduction, Studies in Logical Foundations of Mathematics," North Holland, Amsterdam, 1971.

Hoffman, R. and A. Jain, "An Evidence-Based 3D Vision System for Range Image," *Proceedings of IEEE First Int. Conf. on Computer Vision,* pp. 521–525, 1987.

Horton, R. E., "Erosional Development of Streams and Their Drainage Basins, Hydrophysical Approach to Quantitative Morphology," *Geological Society of America Bulletin* (56), pp. 275–370, 1945.

Howard, A. D., "Optimal Angles of Stream Junctions: Geometric Stability to Capture and Minimum Power Criterion," *Water Resources Research* (7), No. 4, pp. 863–873, 1971.

Hummel, R. and M. Landy, "A Statistical Viewpoint on the Theory of Evidence," *IEEE Trans. on Pattern Analysis and Machine Intelligence* (10), No. 2, pp. 235–247, March 1988.

Huertas, A. and G. Medioni, "Edge Detection with Subpixel Precision," *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, pp. 63–74, October 1985.

Jenson, Susan K., "Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Model Data," *USGS Publication*, November 15, 1984.

Kanizsa, G., "Organization in Vision," Praeger, New York, 1979.

Lehrer, N., G. Reynolds, and J. Griffith, " A Method for Initial Hypothesis Formation in Image Understanding," *Proceedings of IEEE First Int. Conf. on Computer Vision*, pp. 578–585, 1987.

Lee, C., "A Comparison of Two Evidential Reasoning Schemes," *Artificial Intelligence,* No. 4, pp. 127–134, 1988.

Lee, J., "Digital Image Smoothing and The Sigma Filter," *Computer Graphics and Image Processing*, Vol. 24, pp. 255–269, 1983.

Lee, J., "Refined Filtering of Noise Using Local Statistics," *Computer Graphics and Image Processing*, Vol. 15, pp. 380–389, 1981.

Lesser, V. R. and L. D. Erman, "A Retrospective View of the HEARSAY-II Architecture," *Proc. Fifth International Joint Conference on Artificial Intelligence, IJCAI, Cambridge, MA.,* pp. 790–800, August 1977.

Li, Z. and L. Uhr, "Pyramid Vision Using Key Features to Integrate Image-Driven Bottom-Up and Model-Driven Top-Down Processes," *IEEE Trans. on Sys., Man, and Cyb.* (SMC-17) No. 2, pp. 250–263, March/April 1987.

Lowe, D., "Perceptual Organization and Visual Recognition." New York: Kluwer Academic Publishers, 1985.

Lowe, D., "Three-Dimensional Objects Recognition from Single Two Dimensional Images," *Artificial Intelligence* (31), pp. 355–395, 1987.

Marr, D., "Early Processing of Visual Information," *Phil. Tran. Royal Soc.* B 275, pp. 483–519, October 1976.

Marr, D., "Vision, A Computational Investigation into the Human Representation and Processing of Visual Information," San Francisco: W. H. Freeman and Company, 1982.

Marr, D. and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London, B207*, pp. 187–217, 1980.

Malik, J., "Interpreting Line Drawings of Curved Objects," *International Journal of Computer Vision* No. 1, pp. 73–103, 1987.

McKeown, D. and J. Pane, "Alignment and Connection of Fragmented Linear Features in Aerial Imagery," *Proceedings of IEEE Conference on CVPR*, pp. 55–61, June 1985.

Minsky, M., "A Framework for Representing Knowledge," in *The Psychology of Computer Vision,* P. H. Winston, Ed. New York: McGraw-Hill, pp. 211–267, 1975.

Mulgaonkar, P., "Analyzing Perspective Line Drawings Using Hypothesis Based Reasoning," Ph.D. Dissertation, Department of Computer Science, Virginia Polytechnic Institute and State University, August 1984.

Nagao, M. and T. Matsuyama, "A Structural Analysis of Complex Aerial Photographs," Plenum Press, New York, 1980.

Nalwa, V., "Line-Drawing Interpretation: A Mathematical Framework," pp. 103–124, 1988.

Nazif, A. and M. Levine, "Low Level Image Segmentation: An Expert System," *PAMI* (6), No. 5, pp. 555–577, 1984.

Nevatia, R. and K. Babu, "Linear Feature Extraction and Description," *CGIP* (13), pp. 257–269, 1980.

O'Callaghan, J. F. and David M. Mark, "The Extraction of Drainage Networks from Digital Elevation Data," *Computer Vision, Graphics, and Image Processing* (28), pp. 323–344, 1984.

Pednault, E., S. Zuker and L. Muresan, "On the Independence Assumption Underlying Subjective Bayesian Updating," *Artificial Intelligence* (16), pp. 213–222, 1981.

Pearl, J., " On Evidential Reasoning in a Hierarchy of Hypotheses," *Artificial Intelligence* (28), pp. 9–15, 1986.

Qian, J. and R. Ehrich, "A Framework for Uncertainty Reasoning in Hierarchical Visual Evidence Space,"*Proceedings of 10th International Conference on Pattern Recognition / Computer Vision,* Atlantic City, New Jersey, June 1990.

Qian, J. and R. Ehrich, "Uncertainty Reasoning in Feature Space for Knowledge Based Computer Vision," submitted to *IEEE Transaction on Pattern Analysis and Machine Intelligence,* December 1989.

Qian, J., R. Ehrich, and J. Campbell, "DNESYS–an Expert System for Automatic Extraction of Drainage Networks from Digital Elevation Data," *IEEE Trans. on Geoscience and Remote Sensing*, (28), No. 1, pp. 29–45, January 1990.

Qian, J. and R. Haralick, "Discrimination of Water Regions from Shadow Regions on SAR Imagery," *Proceedings of SPIE's 1990 Symposium on Remote Sensing and Signal & Image Processing,* Orlando, Florida, April 15–18, 1990.

Qian, J. and K. Yu, "Multi-threshold Adaptive Filtering– an AI Approach to Image Enhancement," to appear in *International Journal of Intelligent Systems.*

Qian, J., K. Yu, and R. Haralick, "Multi-threshold Adaptive Filtering for Image Enhancement," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* Tokyo, Japan, April 1986.

Qian, J., "Discrimination of Water from Shadow Regions on Radar Imagery Using Computer Vision Techniques," M.S. Thesis, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1985.

Quillian, M.,"Semantic Memory," in *Semantic Information Processing,* M. Minsky, ed., MIT Press, Cambridge, Mass., pp. 354–402, 1968.

Rescher, N., "Many–valued Logic." New York: McGraw-Hill, 1969.

Rescher, N. and Urquhart A., "Temporal Logic." New York: Springer–Verlag, 1971.

Riseman, E. and A. Hanson, "Computer Vision Research at the University of Massachusetts–Themes and Progress," *International Journal of Computer Vision,* No. 2, pp. 199–207, 1989.

Roberts, L., "Machine Perception of Three Dimensional Solids," in *Optical and Electro Optical Information Processing,* J. T. Tippet et. al. (Eds.), MIT Press, Cambridge, Mass., pp. 159–197, 1965.

Rogers, D. and J. Adams, "Mathematical Elements for Computer Graphics," New York: McGraw-Hill, pp. 116–155, 1976.

Rosenfeld, A. and R. Klette, "Degree of Adjacency or Surroundeness," *Pattern Recognition*, pp. 169–177, January/February 1985.

Rosenfeld, A., "Image Analysis: Problem, Progress, and Prospects," *Pattern Recognition* (17), No. 1, pp. 3–12, 1984.

Rosenfeld, A., " 'Expert' Vision Systems: Some Issues," *CGIP* (34), pp. 99–117, 1986.

Roy, A. G., "Optical Angular Geometry Model of River Branching," *Geographical Analysis* (15), No. 2, pp. 87–96, 1983.

Shafer, G., "A Mathematical Theory of Evidence." Princeton, NJ: Princeton Univ. Press, 1976.

Shafer, G., "Probability Judgement In Artificial Intelligence," in *Uncertainty in AI*, L. Kanal and J. Lemmer (Eds.), pp. 127–136, 1986.

Shafer, G. and R. Logan, "Implementing Dempster's Rule for Hierarchical Evidence," *Artificial Intelligence* (33), pp. 271–298, November 1987.

Shapiro, L. G. and R. M. Haralick, "Structural Descriptions and Inexact Matching," *IEEE Trans. on PAMI* (3), No. 5, pp. 504–519, September 1981.

Shapiro, L., "The Role of AI in Computer Vision," *IEEE Proceedings of the 3rd workshop on Computer Vision: Representation and Control, Bellarive, Michigan*, pp. 76–81, October, 1985.

Shirai, Y., "Recognition of Man-made Objects Using Edge Cues," in *Computer Vision Systems*, A. Hanson and E. Riseman (Eds.), Academic Press, New York, 1978.

Shortliffe, E. and B. Buchanan, "A Model of Inexact Reasoning in Medicine," *Math. Biosci.* (23), pp. 351–379, April 1975.

Shortliffe, E., "Computer-Based Medical Consultations: MYCIN," American Elsevier Publishing, 1976.

Walters, D., "Selection of Image Primitive for General-Purpose Vision Processing," *CVGIP* (37), pp. 261–298, 1987.

Wesley, L. P., "Evidential Knowledge-based Computer Vision," *Optical Engineering*, (25), No. 3, pp. 363–379, March 1986.

Yager, R., "On the Dempster-Shafer Framework and New Combination Rules," *Information Sciences*, (41), pp. 93–136, 1987.

Yang, G. and T. Huang, "Median Filters and Their Application to Image Processing," Purdure University, TR–EE 80-1, 1, 1980.

Zadeh, L., "Fuzzy Sets," *Info. and Control* (8), pp. 338–353, 1965.

Zadeh, L., "Fuzzy Logic and Approximate Reasoning," *Syntheses* (36), pp. 407–428, 1975.

Zadeh, L., "Review of 'A Mathematical Theory of Evidence' by G. Shafer," *The Artificial Intelligence Magazine,* (5), pp. 81–83, 1984.

Zadeh, L., "Is Probability Theory Sufficient for Dealing with Uncertainty in AI: A Negative View," in *Uncertainty in AI*, L. Kanal and J. Lemmer (Eds.) pp. 103–116, 1986.

Zerniz, E. R., "Drainage Patterns and Their Significance," *Journal of Theoretical Biology* (40), pp. 498–521, 1932.

VITA

The author was born in Zhejiang, China in June, 1948. He receive the Diploma in Electrical Engineering from the Fuzhou University, China, in 1975. He was an engineer in the Fourth Institute of the Ministry of Electronic Industry of China and engaged in research on communication systems from 1975 to 1979. From 1979 to 1983, he worked as a research staff at the Computer Digital Image Processing Laboratory, Institute of Remote Sensing, Academia Sinica, Beijing, China, and joined research work on image processing, especially on the development of an image processing system. From 1983 to 1984, he was a visiting scholar at the Spatial Data Analysis Laboratory, Virginia Polytechnic Institute and State University, Virginia, where he was a research assistant from 1984 to 1990. In Spring 1984, he enrolled in the Virginia Polytechnic Institute and State University to continue his study in Electrical Engineering in the field of computer vision, digital signal/image processing, pattern recognition, artificial intelligence, and robotics. He received his M.S. degree in Electrical Engineering in 1985 and completed requirements for a Doctor of Philosophy in Electrical Engineering in April 1990, both from the Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Jianzhong Tian