536-61
78

MX 2 ...

A Qualitative Approach for Recovering Relative Depths in Dynamic Scenes

S.M. Haynes and R. Jain
University of Michigan
Ann Arbor, MI 48109

Abstract

This approach to dynamic scene analysis 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.

Keywords: qualitative vision, dynamic scene analysis, relative surface depths.

1 Motivations for qualitative vision

For many reasons computer vision has proven a difficult task, far more difficult than was originally suspected. The complexity of the real world is sampled spatially and temporally and projected onto a time ordered sequence of frames. The description of objects, relationships and events among those objects 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 the frame sequence. Roberts' work in this field, [22], attempted to compensate for noise using several heuristics for line detection and a top-down model-fitting approach. An approach to solving this problem of model-fitting under noise is given by Brooks [8]. In this work, he has the problem of unknown transforms between model and image; to solve this problem he uses a constraint manipulation package to limit the matches between image and model and hypothesize other matches consistent with the constraints. The transforms are initially under-specified, then increasingly constrained.

Two approaches have proven to be particularly restrictive. The first is the focus on single frame analysis. Early researchers felt that it was necessary to first process one frame, and only then examine the subsequent frame. This is a sterile approach because it avoids all temporally changing scenes (i.e., things like pictures and maps), including most scenes of interest. 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 (for citations describing these various approaches see: [1], [20], [24], [27], [32]). Most rely on an inverse transformation, from two dimensions to three, and 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.

Along with the growing interest in dynamic scenes, the realization that improving accuracy in a highly restricted set of features does not particularly help the interpretation process some vision researchers [25] are being drawn to the qualitative approaches being used for common sense reasoning, naive physics and circuit analysis [6], [12]. The reasons for this are that qualitative approaches show that it is possible to obtain useful results when solving problems with uncertain, approximate or only signs of parameters. The representations of the problem domains are an attempt to capture the fundamental nature of the system, while avoiding the complexity of dynamic equations. Currently, much of the work done in qualitative physics involves determining appropriate states and symbols and an understanding of the nature of state change. Another important component is a simulation process, which allows one to get a grasp on causality. The qualitative approaches thus far attempted in AI have generally included things like signs of derivatives [9] [16] or transitions [10]. Computer vision is also a testbed where considerable intelligence is required. Further, the data in computer vision are always noisy, frequently redundant, and often misleading.

Another approach for handling the noise for bottom-up processes, having support from biological vision systems, 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. Unfortunately, the larger the window, the more likely it is computing a single result over two or more different pixel source populations. Some researchers have proposed using a set of different window sizes, large ones for large scale and perhaps low contrast changes and smaller ones for more local changes [17], the scale space is a continuous version of this [32]. 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.

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 a magically chosen window size, but instead to detect where the pixel population changes and avoid any integration across that boundary. Using this paradigm are [5], [11], and [13]; also [24], 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., [26] 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 [14] 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 requiring 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.

2 Using models in computer vision

There are a number of reasons for building or using a model in computer vision.

1. A model provides a simplified representation. For example, the motion of a point may be specified with initial state and state transition equations, thus it is not necessary to store, for each point in time, the position of a point.

2. At some, perhaps low, level, the data can be said to be understood when there is a model which fits them. For example, at a very low level when the data fit, e.g., a straight line, the line is a model for the data, and as a line is understood, so is the underlying phenomenon giving rise to the data. This is like number 1 above. At a higher level, for the model which is a line $y = mx + b$, one can say e.g., m is velocity and b is starting position. At the highest level, if the model is a frame with slots, then the general practice is to use *a priori* default values for slots which are not filled from the data. Thus, a limited data set which cause a particular frame to be instantiated, triggers a top-down use of the model in which more is understood than can be derived in an immediate sense from the data.

3. If the data can be said to fit some model, then, because in general, each data point need not be kept around, the interpretation processes can be made more tolerant to noise. In the representation, data can be allowed to lie within a range of values, as signal + noise, where the noise has some understood or assumed statistical properties.

4. Missing or ambiguous data can be handled by assuming the data exist according to the model, but are not measurable for some reason. This property of models is heavily relied upon by computer vision researchers because occlusion is rampant.

5. When a model is available, it can play the role of a kind of short term memory where the integration of data, especially of errorful data can be incorporated. This property was used in [2] in their very early work on dynamic scene analysis, in a primitive fashion.

There are a number of representation techniques for models. Perhaps the most popular is to use frames having slots [18]. A labelled frame will have a number of labelled slots which can be filled with numbers, attributes (i.e. symbols), variables or links to other frames. The procedures which manipulate these slots, for example how the slot with label velocity relates to dynamical equations which can predict future positions are also part of the modelling process. The dynamical equations are modelling the motion. These sorts of calculations are not as easily represented in frames, since this sort of knowledge is more naturally given as procedures. Interpretation involves instantiating the model from among a set of competitors that best match the data. An event indicates where models, or perhaps only parameters of the models, change. Confidence is a number expressing some kind of probability that the model is correct or that the data measurement is correct, when that information is available.

3 Purpose and use of chronologies

The very earliest works in dynamic scene analysis required the representation of velocities and positions over time. It is not enough to give a simple initial state, because most motions are not describable with simple dynamic equations (consider hierarchical or non-rigid motions), and because they do not incorporate changes in motion descriptions, and because other interesting temporal characteristics are not included in a natural fashion. The early works 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 (for an advanced use of this see [7]).

Tsotsos [28] made extensive use of chronologies which were essentially time-ordered positions of points 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 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 or of motions, or of relationships between object parts, is useful on its own, or for deriving other, even 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 neither provide a simplified representation for the data, nor do they provide understanding. 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.

4 Local Temporal Inferencing

4.1 Introduction

When values can be tied to a number line, they are quantitative. Permitting bounds on values, that is, restricting them to an interval on the number line, one can still do numerical operations on them [3]. Naive physics researchers use the qualitative (symbolic) descriptors: increasing or decreasing. These values are obtained by considering the sign of derivatives, also an interval. If the sign is positive, the variable values are increasing. We also use the intervals $(-\infty, 0^-)$, $(0^-, 0^+)$, $(0^+, +\infty)$ as qualitative values. Another sort of qualitative value is a *relative* statement. For example x is faster than y, or a is closer than b. 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 the notions of state and of symbol. It is a comparison. The ordering qualitative example is also more robust to noise - though not because the error tolerance is greater.

There are other qualitative relations between attributes which are interval in their nature, i.e., which have begin and end points. Vilain [30] 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 [29] has developed a system which will generate parallel plans for achieving goals within time constraints.

4.2 Constraints on domain and general description

For the work reported here we wish to describe the relations among objects, without recourse to object or scene models, over extended frame sequences. In particular, we wish the program to provide the relative depths among surfaces, when computable, and histories of surface to surface relations.

The data for this work are time-ordered lists of occluder-occluded pairs and directions of motion in depth of surfaces (toward or away from camera). [14] and [26] have shown methods whereby occluder-occluded relations may be obtained. No further data are required.

Suppose that all objects are stationary (i.e., as in static scene analysis). The data provided are triples of the sort: A occludes B. In terms of depth, z, this means, for surfaces with changes in depth that are negligible with respect to inter-surface depths, that $z(A) < z(B)$. We use the notation $A < B$. Occlusion data thus places a partial ordering on the depths of surfaces; and for static scenes, transitivity suffices to provide all computable depth constraints between surface patches. This partial ordering does not change over time. Thus, for example, given the data set: $A < B$; $B < C$; $B < D$, transitivity gives us that: $A < C$, $A < D$, and that there is *no* ordering in depth between C and D. Inconsistencies in data and in deductions are trivially detected, though not trivially resolved.

When objects are permitted to move in a plane parallel to the image plane 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.

If objects are allowed to move in depth, then the depth order obtained by a local occlusion analysis can no longer be used as a sorting criterion. Transitivity does not hold into the future when depths change over time. An approach to this problem is to project depth constraints between two objects into the future, and then use those derived constraints in transitive relations at the time of interest.

4.3 Velocity rules

There are four physically derived rules which give the projection into the future for the depth orderings. Motion in depth is v. If $Sign(v_{t,t+\Delta t}) < 0$ then motion is *toward* the observer on the temporal interval $(t, t+\Delta t)$. For $Sign(v_{t,t+\Delta t}) > 0$ motion is *away* from the observer on the same interval. $Sign(v) = 0$ means there is no significant motion in depth. The four rules are:

- *rule1* $A_t < B_t$ and $v_{t,t+\Delta t}(A) = 0$ and $v_{t,t+\Delta t}(B) = 0 \Longrightarrow A_{t+\Delta t} < B_{t+\Delta t}$
- *rule2* $A_t < B_t$ and $v_{t,t+\Delta t}(A) < 0$ and $v_{t,t+\Delta t}(B) = 0 \Longrightarrow A_{t+\Delta t} < B_{t+\Delta t}$
- *rule3* $A_t < B_t$ and $v_{t,t+\Delta t}(A) = 0$ and $v_{t,t+\Delta t}(B) > 0 \Longrightarrow A_{t+\Delta t} < B_{t+\Delta t}$
- *rule4* $A_t < B_t$ and $v_{t,t+\Delta t}(A) < 0$ and $v_{t,t+\Delta t}(B) > 0 \Longrightarrow A_{t+\Delta t} < B_{t+\Delta t}$

These rules are all expressed in
$A_t < B_t$ and $v_{t,t+\Delta t}(A) <= 0$ and $v_{t,t+\Delta t}(B) >= 0 \Longrightarrow A_{t,t+\Delta t} < B_{t,t+\Delta t}$
They are referred to in later text as *velocity rules*. For other motions of A and B, that is, for $A_t < B_t$ where $v(A) > 0$ or $v(B) < 0$ there is no relative depth information between A and B at time $t+\Delta t$. Inconsistencies are detectable in this scheme when conflicting relations are derived (cycles are detected).

4.4 Temporally local inferencing on qualitative relations

The data are occluder-occludee pairs, and the direction of motion in depth (toward or away from observer). From these data, one can derive infront (and behind) relations for the present time using transitivity of depth ordering, and for the future using the velocity rules. The derivations for the future hold under the assumption that there is no change in the direction of depth velocity (magnitude is unimportant). However, we prefer not to have to deal with such an unstructured, open-ended future. Since the data are arriving at this system at each time step, we make the inferences into the future for one time step only. 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. Recall velocity rules are of the sort $A < B$ at t_0 plus some constraints on velocity of A and $B \Rightarrow A < B$ at t_1. There is no point in applying the transitivity rule at this point, it will not add any new arcs. 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. See figure 1 for a layout of the order of operations on the relations.

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. A time step is defined as when the next datum is available. It has no memory beyond one time step. It is *local*, temporally speaking. More global temporal knowledge is kept elsewhere in the system – specifically, in the object histories.

In the system there may be several relations between a given pair of segments. That is, each relation has two segments and a label. For the segment pair, A and B, we may have, e.g.

order	label
$A < B$	*rule1*
$A < B$	*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. 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 graph. For example, suppose we have the relation: $A < B$: *rule1* for the graph in $\Psi^*(t)$. We then read the datum $B < A$. The graph then contains the cycle $A \rightleftharpoons B$.

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 *rule1* 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 *rule1* application at time $t-1$ was wrong then either the $v(A)$ was wrong, $v(B)$ was wrong, the relation $A < B$ 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 $A < B$ 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 a lot of overhead and cannot be resolved or reduced from the information currently available. 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.

So we do not make any attempt to undo any bad effects from possible bad data in the past. The present data are a different matter. We do, however, want the correct data to eventually outweigh any incorrect inferences. We have 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. For example, see figure 2 in which we show the derivation of an inconsistency.

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, especially of the nature of the data, noise, and sometimes even *a priori* probability values of the data, as well as assumptions of independence, or and knowledge of correlations. In the spirit of qualitative processing, we wish to avoid making such restrictive 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 [23] 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. The problem of which of two contradictory relations to propagate we deal with heuristically: 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 only. 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 number of time steps.

5 Chronologies

5.1 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, unlike story understanding, the data are arriving in a time-ordered fashion, e.g. in frame i, the is some set $\Psi(i)$ of relations, at frame $i+1$ some other set $\Psi(i+1)$. The chronology of relations has the same appearance as the data with the addition of derived relations. 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	$(< AB)\,(< AD)$
2	$(< AB)\,(< AD)$
3	$(< AB)\,(< AD)$
4	$(< AB)\,(< BC)\,(< AC)$
...	...

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

The second indexing method is to organize by relation. That is, one surface is the primary index, the second surface the secondary index. For example:

1°	2°	*temporal intervals*
A	B	(1, now)
	C	(4, now)
	D	(1, 3)
B	C	(4, now)
...	...	...

If relations persist, or are repetitious, then it saves on space to index by surfaces. 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.

5.2 History and world model of depths

Despite the fact that dynamic vision has a lot of data available, thanks to its *rampant redundancy* [31], 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 [15].

6 Experiments

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 follows.

- 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 $A < B$ with confidence .70 present, then we derive $A < B$ with confidence .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 $A < B$ because of a *rule1* application with $cf = .9$, then we read the datum $A < B$, the previous *rule1* edge is deleted, only the data edge remains.

We did not perform theoretically rigorous derivations of confidence factors. This is because the exact rules for combining confidences is not important in our scheme. Confidences propagated with velocity rules have a "time-decay" built in. That is, if $A < B(t_0); cf : x$, with appropriate velocities, then we derive $A < B(t_1); cf : x'$, 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.

We present the results of an experiment in the series of figures 3 – 7. The input is echoed in the "input-list" window. The "active-relations" window contains a list of edges with the label (reason) and confidence. In figure 3 the data is only the three v motions of the objects A, B, and C. In figure 7, 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, i.e., tc are decayed. In figure 7, the same three relations remain, because the velocity rule rule1 infer them. The confidences have all decreased. 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 7, we have in the data $D < A$; 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 7 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 7 has the contradictory relation $A < D$ just read in. In figure 7, only the maximum relation of the contradictory relations is printed out. The old $A < D$ because of *rule1* 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 $A > B$ relations, then choose the maximum of the two.

7 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] Borchardt, G.C., "Event calculus," *Proceedings*: 9th Internat. J. Conf. on Art. Intell., 1985, 524-27.

[8] 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.

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

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

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

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

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

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

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

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

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

[18] Minsky, M., A framework for representating knowledge, MIT AI Lab Memo #306, Cambridge, MA 1974.

[19] O'Rourke, J. and N. Badler, "Model-based image analysis of human motion using constraint propagation" *IEEE Transactions on Patt Anal and Machine Intell, PAMI-2*, #6, November 1980, 522-536.

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

[21] Roach, J.W. and J.K.Aggarwal, "Computer tracking of objects moving in space", *IEEE Transactions on Patt Anal and Machine Intell, PAMI-1* #2, April 1979, 127-135.

[22] 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.

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

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

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

[26] 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.

[27] 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.

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

[29] 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.

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

[31] Terry Weymouth, Personal communication.

[32] Witkin, Andrew P. "Scale-space filtering", *Proceedings* 8th Internat J Conf on Art Intell, 1983, 1019-22.

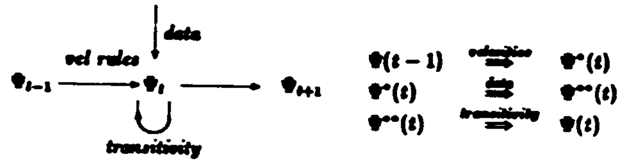

Figure 1: The Ψ system

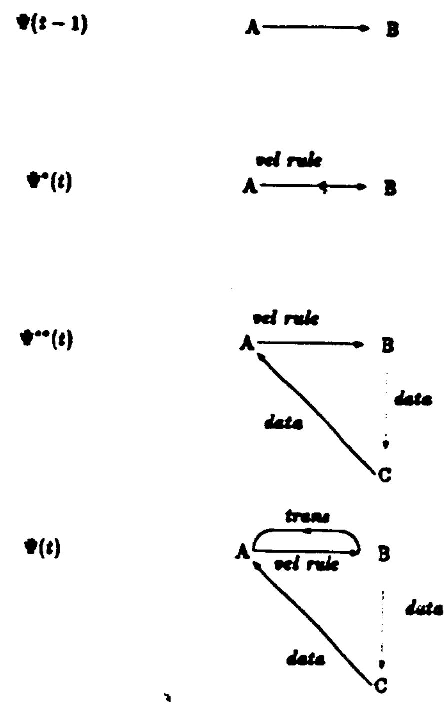

Figure 2: Contradictory relations not having *data* label

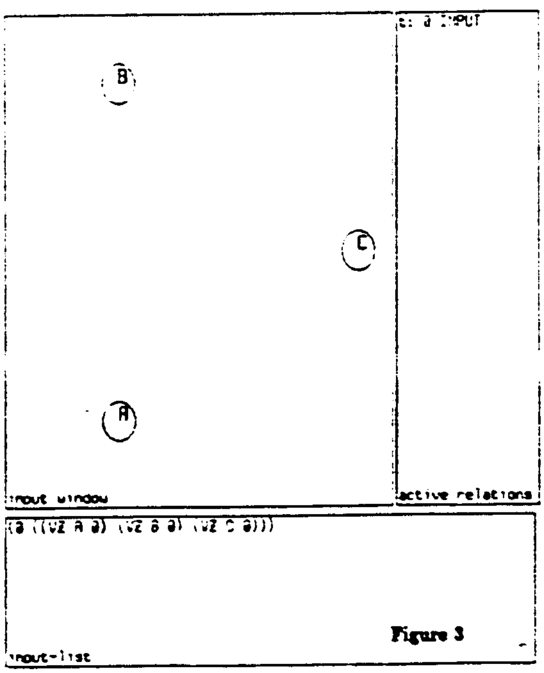

Figure 3

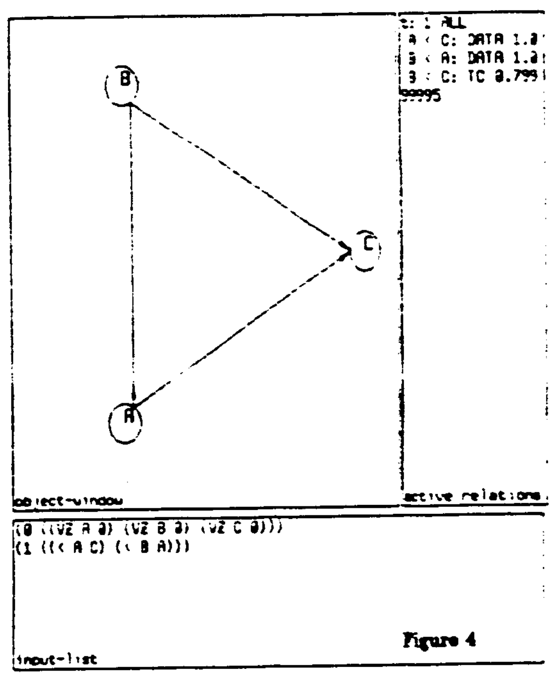

Figure 4

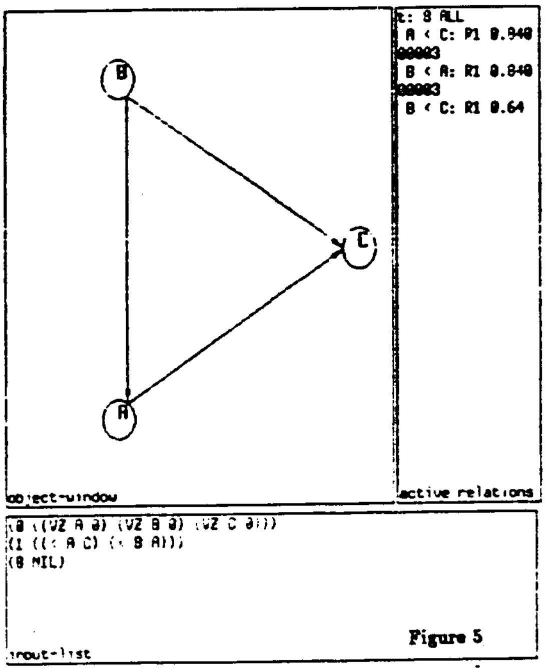

Figure 5

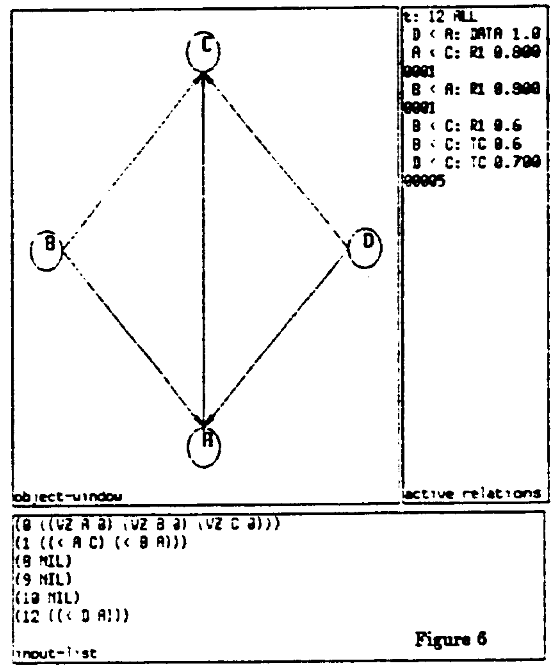

Figure 6

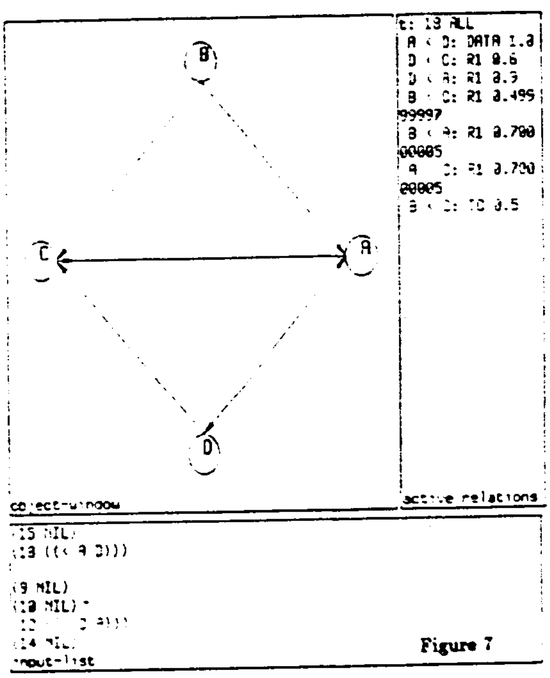

Figure 7

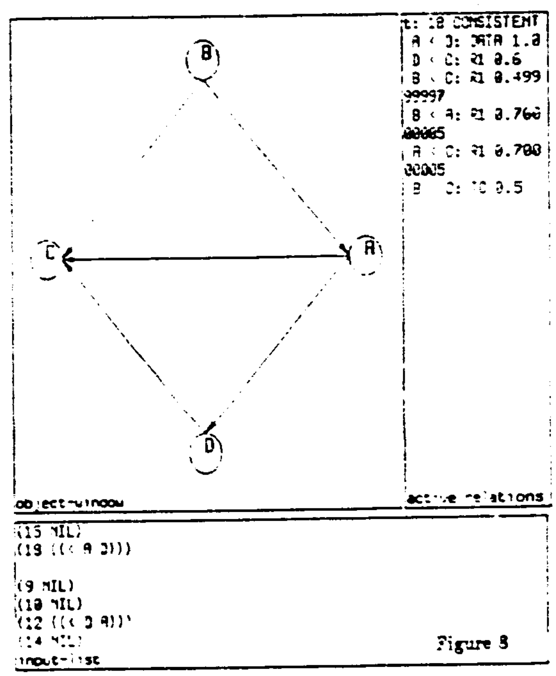

Figure 8