A Computational Framework for Hypothesis Based Reasoning and its Applications to Perspective Analysis

Prasanna G. Mulgaonkar*; Robert M. Haralick**; Linda G. Shapiro**

*SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

**Machine Vision International
325 E. Eisenhower Parkway
Ann Arbor, MI 48104

ABSTRACT

A new hypothesis based reasoning scheme is presented for the analysis of sensory data generated by a well-defined physical processes. Hypotheses are represented by subsets of allowable attributed relational tuples describing relationships between the entities in the world. Interpretations are defined to be those hypotheses which are consistent with the structure imposed by the sensing process and compatible with the observed sensory data.

Although the problem of determining the interpretations of a given sensory image is equivalent to theorem proving in general, recasting it in the above hypothesis generation form makes it equivalent to a propositional logic consistency problem with side conditions on the attributes. In this form, the problem of finding the best or maximal interpretations reduces to an NP complete problem.

We show how to exploit the structure of the search space of all hypotheses to significantly reduce the computational complexity of this NP-complete problem. The technique is applied to the problem of interpreting line drawings of perspective projections of polyhedral objects.

Introduction

One of the central aspects of intelligence is the ability to reason about and operate in a real world environment using sensory information. We have the ability to use various types of visual, auditory, and tactile information to understand the contents of the world around us. The process by which such information is generated from the world is a deterministic physical process. However information is lost in the process, since there is not a unique combination of objects in the world which could give rise to a particular combination of sensory patterns.

The reason for our capacity to compute the inverse mapping for any sensory process seems to be that the three-dimensional world has some regularity and that the physical system which produces the stimulus strongly constrains the possible arrangement of the three-dimensional world. Of all the possible ways in which the stimulus could be interpreted, only a few are actually consistent with the physical process involved.

We believe that the process of interpretation can be considered to be one of formulating "educated guesses" as to the nature of the three-dimensional objects in the world and utilizing the constraints of three-dimensional coherence and world knowledge about the physical processes involved to select the reasonable arrangements from possible guessed arrangements. In essence, we hypothesize what the world could consist of and retain only the consistent hypotheses.

In this paper we present a computational framework for hypothesis generation and testing. We formally define the notion of a hypothesis and develop the space of all hypotheses over which the interpretation process searches for the interpretation. The search problem is NP complete. However the search space is structured by the constraints of the physical sensory system and this structure can be utilized to reduce the size of the search involved. Examples of the computational paradigm are presented where appropriate, based on the problem of analyzing line drawings of perspective projections of three-dimensional objects.

Literature Review

Hypothesis generation and testing has been around for a long time as a scientific analysis technique. Its early origins can be traced back to ancient Greek scientists notably Aristarchus of Samos [1] who formulated the heliocentric theory as a hypothesis which was capable of explaining the body of observations about the movement of the planets. The school of thought to which his arguments belonged have been termed "saving the phenomena" [2]. Philosophy also has a branch of non-demonstrative logic which is called the Method of Hypothesis [3]. Although the philosophers are interested in the logical form of the hypothetical arguments, their concepts bear some resemblance to the computational theories developed in this paper.

Recent computational theories of hypothesis aim more at the interaction with propositional and predicate logic. One of the earliest works of a formal nature is the work of N. Rescher [13], which deals specifically with counter-factual hypothesis manipulation. However in the categorization of hypothetical arguments employed by Rescher our form of hypothesis would be similar to that called "Problematic". Rescher does not formalize this branch of hypothesis any further.

In the field of Artificial Intelligence, one of the first successful expert systems was DENDRAL [4] which also used a form of hypothesize and test to generate the possible structures which could account for observed mass-spectra of organic molecules. The search space in this program was controlled by means of a constraint generation algorithm which would prune inconsistent combinations of molecular structures.

Several recent vision systems have also been constructed to use knowledge based reasoning. These systems integrate knowledge sources codifying world knowledge into the image interpretation process. The largest and most complete such system is the VISIONS system [5]. The use of world knowledge in the image segmentation field and the integration of different sources of sensory input has been attempted by Levine [6]. The use of distributed control in which groups of cooperating processes worked jointly on a global information base was pioneered in the HEARSAY system [16,17]. HEARSAY-I and HEARSAY-II worked in the domain of speech understanding. Several independent computation sources worked on the common database of information. These modules would lie dormant, waiting for the appropriate combination of conditions under which they could work and modify the database based on their computation. A similar control strategy is employed by the VISIONS system.

The main difference between the research presented in this paper and the previous hypothesis based systems is that for the first time, we are attempting to explicitly specify what the search space is and control the process which searches in that space for a solution.

Hypothesis Generation and Testing.

The basic idea behind the concept of hypothesis generation and testing is simple. There is some physical world which consists of several entities and various ways in which these entities relate to each other. The actual relationships between the entities of the physical system are unknown. What is known is the types of relationships that they can participate in. The actual physical system is not accessible for the purpose of determining the relationships between the entities. However, through some well-defined process, it is possible to make some measurements which depend on the relationships. An interpretation of the observations consists of determining the unknown relationships between the entities. If these interpreted relationships are correct, they would give rise to the observations.

The Optimal Hypothesis Problem

An interpretation is a hypothesis which is consistent and is "optimal" in some sense. Although the treatment in this section is formal, it is helpful to keep in mind the intended physical interpretations for the symbols. To that end, consider the application of the hypothesis system to the problem of determining the three-dimensional object structure which gives rise to an observed perspective image.

The world being sensed consists of a set of entities and the attributed relationships between the entities. Let E be the finite set of entities $E = \{e_1,\ldots,e_n\}$ about which we want to reason. In computer vision, these entities can be three-dimensional lines, planes, and points as well as two-dimensional entities such as projected arcs.

The true but unknown state of the world is specified by a union of named attributed relations between these entities. Each relation $\underline{r}_i$ in this union, has a name r_i, and an attribute set a_i. The number of entities which $\underline{r}_i$ relates is the order of $\underline{r}_i$ and we denote it by o_i. Thus each $\underline{r}_i$ can be represented as a subset

$$\underline{r}_i \subseteq \{r_i\} \times E^{o_i} \times \Psi_i$$

where each $f \in \Psi_i$ is a single valued relation which associates a real number for each attribute in a_i. That is, $f \subseteq a_i \times \mathbb{R}$. This associated number is called the attribute value. The state of the world is then represented by

$$\underline{R} = \bigcup_{i=1,m} \underline{r}_i .$$

In the case of the perspective geometry problem, the relations are spatial relations such as parallel lines, groups of lines in a plane or pairwise touching lines or arcs. These relations have attributes such as distance between lines, coordinates of the starting and ending points or directions of plane normals.

The information given to the interpretation process consists of the relation names, orders of the relations, names of the relation attributes, the entities in the observed instance of the world and measurements made by the sensing process. The interpretation process constructs the best guesses for the unknown world state $\underline{R}$. The basis for the interpretation is the observed measurements and the known form or semantics of $\underline{R}$

Let P be the set of all possible instances of tuples from any relation. Thus

$$P = \bigcup_{i=1,m} [\{r_i\} \times E^{o_i} \times \Psi_i]$$

Any state of the world is a subset of the set P. The interpretation process must effectively search all subsets of P to find the best guesses for $\underline{R}$. This is an enormous search problem.

However, by effectively using the knowledge about the semantics of the relationships between the world entities, it is possible to reduce the search space considerably. Because the process which generates the sensory data from the real world is a well defined physical process, the combinatorial search for interpretations does not have to be performed over the entire space of subsets of P defined above. The measurements and the physics of the projection process relate the attribute value relations in the tuples in $\underline{R}$ to the entities involved in the relations. It is possible to use expert knowledge encoded as inference engines to compute large portions of the attribute value relations in the last component of each tuple given only the measurements and a set of tuples complete except for the last component as being in $\underline{R}$. This means that the relational portion of the tuples (the projection over all but the last position) determines the remaining position (the attribute value relation). Thus the combinatorial search space involved is all subsets of

$$P' = \bigcup_{i=1,m} [\{r_i\} \times E^{o_i}]$$

Tuples from P which are examined during the process of searching for an interpretation, are called predictions.

A hypothesis H is a subset of the set of predictions $H \subseteq P$. A hypothesis may be consistent which is denoted by C or inconsistent which is denoted I. Thus there exists a function $F:H \longrightarrow \{C,I\}$ which determines if a hypothesis is consistent or not. A hypothesis is consistent if upon using all inference engines, the computed attribute value relations for the tuples in H are each single valued. We will have more to say about the nature of F later.

An interpretation I is a hypothesis set $H \subseteq P$ such that $F(H) = C$. We are interested in "best" possible interpretations I. That is, we are interested in any I which maximizes some property g(I). Best possible interpretations may not be unique.

The simplest g(h) is $g_1(H) = \#H$, the number of consistent predictions about the domain. The only requirement on the function g(H) is that it be integer and bounded from above. With only that restriction, any appropriate function can be used.

With this formalization, the problem of determining the optimal hypothesis can be shown to be an NP complete problem, provided that consistency of a hypothesis can be determined in polynomial time. The proof uses the fact that the consistent labeling problem (CLP) is a known NP Complete problem [8], and constructs a polynomial time transformation from the CLP to the hypothesis determination problem. For complete details on the proof, the reader is refered to [9].

In the next section we develop ideas about consistency checking. Unless consistency checking can be performed efficiently, the problem of generating the optimal hypothesis problem will be practically intractable. We will show that for a given hypothesis, we can determine consistency without performing any search.

Determining Consistency of a Hypothesis

The sensory process can have parameters. For example, the perspective image produced by a camera depends not only on the arrangement of entities in the scene but also on the location and the focal length of the camera. The parameters of the sensory process can be considered to be attributes of Ø-th order relations (those which involve no world entities). Based on physical laws, the sensory process interacts with these relationships and their attributes and produces a sensory pattern consisting of sensory entities and their relationships which are also attributed. For purposes of discussion, we can call this produced pattern an "image" of the world. The attributed relationships in the image are fixed and depend on the instance of the world being sensed. These relationships between the sensory entities are termed measurements.

Formally, an image may be characterized by the following. An image consists of a set of entities $E' = \{e_1', \ldots, e_m'\}$, where each e_i' is the image of the corresponding world entity e_i. The entities e_i' are called sensory entities.

Sensory entities stand in named attributed relationships to each other. These relations are termed sensory relations and are defined as $R' = \{R_1', \ldots, R_k'\}$ where each R_i' is an attributed relation over the sensory entities. Each relation R_i' has a name r_i', and an attribute set a_i'. The relation r_i' relates o_i' sensory entities. Thus each r_i can be represented as the subset

$$R_i' \subseteq \{r_i'\} \times E'^{o_i'} \times \Psi_i'$$

where each $f' \in \Psi_i'$ is a single valued relation which assigns a value for each attribute in a_i' taking a value. We use the term measurement (denoted M) to be $M = (E', R', O', A')$ where O' is the set of all the o_i' and A' is the set of all attribute names. Note that all the information in M is known and determined unambiguously from the image.

A hypothesis is a collection of relational tuples. These tuples define relationships between entities in the real world. The only way to check if a hypothesis is consistent is to determine if the physics of the sensory system would give rise to the observations if the hypothesis were correct.

Inference Engines

For the purpose of efficiency, it is necessary to organize the knowledge about the inference process into compact manageable units.

We term these computational units "inference engines". Inference engines operate on relational tuples and measured attributes of the image, and from them, compute values for the unknown attributes of the hypothesis. A hypothesis will be declared inconsistent if there exist two inference engines which both compute a value for the same attribute, and the two values are incompatible.

To be able to prove interesting properties about the process of using these computing units, we tighten the definition of inference engines as follows: inference engines are modular computation units which accept as input, a specific set of attributed relation tuples from the possible relationships between the world entities and a set of measurements taken from the sensory input. Based on the measurements and on previously computed attributes of the relational tuples, they compute values for other attributes of the input tuples. The initial hypothesis has all the attribute values unknown. No attribute has a value. The attribute value relations are empty. Attributes can only be given values based on the computations performed by the inference engines. An inference engine is defined to be in cannonical form if it computes the value of only one attribute.

The questions that arise in this context deal with the termination of the stability of the computations and the termination of the process. In the remainder of this section, we state several lemmas leading up to a pair of theorems which show that consistency of hypotheses can indeed be defined in computationally stable terms, and that no search is needed to determine the order in which these engines need to be controlled.

An inference engine E is applicable to a subset $K \subseteq P$ iff the following two conditions are met:

a) K satisfies the input requirements of the engine E. That is K contains the relational tuples that E uses as a basis of its computation and that the attributes of these tuples which E uses in its computation, are all defined.

b) No proper subset of K satisfies the input requirements of E.

An application of an engine E to a set K is denoted as E(K) where E is applicable to K. An application of E to K may succeed or fail (with success and failure being as defined earlier). If an application succeeds, a new subset K' results which differs from K in that atmost one attribute of K which was previously "unknown", now has a value. To make the definition of E(K) complete, we provide that if the application fails, the output subset K' is the same as K. Thus E(K) = K' where if E(K) fails, K' = K. If however, the application succeeds, then K' differs from K by at most one value. That is,

$$\#\{x \mid x=(r,e_1,\ldots,e_x,\psi) \in K \text{ and } x'=(r,e_1,\ldots,e_x,\psi') \in K' \text{ and } \psi \neq \psi'\} \leq 1.$$

We are interested in running our inference engines on an arbitrary hypothesis set. To that end we define an inference step as E(H) where $H \subseteq P$ to be the application of E to some subset $K \subseteq H$ to which E is applicable. If the application is successful, we replace K by K' in the set H. In what follows, we will only be interested in the output of E(H) for the cases where E(H) succeeds. Therefore we can talk of E(H) = H' where H, $H' \subseteq P$.

We define a sequence of inferences on a hypothesis H to be a sequence $E_j(E_{j-1} \ldots (H) \ldots)$ of inference steps where each inference engine E_i is applied to the output of the previous inference step and each of the inference steps 1 through j-1 is successful. Further, if for some i and k, $1 \leq i,k \leq j$, E_i is the same inference engine as E_k, we will assume that the subset of H to which they apply are different. Sequence of inferences are denoted in functional form to emphasize the fact that though they are applied sequentially, each engine operates not on the original hypothesis but on the output of the previous inference step.

A sequence of inferences $E_j(E_{j-1}..(H)..)$ is called a terminating sequence if either $E_j(.)$ is a failed inference or there does not exist any other inference engine E_{j+1} which can be applied to the result of the sequence in order to extend it.

Lemma (1): Consider a sequence of applications of the inference engines to a hypothesis set $H \subseteq P$. If at the ith step, engine E_i is applicable, then E_i will be applicable at all steps $j > i$.

Lemma (2): If $E_i(E_{i-1}..(H)..)$ is a sequence of inferences such that application E_i fails, then for any sequence of inferences $E_i(E_{k'}(E_{k'-1}..(E_{i'}(E_{i-1}(..(H)..)$ where applications $E_{i'}$ through $E_{k'}$ succeed, E_i will still fail. In other words, postponing the application of a failing inference engine does not lead to a successful sequence.

Lemma (3): Let $E_k(\ldots E_2(E_1(H)\ldots)$ and $E_j(\ldots E_b(E_a(H)\ldots)$ be two sequence of inferences. Let the second sequence be a successful sequence (i.e. the application $E_j(.)$ succeeds). Let these sequences satisfy:

(a) Every E_i, $1 \leq i \leq k-1$ in the first sequence is also in the second sequence (though not necessarily in the same order).

(b) E_k is not in the second sequence.

Then:

$E_k(E_j..(H)..)$ is a valid sequence, i.e. E_k is applicable at the end of the second sequence.

Lemma (4): If $E_i(E_{i-1}\ldots(H)\ldots)$ is a sequence of inferences on the hypothesis H, then either it is a terminated sequence or it can be extended to form a terminated sequence.

Theorem (1): If $E_i(E_{i-1}\ldots(H)\ldots)$ is a sequence of inferences on the hypothesis H which ends in failure, then all possible terminating inference sequences end in failure.

Corollary: If $E_i(E_{i-1}\ldots(H)\ldots)$ is a terminating sequence of inferences which ends in success, then all possible terminating sequences will end with success.

Theorem (2): Ignoring permutations, there is atmost one successfully terminating sequence of inferences for a given hypothesis H.

Implications of the Theorems. The theorems tell us that the sequence of applications of inference engines to a hypothesis must terminate. Further, the result obtained does not depend on the order in which the engines are applied to the hypothesis set. The definition of applicability and inference steps tells us that adding new relational tuples into a hypothesis will not affect the inference engines that would work on the smaller hypothesis set. Further, given a consistent hypothesis, there is only one (ignoring permutations) sequence of applications which characterizes the set. There is no search involved in the determination of the consistency of a hypothesis because there is no order dependency. Thus the process of determining consistency is linear in nature.

One consequence of this definition of consistency is that if a hypothesis $H \subseteq P$ is consistent, then any subset $K \subseteq H$ is also a consistent hypothesis. Thus we can replace the notion of searching for an "optimal" hypothesis by searching in P for the "maximal" hypotheses. We define a subset $I \subseteq P$ to be an interpretation if it is consistent and is maximal in the sense that for any $p \in P-I$, the hypothesis $I+\{p\}$ is inconsistent. Note that there may be more than one possible interpretation to a scene and each interpretation will correspond to one such maximal subset. We now discuss one method of finding such an interpretation.

Searching for the Maximal Hypothesis

The number of possible subsets of a set of relational tuples is 2^n where n is the number of possible tuples. This number grows exponentially in n, and as n grows, the time taken to exhaustively search the entire space, grows prohibitively. However the possible solutions are related as was pointed out in the previous section. If some subset $K \subseteq P$ is a consistent hypothesis, then all its subsets are also consistent and can be ruled out of the search pattern. Secondly, since we are interested only in the maximal consistent subsets, we can attempt to "grow" the solutions in a sequential fashion.

The search space can be organized as a binary tree. Suppose there are n possible relational tuples. The tree would then have n+1 levels: Ø through n. At each node of the tree (with the exception of the leaf nodes) there are two children. For a node at depth i, $0 \leq i \leq n-1$, these children correspond to the selection of and the rejection of the relational tuple i+1 respectively. By convention, we choose the left child to correspond to the inclusion of the tuple and the right child to correspond to its elimination. Thus each leaf node corresponds to a subset in which some tuples have been selected and some left out.

The advantage of this representation is that the entire tree does not have to be stored at any time. The tree can be searched by a backtracking tree search in which once a subtree is examined, all information about it can be discarded. However, it is not convenient to prune subsets of known solutions in this representation, because the subsets are not related in any consistent or simply denoteable fashion. We will examine this problem and indicate the solution that allows rapid pruning of the tree.

Let us assume some ancillary data structure is used to record solutions (or consistent leaf nodes) as they are visited in a preorder traversal of the binary tree. Associated with this storage is the required set of procedures which can quickly determine if a given subset of P is a subset of any of the solutions which have been generated. One way to avoid outputing the subsets is to wait until a "solution" is generated and then check if it is a subset of a previously generated solution. Note that since the tree is traversed in preorder, and since the tree organization is as described earlier, the larger subsets of P are visited before the smaller ones. Thus there cannot be any case where we have a subset of P (say S) marked as a solution and then visit a leaf node which is also a solution but contains S.

This strategy forces all the nodes of the tree to be visited. However subsets of solutions are clustered together and detection of subsets early would improve the efficiency of the search process. We can prove two important theorems which deal with this problem.

Theorem (3): At any non-leaf node in the tree, define the "leftmost solution" to be the leftmost leaf node of the subtree rooted at the non-leaf node under consideration. If the leftmost solution in a subtree is a subset of a previously generated solution, then all the possible solutions in the subtree are subsets of previously generated solutions. This allows us to prune that entire subtree and back up.

Theorem (4): At any non-leaf node in the tree, define the "rightmost solution" to be the rightmost leaf node of the subtree rooted at the non-leaf node under consideration. If the rightmost solution is not a subset of any previously generated solution, then none of the possible solutions in the subtree can be subsets of previous solutions.

This theorem says that if a node passes the test for the rightmost solution, we can do away with the checking for subsets of previous solutions while we are exploring the subtree under the node. This allows the search procedure another degree of efficiency. In addition, forward checking [10] can be used.

Logical Inconsistency and Completeness for Pruning.

In addition to the numerical consistency criteria imposed by the inference engines, another form of consistency is important in the hypothesis generation process. This is termed the Logical Consistency of the space. For example, two lines cannot be simultaneously parallel and perpendicular. Logical consistency arises from the world semantics and it is often not possible to define in a numerical sense. Other examples of such logical consistency are the symmetry and transitivity of relations. These constraints can also be encoded as extensions of the forward checking which rule out possible future tuples from consideration. For example, if the left branch at any internal node takes the tuple (parallel line1 line2) into the hypothesis set, forward checking can rule out (perpendicular line1 line2). This dichotomy between the logical and numeric constraints parallels the distinction between "geometric" and "relational" consistency which was reported in previous work [11].

In addition to ruling out possiblities, the concept can be extended further to forcing the selection of certain tuples based on the tuples in a partially completed hypothesis. Note that if a partial consistent hypothesis is extended by adding a new tuple, it may become inconsistent. This will be tackled in the tree search by taking the right branch at the appropriate node. However in the case where the relation semantics dictate that the tuple must be included in a hypothesis, the resulting subset of P must be ruled out (although by the numerical computations of the inference engines, it is consistent). For example, a hypothesis which consists of two tuples (parallel line1 line2) and (parallel line2 line3) alone must be ruled out because the transitive nature of the parallel relation dictates that the tuple (parallel line1 line3) should also be included in the hypothesis. The original hypothesis is in that sense "incomplete".

This fact can be utilized in the tree search as follows. At any internal node in the tree, whenever a new tuple is added (i.e. a left branch is taken), all the tuples that are logically implied by the previously accepted set of tuples and the new tuple are also put into the partial subset. If any of these logically implied tuples have been ruled out by forward checking at previous levels, the subtree below the current node is discarded and the procedure backs up.

Thus at each node of the tree, tuples are not instantiated individually but in "chunks". Thus the effective depth of the tree decreases. The actual amount by which the tree search gets pruned depends on the degree to which subsets of tuples imply the selection of other tuples. As a simple example consider the space P with 3 tuples in it. Let these tuples be called a, b and c and let their semantics imply that tuple a and b together logically imply tuple c, and b and c imply a. The search space consists of $2^3 = 8$ subsets. However because of the chunking effect of the logical relationship, the number of viable subsets that must be searched reduces. The subsets {a,b}, {b,c} are ruled out leaving only six possible subsets to consider.

Applications to Computer Vision

In this section we apply the method of hypotheses developed in the previous sections to a computer vision problem. We are given a single image which is a line drawing of the perspective projection of some unknown three-dimensional object. The task to be undertaken is the analysis of the image and determination the orientations of the lines and planes in the three-dimensional scene being viewed. In the process of forming the perspective image, information is lost. There is typically not a unique combination of three-dimensional entities which could have caused a given image.

The entities and relations about which we reason are entities and spatial relations in the three-dimensional world (lines, planes, and arcs). The relations are attributed three-dimensional relationships between the entities. Lines can be pairwise parallel, groups of lines can lie in a plane, lines can be perpendicular, and planes can be parallel or perpendicular.

We assume that the search is over the possible three-dimensional tuples and that the attributes of the relations do not form a part of the search space. As described in Section III, the attributes are computed from known measurements and play a part in determining the consistency of the generated hypotheses.

Such a hypothesis based scheme has been implemented in full for analyzing perspective projections of polyhedral objects. Currently the system understands and can reason about straight lines and planes. Inference engines have been implemented which can reason about inter-entity relationships: parallel, perpendicular, collinearity, and lines in a common plane. The inference engines relate these relationships and their attributes to the camera position and orientation. The system can relate vanishing points of lines and the vanishing traces of planes and use them in computing the unknown camera position and orientation. For an example of the nature of the inference engines, see Figure 1.

The system has been exercised on perspective line drawings generated by a graphics system. Three-dimensional wireframe objects can be constructed and viewed from arbitrary positions with user specified camera focal lengths. The resulting line drawings serve as input to the inference system. The inference system computes from the input line drawing the relations between the lines in the image in the camera coordinate system and the unknown camera focal length. Current work is focused on extracting line drawings from images digitized by a camera and using the extracted lines as input to the inference process. An excellent source of perspective drawings is [14]. One example of the drawings in that book is shown in Figure 2. Our aim is to be able interpret drawings of that complexity.

All the perspective equations reported in [12] dealing with lines, planes and distances are being implemented. Work is currently underway to extend the knowledge encoded in the system to include conic arcs and curved surfaces and to include more high level and in some cases heuristic information dealing with junction labellings, skewed symmetry and stable physical configurations of three-dimensional solids. The system is implemented in an ex-

Inference engine 1:

Given the hypothesis that two lines are parallel and are whose images are not parallel, determine the vanishing point on the image as the intersection of the lines' projections.

Inference engine 2:

Given the hypothesis that two lines are coplanar, such that they do not have a common vanishing point, compute the vanishing trace (locus of the vanishing points of all lines in that plane) for the plane.

Inference engine 3:

Given a plane with a known vanishing point, and a line in the plane with an unknown vanishing point, compute the vanishing point of the line.

Figure 1: A sample of the types of inference engines currently encoded in the vision system.

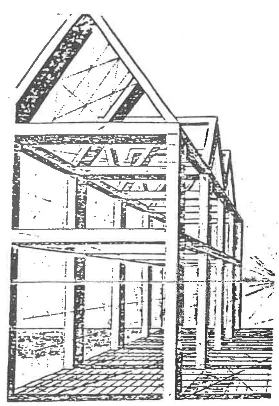

Figure 2: A sample perspective line drawing from [14].

tended PROLOG which has procedural as well as non-procedural statements and which has list, array and associative table data structures.

It is interesting to contrast the technique described in this paper with that of the ACRONYM system [15]. ACRONYM used symbolic reasoning to produced model driven interpretations for perspective images. The system described here does not use any form of object models and reasons based only on the known structure and constraints inherent in the projection process. Secondly, ACRONYM's mode of reasoning was essentially theorem proving over the domain of arithmetic and trigonometric inequalities. Our technique captures the knowledge of the domain in inference engines which have the required mathematical functions encoded in closed form and reasons over a space which is equivalent to that of the predicate calculus and is therefore much more efficient.

Bibliography

[1] Heath, T, "A History of Greek Mathematics", Volume II, Oxford University Press, 1921.

[2] Duhem P.M.M., "La Theorie Physique: Son Objet, Sa Structure" Translated to: "The Aim and Structure of Physical Theory", by Wiener, P.P., Princeton University Press, 1954.

[3] Baker S.F., "Induction and Hypothesis-A Study of the Logic of Confirmation", Cornell University Press, Ithaca, N.Y. 1957.

[4] Lederberg J., "DENDRAL-64, A System for Computer Construction, Enumeration and Notation of Organic Molecules as Tree Structures and Cyclic Graphs, Part I, Notational Algorithm for Tree Structures", Report No. CR-57029, NASA.

[5] Wesley L.P. and Hanson A.R., "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the VISION System", IEEE-PRIP 1982.

[6] Levine M. and Nazif, "Rule-Based Image Segmentation: A Dynamic Control Strategy Approach", TR-83-9, Computer Vision and Robotics Laboratory, Department of Electrical Engg., McGill University, 1983.

[7] Date C.J., "An Introduction to Database Systems" Third Edition, Addison-Wesley Publishing Company, 1981.

[8] Ullman J.R., Haralick R.M., and Shapiro L.G., "Computer Architecture for Solving Consistent Labeling Problems" Technical Report, Department of Computer Science, VPI&SU.

[9] Mulgaonkar, P.G., "Analysis of Perspective Line Drawings Using Hypothesis Based Reasoning". PhD dissertation, Department of Computer Science. 1984.

[10] Haralick, R.M., and Elliott G., "Increasing Tree Search Efficiency for Constraint Satisfaction Problems", Proc 6th. IJCAI, 1979.

[11] Mulgaonkar P.G., Shapiro L.G., and Haralick R.M., "Matching 'Sticks, Plates and Blobs' Objects using Geometric and Relational Constraints", Image and Vision Computing, 1984.

[12] Haralick R.M., "Using Perspective Transformations in Scene Analysis", CGIP 13, 1980.

[13] Rescher N.,"Hypothetical Reasoning", Amsterdam North Holland Publishing Company, 1964

[14] de Vries J.V.,"Perspective", Dover Publications, New York, 1968.

[15] Brooks R.A.,"Symbolic Reasoning among Three-Dimensional Models and Two-Dimensional Images", Artificial Intelligence, special volume on Computer Vision, AI 17, 1981.

[16] Reddy R., Erman L., Fennell R., and Neely R., "The Hearsay Speech Understanding System: An Example of the Recognition Process", IEEE Trans. Computers. C-25, 1976.

[17] Erman L.D., Hayes-Roth F., Lesser V.R., and Reddy D.R., "The Hearsay-II Speech Understanding System: Integrating Knowledge to resolve uncertainty", Computing Surveys, 12(2), 1980.