RSD-TR-4-87

A Qualitative Approach for Recovering Relative Depths in Dynamic Scenes

by

S. M. Haynes

Ramesh C. Jain

Department of Electrical Engineering Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

March 1987

CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING

Robot Systems Division
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109-1109

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 The Complexities In Vision 1
1.2 Motivations for Qualitative Approaches to Vision 3
1.3 Qualitative Descriptions and Processing 5

2. LOCAL TEMPORAL INFERENCING 7

2.1 Constraints on Domain and General Description 7
2.2 Temporally Local Inferencing on Qualitative Relations 13
2.2.1 Inconsistencies 15
2.2.2 Taking Advantage of Redundancy 17

3. CHRONOLOGIES 19

3.1 Purpose and use of Chronologies 19
3.2 Representational Issue-Indexing 21
3.3 History and World Model of Depths 22

4. EXPERIMENTS 23

4.1 Notational Comments 23
4.2 Restrictions on Relations and Confidences 23
4.3 A Sequence of Data 24

5. CONCLUSION, CONSEQUENCES AND NEXT STEPS 25

6. REFERENCES 26

RSD-TR-4-87

A Qualitative Approach for Recovering Relative Depths in Dynamic Scenes [1]

S. M. Haynes and Ramesh Jain

Robot Systems Division
Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109

ABSTRACT

The approach to dynamic scene analysis described in this paper is a qualitative one. It computes relative depths using very general rules. The depths calculated are qualitative in the sense that the only information obtained is which object is in front of which others. The motion is qualitative in the sense that the only required motion data is whether objects are moving toward or away from the camera. Reasoning, which takes into account the temporal character of the data and the scene, is qualitative. This approach to dynamic scene analysis can tolerate imprecise data because in dynamic scenes the data are redundant.

[1] We gratefully acknowledge the support of AFOSR F33615-85-C-5105.

1 Introduction

Interest in computational qualitative approaches has been driven by the complexity of problem solving in real-world domains. While various physical systems have received the lion's share of attention from those interested in qualitative reasoning, vision, a domain also requiring problem-solving in complex real world domains, appears to be amenable to qualitative approaches. For both physical systems and vision problems the representations are insufficiently rich. Mathematical models are not available for some problems; for other domains they may be available but analytic solutions are not feasible for a variety of reasons. Sometimes the problem statement itself is qualitative (e.g., "will the ball roll over the hump?" or "is region A closer to the viewer than region B?"), and there is little to be gained by forcing a quantitative statement of the problem, finding the solution (when possible), and converting back to a qualitatively expressed answer. Both physical and visual domains where there are temporal effects are especially recalcitrant to quantitative approaches.

1.1 The complexities in vision

For many reasons computer vision has proven to be far more difficult than was originally suspected. The real world is sampled spatially and temporally and projected onto a time-ordered sequence of frames. The task of computer vision is to provide a description of objects, relationships and events among those objects. This is a signal to symbol transformation which requires the top-down use of knowledge (i.e., an interface to a memory). Because the projection process "loses" a dimension, interpretation must be able to tolerate ambiguous data.

Noise compounds the ambiguity of vision data in the frame sequence. For most of the areas of vision which can be modelled mathematically, the equations are non-linear. Solution techniques to non-linear equations are very brittle, being easily swamped by noise and are highly sensitive to the starting values used in iterative algorithms. Roberts' work in this field ([23]) attempted to compensate for noise using several heuristics for line detection and a top-down model-fitting approach.

Two approaches to vision have proven to be particularly restrictive. The first is the focus on single frame analysis. Early researchers felt that it was necessary to first interpret one single frame. This is a sterile approach because it avoids all temporally changing scenes (e.g., things like pictures and maps), including most scenes of interest. The data in a single frame underconstrains the interpretation, and ambiguity and noise only make matters worse. Researchers were forced to turn to reliance on model-driven interpretation (e.g., [7]); but such a top-down approach does not work well all the way down to the data level.

The second approach which has disappointed is the careful computation of numerical features in a data driven manner. Examples are 3-D positions of feature points obtained via structure-from-motion or of surface normals from optical flow, shape from shading, or texture, motion parameters: v_x, v_y, v_z, and optimization of objective functions (examples of these various approaches: [1], [20], [22], [26], [29], [34]). Most rely on an inverse transformation, from two dimensions to three; that, combined with the noise inherent in sensor and the sampling and digitizing processes, means that algorithms providing quantitative solution will be inherently very sensitive to noise.

1.2 Motivations for qualitative approaches to vision

Along with the growing interest in dynamic scenes, has come the realization that improving accuracy in a restricted set of features does not particularly help the interpretation process. Some vision researchers [27] are being drawn to the qualitative approaches being used for common sense reasoning, naive physics and circuit analysis [6], [13]. Qualitative approaches show that it is possible to obtain meaningful results when solving problems with uncertain, approximate or only signs of parameters. Qualitative representations of the problem domains are an attempt to capture the fundamental nature of the system, while avoiding the complexity of dynamic equations.

One approach in computer vision for handling the noise in a bottom-up fashion is to use a variety of window sizes or a collection of band-passed images. Larger sized operators average over a greater area, and thus, for reasonably well behaved noise, the noise has less effect on the result. It is not clear how to combine information among these many channels, partly because the channels are being used for two different things: detecting (or measuring) at different scales, and using larger channels to reduce noise effects at the lower channels. Unfortunately, the larger the window, the more likely it is computing a single result over two or more qualitatively different pixel source populations. Event detection, in its most general sense, locates the interface between qualitatively different sources of pixel population. The idea of event detection is *not* to smooth over noise, and thus over different pixel populations by using an arbitrarily chosen window sizes, but instead to detect where the pixel population changes and avoid any integration across that boundary. Using this paradigm are [5], [12], and [14]; also [26], using a finite element approach, can fracture the surface at appropriate places.

This noise issue has especially frustrated dynamic scene researchers because it has been shown mathematically that all 3-D information (to a scale factor) is available in the optical flow field. Attempts to get the information have been fruitless because even the best obtainable flow fields are too badly corrupted. Thompson, et al., [28] take the approach that if precise values are not computable, then compute the qualitative information: which segment is the occluder and which is the occluded. Jain [15] has also obtained this information for different sorts of scenes. Both use only a crude, though computable, approximation to optical flow. The first uses an approximation to the flow field called a disparity field which requires good feature detection and correspondence algorithms. The second uses a more qualitative approximation, computing the time history of pixel changes. But in any case it is clear that useful results are possible, even from the noisy data available, using and computing qualitative attributes rather than precise, brittle ones.

Qualitative solutions can be used to constrain quantitative equations or starting points, or as data structures for focus of attention mechanism, or in planning.

Also, biological vision researchers have long been aware of qualitative visual characteristics, perceptual judgements. Color vision has received the greatest attention and a qualitative approach has been used to drive the experiments which eventually revealed the opponent process theory of color vision, [19].

1.3 Qualitative descriptions and processing

Qualitative representations of problem domains attempt to capture the fundamental nature of the system avoiding those characterizations which are brittle. Much of the work done in qualitative physics involves determining appropriate states and symbols and obtaining an understanding of the nature of state change. An important qualitative reasoning ability is the simulation process, allowing for a grasp on causality. The qualitative parameters thus far attempted have generally included things like *signs* of derivatives ([9], [17]) or transitions [11].

When values can be tied to a number line, they are quantitative. Permitting bounds on values, that is, labelling the range to intervals on the number line, one can still do numerical operations on them [3]. An example of dividing up the range of a variable into intervals on the number line is the *Sign* function:

$$Sign(x) = \begin{cases} -1 & \text{for} \quad x \,\varepsilon\, (-\infty, \epsilon^-) \\ 0 & \text{for} \quad x \,\varepsilon\, [\epsilon^-, \epsilon^+] \\ +1 & \text{for} \quad x \,\varepsilon\, (\epsilon^+, +\infty). \end{cases}$$

This *Sign* function is a typical qualitative parameter. The numerical intervals are given the labels **negative**, **zero** and **positive**, and reasoning is performed using these symbols. One must take care with such a large grain size because not much pruning of hypotheses (for those hypothesize-and-test control paradigms) is possible unless further constraints are included.

Naive physics makes heavy use of the *Sign* of a derivative: **increasing**, **decreasing** and **no change**. In the research to be described in a later section we also use the *Sign* function as a qualitative parameter.

The number line can be divided into other intervals. For example, a variable *loudness* may have intervals labelled **piano**, **mezzo-piano**, **mezzo-forte**, **forte**, **fortissimo**. The problems of effectively dividing up the number line into labelled intervals including things like locating endpoints, hysteresis effects, and "state transitions," thus far have been addressed only from a domain dependent viewpoint. The more abstract issues have not yet been addressed.

In any case, it is clear that this sort of qualitative descriptor, where the values are tied, at least initially, to a number line, is a subset of the standard AI symbolic descriptors.

Another sort of qualitative value is a *relative* statement. For example **x is faster than y**, or **x is closer than y**. This sort of relation constrains the value of x with respect to y (and vice versa), but does not tie the value to the number line. Hasse diagrams are a graphical representation describing such relative statements when the relation provides a partial ordering. The qualitative example involving partial order is different from commonly used notions of state and of symbol. It is a comparison. The ordering qualitative example is also more robust to noise. Shepard's classic paper [25] makes use of this ordering information.

There are other qualitative relations between attributes which are interval in their nature, i.e., which have begin and end points. Vilain [32] and Allen [4] have developed an interval-based temporal reasoning and labelling system. Their works, and those of others, are applicable to domains like story understanding, where there tend to be fixed endpoints to the temporal intervals. Vere [31] has developed a system which will generate parallel plans for achieving goals within time constraints.

2 Local Temporal Inferencing

2.1 Constraints on domain and general description

There are many cues which may be exploited to obtain at least approximate distances from the viewer to objects. Among these are various perspective effects (objects further away are smaller, texture gradients become tighter), and occlusion. When surface A occludes surface B, A is partially obscuring B. Unless there is evidence to the contrary, observers will assume that the occluding surface, A· is closer than the occluded surface B. Determining occluder-occluded pairs is very difficult for single frame analysis even with object models because of the ambiguity and the noise in the data. Single frame analysis is sufficiently challenged with just finding occluding boundaries ([8], [18]).

Computer vision researchers have discovered that motion helps an occlusion analysis. An example will demonstrate. In figure 1 alone it is not possible to determine the order of the occlusion between region A and region B. (Aside—this is just the famous vases-faces reversal effect) But suppose

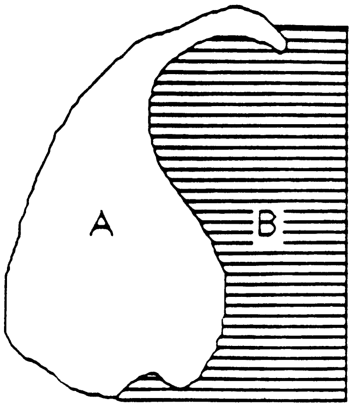


Figure 1: Frame i

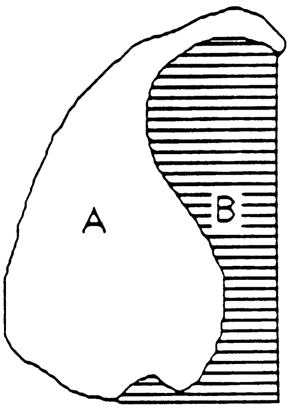


Figure 2: Frame $i+1$

one has both figure 1 and figure 2 where A has moved to the right. We perceive, unless there is strong evidence to counter the effect, that region A is the occluder, region B the occluded. Thompson, et al. [28] has show how to obtain occluder-occluded relations for regions with significant texture. Jain [15] has shown the same for uniform regions.

Figure 3 shows a typical viewer geometry. Imagine the image plane is at $z = 0$, that is having zero depth. Increasing depths have increasing z values. For perspective projection the focal point is at some $z = -d$, d being the focal length of the imaging system; for orthographic projection, the focal point is imagined at $z = -\infty$. In both cases, $z^A < z^B$ means the distance from the viewer to A is less than the distance from the viewer to B. We can ignore the viewer's position and concentrate on depths.

Consider obtaining occlusion relations for a fixed time when all objects are temporarily frozen. The implication A occludes $B \Rightarrow z^A < z^B$ holds

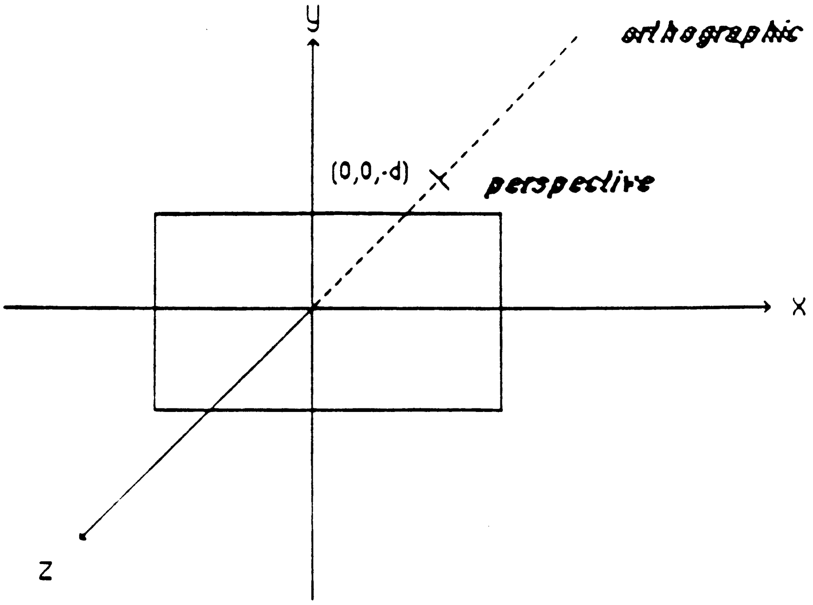


Figure 3: Projection onto the image plane

for many arrangements of surfaces and viewer. There is one important situation where it does not hold. The surface of an object is actually a function $z(x, y)$. If there is significant change in depth across a surface, then the occlusion to relative depth implication may not hold across the the entire surface. Thompson calls this the "boxtop" case [27]. For example consider the sketch in Figure 4. A occludes B on the image. Where the occlusion occurs the depth relation A is closer than B also holds. But because the surface of A has significant surface tilt, the surface of A in its entirety is not closer than the surface of B. We will not discuss this problem further in the paper. We assume the intra-surface depth change is not significant with respect to extra-surface depths.

Occlusion data thus places a partial ordering on the depths of surfaces;

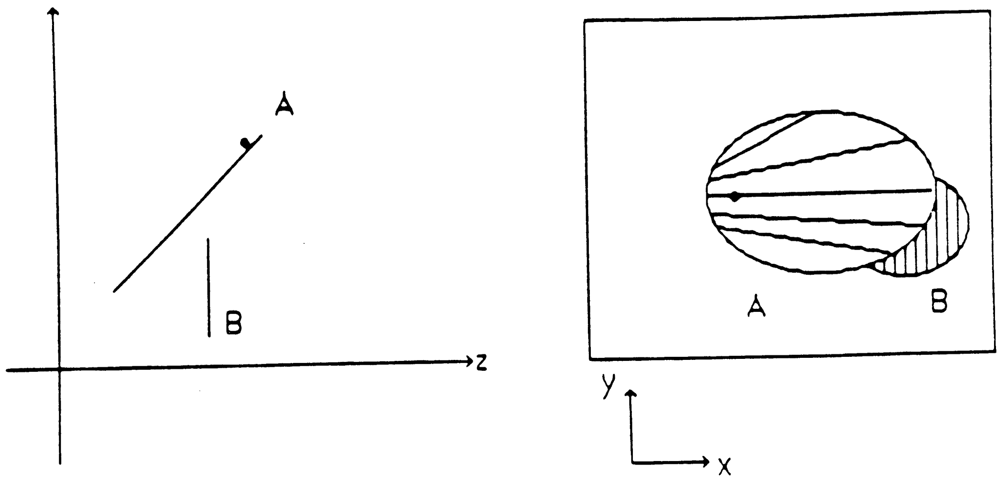


Figure 4: An arrangement with significant surface tilt

and when frozen in time, transitivity provides all computable depth constraints between surface patches. This partial ordering does not change over time. Thus, for example, given the data set:

$$z^A < z^B;\ z^B < z^C;\ z^B < z^D\ ,$$

transitivity gives:

$$z^A < z^C;\ z^A < z^D\ .$$

There is *no* ordering in depth between C and D. By transitivity we obtain relative depths between some surfaces which have not direct occlusion evidence for depth ordering. See Figure 5.

Now permit objects to move about freely in planes parallel to the image plane, that is, objects may not move in the z direction. In this case the rule for combining depth constraints is again transitivity. If there is no change in depths of objects then the relative depths will not change.

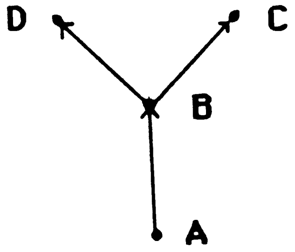


Figure 5: Directed graph for depth ordering

A more interesting situation occurs when objects are permitted to move toward or away from the observer. Temporal effects must be considered. While for a fixed time, the depth ordering obtained from local occlusion analysis and transitivity is valid, that order may not hold into future times. Transitivity does not hold into the future when depths change over time. The depth ordering depends on the motion in depth v. When there is no acceleration, a depth ordering depends on the sign of v. $Sign(v) = -1$ means motion is toward the observer. $Sign(v) = +1$ means motion is away from the observer. $Sign(v) = 0$ means there is no significant motion in depth.

Consider the problem from the following perspective: what is the relative depth between two surfaces A and B at future time t_{k+i} if we know the order at time t_k, $i > 0$. Considering only the sense of the z motion, there are 9 possible velocity combinations $v^A \epsilon \{-1,0,1\}; v^B \epsilon \{-1,0,+1\}$. The depth order at t_{k+1} can be revealed through a case analysis. For example:

$$z^A(t_k) < z^B(t_k)$$

v^A \ v^B	-1	0	+1
-1		A(j)<B(j)	A(j)<B(j)
0		A(j)<B(j)	A(j)<B(j)
+1			

for
A(i)<B(i);
i<j

Figure 6: Velocity rules

$$v^A = -1 \; ; \; v^B = 0$$

$$z^A(t_{k+i}) = z^A(t_k) + v^A \, i \; ; \; z^B(t_{k+i}) = z^B(t_k) + v^B \, i$$

$$z^A(t_{k+i}) < z^A(t_k) \; ; \; z^B(t_{k+i}) < z^B(t_k)$$

$$z^A(t_{k+i}) < z^A(t_k) < z^B(t_k) = z^B(t_{k+i})$$

$$z^A(t_{k+i}) < z^B(t_{k+i})$$

There are four cases where predictions can be made about future depth ordering using only sense of velocity. These are shown in figure 6. The other 5 cases require more constraints on the velocities than only sense of motion in depth in order to make statements about future depth order. These four cases are formulated as rules for combining depth relations and velocities. Naturally, they are applicable on those temporal intervals where the velocities do not change. The four rules are:

- *rule* 1 : $z^A(t) < z^B(t)$; $v^A(t, t + \Delta t) = 0$; $v^B(t, t + \Delta t) = 0 \Longrightarrow$ $z^A(t + \Delta t) < z^B(t + \Delta t)$
- *rule* 2 : $z^A(t) < z^B(t)$; $v^A(t, t + \Delta t) < 0$; $v^B(t, t + \Delta t) = 0 \Longrightarrow$ $z^A(t + \Delta t) < z^B(t + \Delta t)$
- *rule* 3 : $z^A(t) < z^B(t)$; $v^A(t, t + \Delta t) = 0$; $v^B(t, t + \Delta t) > 0 \Longrightarrow$ $z^A(t + \Delta t) < z^B(t + \Delta t)$
- *rule* 4 : $z^A(t) < z^B(t)$; $v^A(t, t + \Delta t) < 0$; $v^B(t, t + \Delta t) > 0 \Longrightarrow$ $z^A(t + \Delta t) < z^B(t + \Delta t)$

It is interesting that while we require knowledge of motion to and from the observer for computational reasons, there is considerable biological evidence for such "detectors," which are independent of detectors for image plane motion ([21]).

2.2 Temporally local inferencing on qualitative relations

For the work reported here the data are time-ordered lists of occluder-occluded pairs and directions of motion in depth of surfaces (toward or away from camera). We wish the program to provide the relative depths among surfaces, when computable, and histories of surface to surface relations without recourse to object or scene models, over extended frame sequences.

Transitivity works on a fixed time instant. The velocity rules project into the future. The velocity rules are expressed using intervals over which the velocity direction is fixed. Rather than becoming involved in a calculus over intervals, we take advantage of the nature of the data: data arrives, in order, at discrete times. We use the velocity rules to make inferences into the future for one time step only; a time step is the time the next datum is available. Thus, this system incrementally incorporates the data as it becomes available. It makes no attempt to predict further into the future than to the next time step. It is *local*, temporally speaking. More global temporal knowledge is kept elsewhere in the system – specifically, in the object histories.

We have three operations which provide depth ordering relations for a given time:

1. local occlusion analysis,
2. transitivity on depth relations,
3. velocity rules.

Figure 7 gives a layout of the order in which these operations are applied.

Thus, at time t_0, we have a set of depth relations $\Psi(t_0)$. From these relations we use the velocity rules to derive, for the next time t_1, a set of relations $\Psi^*(t_1)$. This set of relations, ignoring time and labels, will be a subset of the relations at t_0. Incorporating the data at time t_1 will add some new relations. This will give rise to the set of relations $\Psi^{**}(t_1)$. Now one applies transitivity at this point to obtain the set $\Psi(t_1)$, and the set of relations is ready to project into the future one time step again. Transitivity is not applied following the velocity rules because it provides no new relations.

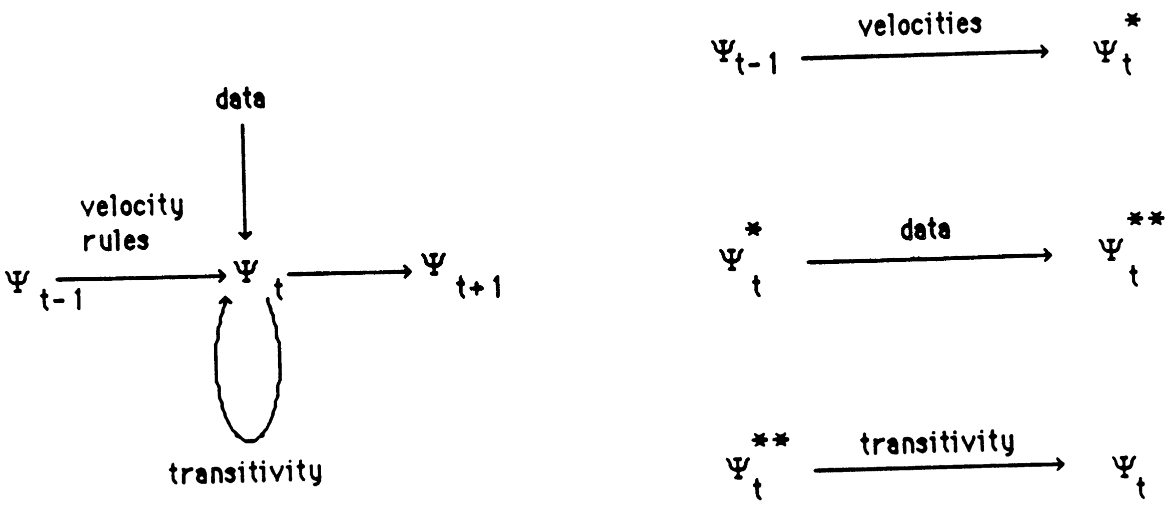


Figure 7: The collections of relations (Ψ system)

When an operation derives an order, that order is labelled with the operation. There may be several relations between a given pair of segments. That is, each relation has an ordering (i.e., two segments) and a label. For the segment pair, A and B, we may have, e.g.,

order		label
$z^A < z^B$	;	rule 1
$z^A < z^B$	;	data

As long as the data are consistent and correct, these inferences will iteratively build a consistent partial order on the segments which is as complete as is possible for these rules at the current time.

2.2.1 Inconsistencies

What happens when a datum is incorrect? In that case we will have an *inconsistent* set of relations. This inconsistency is signalled by a *cycle* in the directed graph giving the depth ordering. For example, suppose we have the relation: $z^A < z^B$: rule 1 for the graph in $\Psi^*(t)$. We then read the datum $B < A$. The graph will then contains the cycle $z^A \rightleftharpoons z^B$.

Because we have labels on relations, we know what gave rise to the inconsistency. For the above example we know that, because there is a cycle, the datum $B < A$ is wrong, or the rule 1 applications was wrong or both. Conceivably, we could trace the cause of the inconsistency back further into the past. For the above example, if the rule 1 application at time $t-1$ was wrong then either the v^A was wrong, v^B was wrong, the relation $z^A < z^B$ at time $t-1$ was wrong, or any subset of these three was wrong. For this one inconsistency involving only two objects and two relations we have already fingered as possible culprits four attributes or relations going back only one time step. Indeed, if we kept only a slightly more complete audit trail the relation $z^A < z^B$ at time $t-1$ could be further tracked down. This gives rise to even more possibilities of the source of inconsistency even more remotely in time.

We are not doing this for a number of reasons. The most important of these is than in a dynamic scene understanding system, one does not have the resources to spend a lot of time and energy resolving past conflict; data are continually arriving, and it is better to have the current (and future) interpretations be correct than those of the past. Secondly, many culprits are fingered for each inconsistency. This is primarily because transitivity in this case is too general, it suffers from the same flaws as do weak methods in AI. Because of the large numbers of dubious relations, there is a lot of overhead. The inconsistencies cannot be resolved or reduced from the information available to the system (unlike e.g., story understanding systems). For a third reason, resolution is possible only in the future when more data is available – there is no resolution possible in the past (where the inconsistency arose). Fourth is that we rely on the fact that there are a lot of data. Even though some are wrong, most will be right; we do not want to devote much effort to inconsistency resolution because we may expect that future data will set things right. There is one important consequence of this for the implementation: we do not keep an extensive audit trail. We label each arc with only the rule that most recently derived it.

2.2.2 Taking advantage of redundancy

So we do not make any attempt to undo any bad effects from possible bad data in the past. We want the correct data to eventually outweigh any incorrect inferences. There are a number of options on how to go about doing this. Essentially there are two questions:

1. how to propagate, to the next time step a relation which has a contradiction,

2. how to incorporate a contradicting pair of relations into object histories.

We deal with this difficulty by taking the position that one must trust the current data at the current time. Any inferences from that data, especially into the future may be suspect, but the data themselves are assumed to be correct for the time now. We could take the position that any relation which contradicts data will be deleted immediately, and is not allowed to propagate into the future. But we still have the problem of contradictory relations which do not have data on either of the labels.

We are drawn to the use of a certainty factor for each relation in order to accomodate the possibility of occasionally invalid data. The certainty of data relations will be highest. As inferences are derived, the certainty factor of those relations will decrease.

There are a number of technical difficulties involved in dealing with certainty factors and getting them to be rigorously correct. We avoid these by relying on the fact of large amounts of mostly right data. To rigorously derive a calculus of certainty factors, it is necessary to have a sufficiently deep understanding of the nature of the domain. Such an understanding includes the nature of the data, noise, and *a priori* probability values of the data. Generally restrictive independence or correlations assumptions are required. In the spirit of qualitative processing, we wish to avoid making such stringent assumptions until necessary. In this system we are attempting a qualitative approach in which we know there is noise, though we don't know its precise precise properties.

Because of the fortunate choice of using dynamic scenes as data, however, we can use the fact that the data will be mostly redundant. Thus even though rigorous derivations for certainties have been done (e.g., [24]), and may be applicable, we are not currently investigating that direction. We just want certainty factors to decrease with time and with transitive "distance" from *data*. There are a number of choices on how to combine certainty factors when making inferences. We are currently experimenting with this.

We deal heuristically with the problem of which of two contradictory relations to propagate. Relations which are contradictory are not permitted to activate the transitive rule. All other relations may activate both transitive and velocity rules. We put this restriction on transitive-derived contradictory rules for computational reasons, because, as mentioned earlier, transitivity is too general. Many relations are derived using transitivity, and when one of the links is suspect, all links derived from it are suspect. Inferences whose certainties are decreasing to zero are deleted after a fixed temporal interval.

There is no normalizing of confidences necessary because aa confidence on a particular relation can only decrease. A confidence starts at 1.0 for data relations and decays with each temporally or spatially based inference.

3 Chronologies

3.1 Purpose and use of chronologies

The storage and use of velocities and positions over time is required. It is not enough to give a simple initial state and equation of motion because most motions are not describable with simple dynamic equations (consider hierarchical or non-rigid motions). Such representations also do not incorporate changes in motion descriptions easily; also, other interesting temporal characteristics are not included in a natural fashion. Early works in computer vision did not keep chronologies. Instead, in [2] for example, they kept a model of the scene for one time instant only, and used that to predict the model for the next frame. This implicitly incorporates the initial state description. Robot planning frequently requires the description of several actions over an extended period of time. These are generally inspired from the approach of describing the state of the world and robot at each time instant ([10]).

Tsotsos [30] made extensive use of chronologies. These were time-ordered positions of points. They were used to choose among hypotheses for high level motion descriptions (e.g. **expand, sway**). His system chose the best hypothesis by examining the time-course of confidences of the possible schemas. This example exemplifies a major use of chronologies: to disambiguate local motions into more global, longer term motion descriptions. Other uses of chronologies are to be able to predict future positions and circumstances, to identify interesting motions, and to localize events in the motions. In addition to obtaining long term motion descriptions, a history of events, motions, and relationships between object parts, is useful. Such a history is useful on its own merits, as well as for deriving higher level descriptions. That is, one may be able to describe oscillatory motion as such, rather than as a repeating sequence of position and velocity.

Chronologies are not really models, however, because they provide neither a simplified representation for the data, nor an understanding of relationships. They provide a description. Chronologies also provide a representation in which noise tolerance, occlusion and integration of data in a temporal fashion can be supported, especially under the control of temporally dependent operation.

3.2 Representational issue—indexing

In building a chronology of depth-ordered relations among surface patches for use, either as a descriptive device, or as an intermediate data structure for further processing, there are two ways of indexing.

The first is to organize the relations *temporally.* In dynamic scene analysis the data are arriving in a time-ordered fashion, e.g., in frame i, there is some set $\Psi(i)$ of relations, in frame $i+1$ some other set $\Psi(i+1)$. The chronology of relations when temporally indexed has the same appearance as the data. In this case it is easy to see what is happening at a particular time instant, because time is the index into list of relations, e.g.,

time	*relations*
1	$(z^A < z^B)\,(z^A < z^D)$
2	$(z^A < z^B)\,(z^A < z^D)$
3	$(z^A < z^B)\,(z^A < z^D)$
4	$(z^A < z^B)\,(z^B < z^C)\,(z^A < z^C)$
...	...

We see in one indexing step which relations exist at t=3. Given the way the velocity rules are formulated, it is also easy to make predictions into the next time step. For example, if the motions in depth of z^A, z^B, z^C are negligible, then at time 4 we can make the prediction that at time 5, the following relations will hold: $(z^A < z^B), (z^B < z^C), (z^A < z^C)$.

The second indexing method is to organize by relation. For example:

relation	*temporal intervals*
$z^A < z^B$	(1, now)
$z^A < z^C$	(4, now)
$z^A < z^D$	(1, 3)
$z^B < z^C$	(4, now)
...	...

If relations persist, or are repetitious, then it saves on space to index by relations rather than by time. This representation trades off relationship storage for temporal storage space advantageously when relations are long-lived or recur frequently. To determine at a particular time instant which relations are active requires inspection of a lot of data. The time course of relations is easier to access. Event marking makes deriving an interval-based description easy. And this method of indexing is better for dealing with noise removal, occlusion and integration over time.

3.3 History and world model of depths

Despite the fact that dynamic vision has a lot of data available, thanks to its *rampant redundancy* [33], it is both more efficient as well as satisfying to keep histories of relations which are indexed by surface. The fact remains, however, that in order to make derivations (or predictions) using the velocity rules and to be able to make a computationally fast statement about the relative depths at time now, we keep the current list of relations, though redundant with histories. That is, for now we have a "temporally indexed" set of relations. For now as well as all the past we have object-indexed relations. This means that if relative depths for any past time is desired, though the information is calculable (indeed, was calculated, then discarded), from the histories, it is not immediate. Rather, the system will have to step through the histories, essentially re-creating the world for the desired time. There are certain similarities with *envisioning* [9]. For now we can get an ordered relative depth map by doing a straight-forward topological sort [16].

4 Experiments

4.1 Notational comments

Relations have attached labels and confidences. $A < B$ is used to describe the depth ordering $z^A < z^B$. It is also used to describe the occlusion relationship A occludes B. This is for the sake of convenience. Confidences are given as $cf : x$ where x is some number, $0 < x \leq 1.0$. Thus the expression $A < B; cf : .82$ means the depth order $z^A < z^B$, with confidence .82.

4.2 Restrictions on relations and confidences

The local temporal inferencing system was implemented as described in a previous section. We made a few adjustments, for pruning purpose, to the inferencing procedure.

- As mentioned earlier, relations which were contradictory, e.g. $A < B$ and $B < A$, were not permitted to participate in any transitivity inferences. This is because the only result from applying transitivity on contradictory relations is *many* more contradictory relations. Contradictory relations are allowed to propagate into the future with the velocity rules.

- Only relations derived from transitivity which had a larger or equal confidence factors than other relations already present (between the same nodes) were posted. For example, suppose we have the relation $A < B; cf : .70$, then we derive $A < B; cf : .35$ from transitivity. That new relation is ignored.

- Data relations are taken with confidence 1.0 (indicated as $cf : 1.0$), and other relations already present between the same nodes as the data relation were deleted. That is, if we have $A < B; rule\, 1; cf : .9$, then we read the datum $A < B$, the previous rule 1 edge is deleted, only the data edge remains.

Recall that the numerical value of confidences count only in how they contribute to a decay in order to choose between inconsistent relations and delete spurious one. We did not perform theoretically rigorous derivations of confidence factors. Confidences propagated with velocity rules have a "time-decay" built in. That is, for $A < B; cf : x$ at time t_0, with appropriate velocities, then we can derive $A < B; cf : x'$, for time t_1, where $x' < x$. Confidence factors are combined for transitivity rules by taking the *min* of all the relations involved, then applying a decay factor (called a "spatial decay") to the resulting number.

4.3 A sequence of data

We present the results of an experiment in the series of figures 8 – 13). The input is echoed in the "input-list" window. The "active-relations" window contains a list of edges with the label and the confidence.

In figure 8 the data is only the three v motions of the objects A, B, and C. In figure 9, the data are $A < C$ and $B < A$, the transitive closure is performed which derives $B < C$. Data relations have confidences of 1.0. Transitive relations, labelled tc, are decayed.

In figure 10, the same three relations remain, because the velocity rule

rule 1 infer them. The confidences have all decreased because of the "temporal decay" on confidences. Notice the confidence for the relation $B < C$, the relation originally derived through transitivity, is less than that of the other two relations originally derived from data.

In figure 11, we have the new datum $D < A$; note the new relation has confidence 1.0. In addition, we have derived through transitivity the new relation $D < C$. In this figure notice that $B < C$ actually has two edges. One is a velocity rule edge with confidence 0.6, derived from previous $B < C$ edge. The other is a transitivity edge derived from the edges you see present, $B < A; cf$: 0.8 and $A < C; cf$: 0.8. This did not happen in figure 10 at time 8 because of the nature of the spatial and temporal decay factors. For this experiment, the spatial decay is larger than the temporal decay.

Figure 12 has the contradictory relation $A < D$ just read in. In figure 13, only the maximum relation of the contradictory relations is printed out. The old $A < D$ because of rule 1 is not drawn, though it will be propagated into the future. There are other ways of choosing a set of relations without cycles. For example one may add up the confidences on $A < B$ relations and on $B < A$ relations, then choose the maximum of the two.

5 Conclusion, consequences and next steps

In this paper we have described a dynamic scene analysis system which uses qualitative information, available with current computer vision abilities, to calculate relative depths between surfaces. The qualitative information required are motion toward or away from observer and occluder-occludee ordering. The system derives further relations from the data. Errors and inconsistencies are tolerated by requiring confidence factors to decay on each inference step. We have also described in this paper our approach to representing histories of such qualitative values.

This research has opened a number of questions. Among the more important is the problem of using such qualitative calculations as control for other computer vision processes, or as intial estimations for those iterative algorithms requiring them. We see this qualitative assessment as capable of providing a focus of attention when resources are limited and for making real-time dynamic scene analysis possible. The integration of qualitative procedures with numerical ones is an interesting problem.

References

[1] Adiv, G. "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans on Patt. Anal. Machine Intell., PAMI-7*, 1985, 384-401.

[2] Aggarwal, J.K. and R.O. Duda, "Computer analysis of moving polygonal images," *IEEE Trans on Computers, C-24*, #10, October 1975, 966-76.

[3] Alefeld, G. and J. Herzberger, *Introduction to interval computations*, Academic Press: New York, 1983.

[4] Allen, James F., "Maintaining knowledge about temporal intervals", *Communications of the ACM, 26*, #11, November 1983, 832-843.

[5] Besl, Paul and Ramesh Jain, "Segmentation through symbolic surface descriptions", *Proceedings* CVPR, 1986.

[6] Bobrow, Daniel G. (Editor), *Qualitative reasoning about physical systems*, MIT Press: Cambridge, MA, 1985.

[7] Brooks, R.A., "Model-based three-dimensional interpretations of two-dimensional images", *IEEE Transactions on Patt Anal and Machine Intell, PAMI-5* #2, March 1983, 140-150.

[8] Canny, John, "A variational approach to edge detection", *Proceedings* National Conference on Artificial Intelligence, *AAAI-83*, 1983, 54-57.

[9] de Kleer, J. and J.S. Brown, "A qualitative physics based on confluences," in [6].

[10] Fikes, R.E., P.E. Hart and N.J. Nilsson, "Learning and executing generalized robot plans", *Artificial Intelligence, 3*, 1972, 251-288.

[11] Forbus, K., "Qualitative reasoning about space and motion," in D. Gentner and A. Stevens (Eds), *Mental Models*, Erlbaum: Hillsdale, NJ, 1983.

[12] Grimson, W.Eric L. and Theo Pavlidis, "Discontinuity detection for visual surface reconstruction" *Computer Vision, Graphics, and Image Processing, 30*, 1985, 316-330.

[13] Hayes, P.J., "The naive physics manifesto" in D. Michie (Ed), *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press: Edinburgh, 1979.

[14] Haynes, S.M. and R. Jain "Event detection and correspondence" *Optical Engineering, 25*, #3, March 1986, 387-393.

[15] Jain, Ramesh, "Dynamic scene analysis using pixel-based processes", *Computer*, August 1981, 12-18.

[16] Knuth, D.E., *The Art of Computer Programming, III*, Addison-Wesley: Reading, MA, 1973.

[17] Kuipers, B., "Commonsense reasoning about causality: deriving behavior from structure," in [6]

[18] Marr, David and Ellen Hildreth "Theory of edge detection", *Proc Royal Soc B, 207*, 1980, 187-217.

[19] Miller, George A. and Philip N. Johnson-Laird, *Language and Perception*, Harvard University Press:Cambridge MA, 1976.

[20] Prazdny, K., "Egomotion and relative depth map from optical flow," *Biological Cybernetics, 36*, 1980, 87-102.

[21] Regan, D., K. Beverley, M. Cynader, "The visual perception of motion in depth," *Scientific American, 241*, July 1979, 136-151.

[22] Roach, J.W. and J.K.Aggarwal, "Computer tracking of objects moving in space", *IEEE Transactions on Patt Anal and Machine Intell, PAMI-2* #2, April 1979, 127-135.

[23] Roberts, L.G. "Machine perception of three-dimensional solids", in *Optical and Electro-optical Information Processing*, Eds: J.T. Tippett,

et.al., 1965, 159-197.

[24] Shafer, G., *A Mathemetical Theory of Evidence*, Princeton University Press: Princeton, NJ, 1976.

[25] Shepard, Roger N., "The analysis of proximities: multidimensional scaling with an unknown distance function. I." *Psychometrica, 27*, #2, June 1962, 125-140.

[26] Terzopolous, Demetri, "Image analysis using multigrid relaxation methods", *IEEE Trans on Patt Anal and Machine Intell, PAMI-8*, #2, March 1986, 129-139.

[27] Thompson, W.B., "Inexact vision," *Proceedings* IEEE Workshop on Motion: Representation and analysis, May 1986, 15-22.

[28] Thompson, W.B., K.M. Mutch, and V.A. Berzins "Dynamic occlusion analysis in optical flow fields", *IEEE Transactions on Patt Anal and Machine Intell, PAMI-7*, #4, July 1985, 374-383.

[29] Tsai R.Y. and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans on Patt Anal and Machine Intell, PAMI-6*, 1984, 13-27.

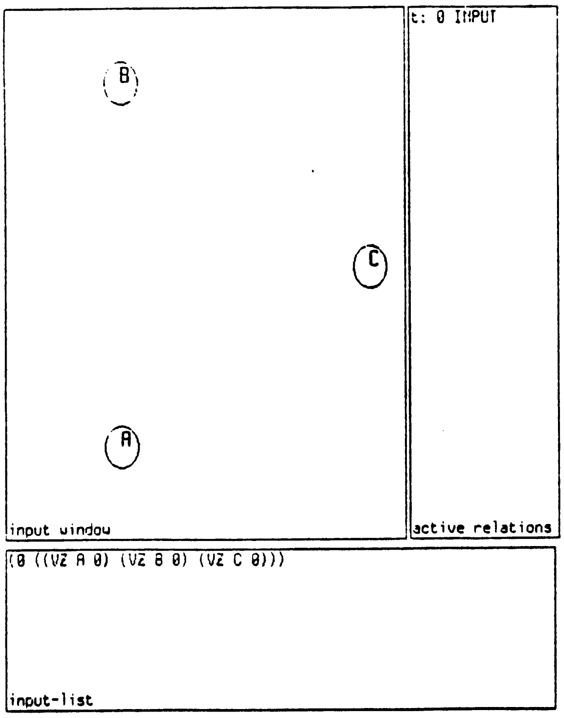
[30] Tsotsos, John, "A framework for visual motion understanding", Technical Report CSRG-114, University of Toronto, June 1980.

[31] Vere, Steven A., "Planning in time: windows and durations for activities and goals", *IEEE Trans on Patt Anal and Machine Intell, PAMI-5*, #3, May 1983, 246-267.

[32] Vilain, Marc B., "A system for reasoning about time", *Proceedings:* National Conference on Artificial Intelligence, AAAI, August 1982, 197-201.

[33] Terry Weymouth, Personal communication.

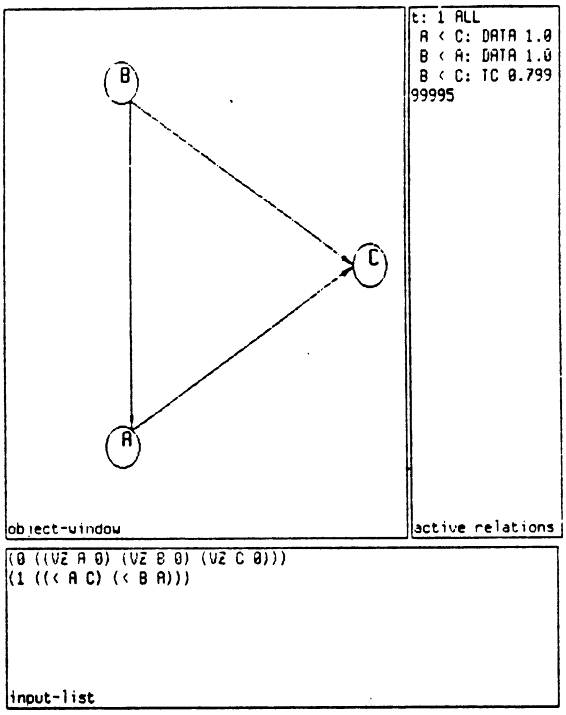
[34] Witkin, Andrew P. "Recovering surface shape and orientation from texture", *Artificial Intell., 17*, 1981, 17-45.



```
t: 0 INPUT
```

```
(0 ((VZ A 0) (VZ B 0) (VZ C 0)))
```

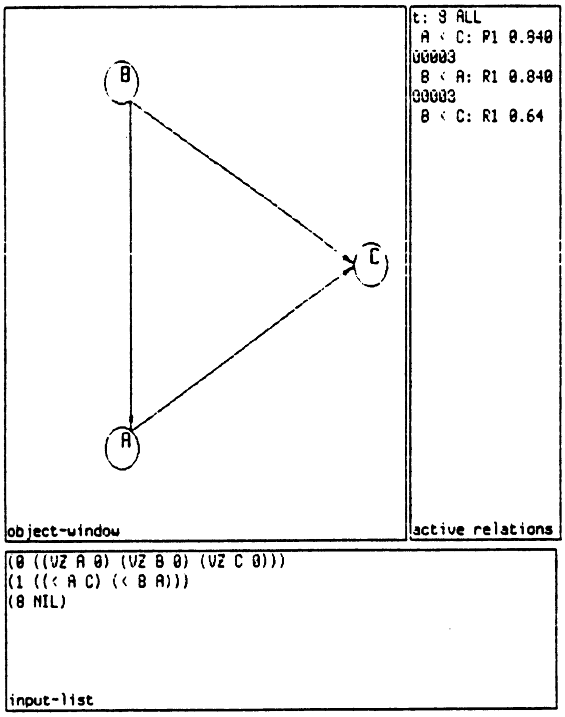
Figure 8



```
t: 1 ALL
 A < C: DATA 1.0
 B < A: DATA 1.0
 B < C: TC 0.799
99995
```

```
(0 ((VZ A 0) (VZ B 0) (VZ C 0)))
(1 ((< A C) (< B A)))
```

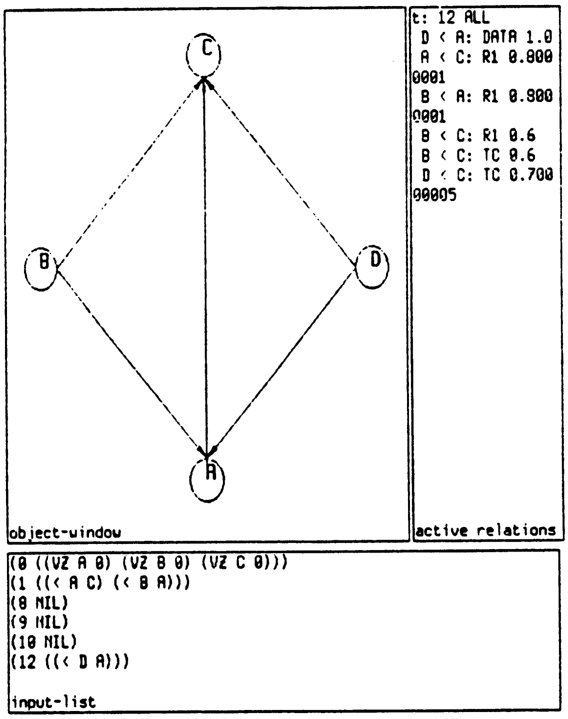
Figure 9



```
t: 9 ALL
 A < C: P1 0.840
00003
 B < A: R1 0.840
00003
 B < C: R1 0.64
```

```
(0 ((VZ A 0) (VZ B 0) (VZ C 0)))
(1 ((< A C) (< B A)))
(8 NIL)
```

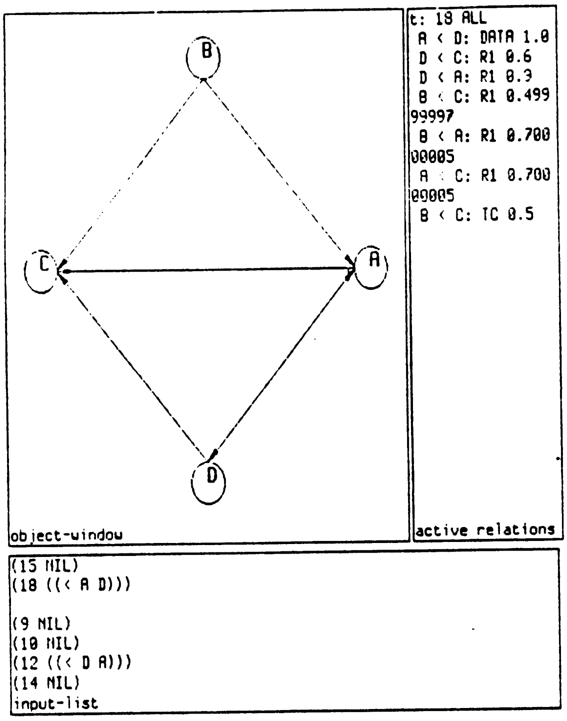
Figure 10



```
t: 12 ALL
 D < A: DATA 1.0
 A < C: R1 0.800
0001
 B < A: R1 0.900
0001
 B < C: R1 0.6
 B < C: TC 0.6
 D < C: TC 0.700
99005
```

```
(0 ((VZ A 0) (VZ B 0) (VZ C 0)))
(1 ((< A C) (< B A)))
(8 NIL)
(9 NIL)
(10 NIL)
(12 ((< D A)))
```

Figure 11

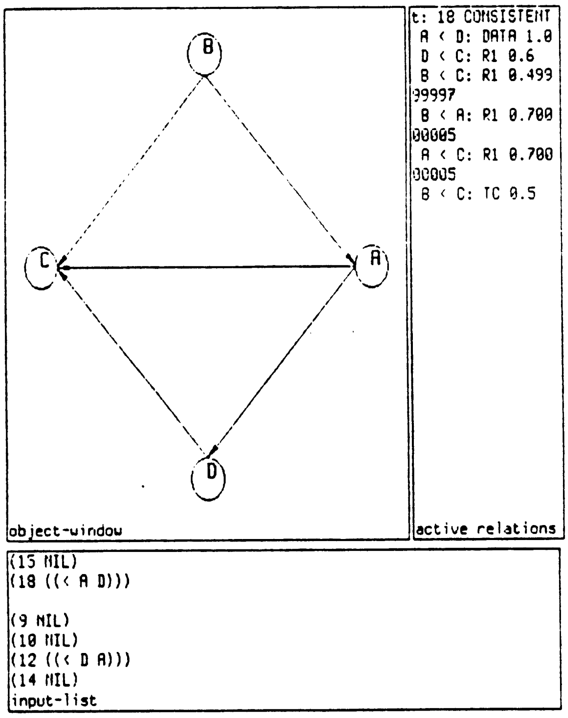


```
t: 18 ALL
 A < D: DATA 1.0
 D < C: R1 0.6
 D < A: R1 0.3
 B < C: R1 0.499
99997
 B < A: R1 0.700
00005
 A < C: R1 0.700
00005
 B < C: TC 0.5
active relations
```

```
(15 NIL)
(18 ((< A D)))

(9 NIL)
(10 NIL)
(12 ((< D A)))
(14 NIL)
input-list
```

Figure 12



```
t: 18 CONSISTENT
 A < D: DATA 1.0
 D < C: R1 0.6
 B < C: R1 0.499
99997
 B < A: R1 0.700
00005
 A < C: R1 0.700
00005
 B < C: TC 0.5
active relations
```

```
(15 NIL)
(18 ((< A D)))

(9 NIL)
(10 NIL)
(12 ((< D A)))
(14 NIL)
input-list
```

Figure 13