

# The Consistent Labelling of Image Features using an ATMS

R. Bodington, G.D. Sullivan & K.D. Baker

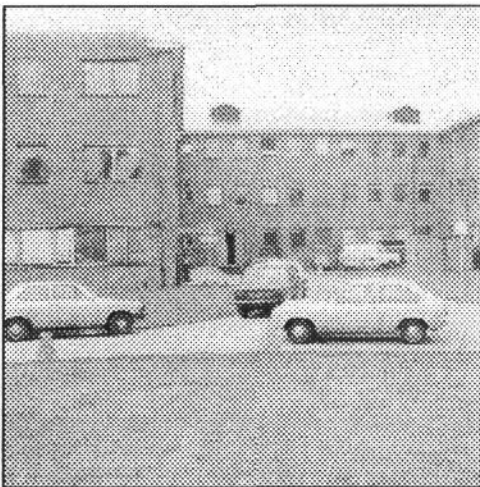
Intelligent Systems Group, Department of Computer Science,  
University of Reading, RG6 2AX, U.K.  
R.Bodington@reading.ac.uk

---

*Labelling sets of 2-D image features as model features is a constraint satisfaction problem that occurs in model-based vision. The labelling must be consistent with constraints that describe how image features originating from the modelled object would appear in the image. This paper discusses how an assumption-based truth maintenance system, ATMS, can be used to solve such a constraint satisfaction problem. The ATMS is used to limit the number of constraints applied, and to represent the multiple sets of consistent labels that are possible. The effectiveness of the ATMS in limiting the constraints is analysed.*

---

This paper concerns the use of model-based vision techniques for the identification of vehicles in complex outdoor scenes<sup>1</sup>. Figure 1 shows a typical scene. There are multiple classes of vehicle and several instances of vehicles in the scene.



**Figure 1. Typical scene**

Image features extracted from such scenes lead to multiple, ambiguous interpretations. This is due to the incompleteness of the image features generated by a vehicle in the scene, and the generation of false positives. The incompleteness of image features arises from the inability of the image processing techniques to identify all features, and because a vehicle may be self-occluding or occluded. False positives are image features generated by other objects in the scene, such as buildings, vegetation, and other vehicle types.

In this paper, we describe how the assumption-based truth maintenance system (ATMS) of DeKleer<sup>2</sup> has been used to represent multiple interpretations, and how the ATMS can support constraint based reasoning to identify

the correct interpretation. Some 3-D constraints involve a vehicle model being instantiated and projected into the image. Typically, these are computationally expensive. The paper will show how the ATMS is used to control the invocation of such constraints and to record and reflect the results throughout the search space. We analyse how effective the use of the ATMS has been.

## AN APPROACH TO MODEL-BASED VISION

Our approach to model-based vision, relies on the ability to verify a hypothesis for the existence of an object in the scene by matching an instantiation of a model in the 2-D image. Model-based verification proceeds by inverting the view perspective<sup>3</sup>, and then performing an iconic evaluation of the model<sup>4</sup>. View perspective inversion requires that a set of lines in the image are labelled as model features, and an indication of the viewpoint and position of the object in the image. View perspective inversion produces an instantiation of the model that can be verified by the iconic evaluator.

Conceivably, model-based verification can be exhaustively applied to all sets of lines extracted from the image, however, the combinatorics are overwhelming. Consequently the use of heuristic knowledge to limit the search is appropriate.

Experience has shown that the existence of a vehicle in an image may lead to the generation of certain distinctive image features. These features are not solely, nor are they always generated by a vehicle. However, they do provide independent evidence that allows a hypothesis for the existence of a vehicle to be made with a degree of confidence. These image features can be considered *cues* to subsequent vehicle recognition. In this paper, cues are groups of edges in the image that can be labelled as features on a model of a vehicle. An example of such cues are closed polygons. They may have been caused by windows on a vehicle and so should be labelled as windows on the model. At the outset, all such cues are identified. A search then takes place for a group of cues that can be labelled as the features of a single model. The search is restricted by ensuring that the groups identified are consistent with knowledge of how a vehicle that generates such cues, would appear in the 2-D image. This knowledge can be expressed as constraints. The problem of identifying sets of labelled cues that are consistent with the constraints is a *consistent labelling*, or *constraint satisfaction* problem.

The identification of a consistently labelled group of cues will enable bounds on the viewing angle to be

constructed using viewpatch reasoning as described by Rydz<sup>5</sup>. The inversion of the view perspective and an iconic match can then be performed in an attempt to verify the hypothesis for the existence of the associated vehicle. Verification will return a numerical score indicating the degree of success. The scores from processing different groups can be compared to determine those most likely to result from a vehicle. The complete process can be summarized in figure 2.

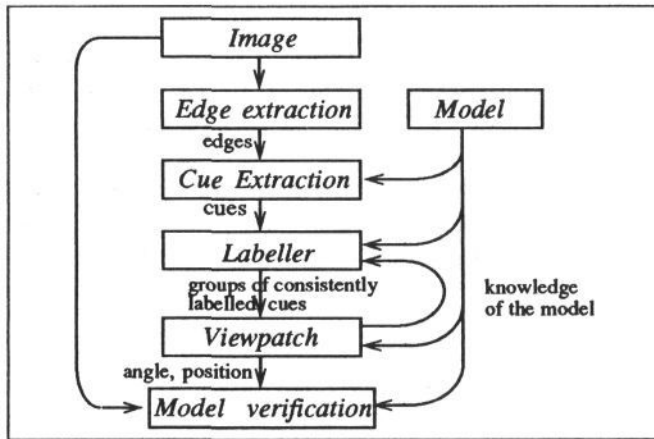


Figure 2. The labelling process

The result of using constraints in this manner is to reduce the amount of model-based verification.

### CONSTRAINT SATISFACTION

The central problem addressed in this paper is the constraint satisfaction problem associated with the identification of sets of consistently labelled cues that provide a starting point for model-based verification.

The structure of a constraint satisfaction problem can be defined by:

- a set of variables.
- a set of values that can be assigned to a variable. This is referred to as the domain of a variable.
- a set of constraints on the values assigned to the set of variables.

A set of variables, each assigned a single value from their respective domains, is a CSP-labelling. If the set of variables in the CSP-labelling is a subset of the variables in the structure of the constraint satisfaction problem, then the labelling is partial, otherwise, it is complete. A constraint is applicable to a CSP-labelling if the set of variables involved in the definition of the constraint are a subset of variables of the CSP-labelling. A CSP-labelling is consistent, if all applicable constraints are satisfied. A CSP-labelling is maximal if no consistent CSP-labelling exists that is its superset. A solution to a constraint satisfaction problem, is a complete and consistent CSP-labelling.

In the model-based vision system, the model features are the variables, the domains of which are the cues that could possibly be labelled as that model feature. An assignment is then the labelling of a cue as a model feature. The identification of a group of labelled cues is the identification of a CSP-labelling.

There are three basic forms of constraints used. Constraints based on the geometry of the 3-D vehicle

model and the 2-D geometry of image features. These are not strong discriminators as foreshortening due to different viewpoint and the affect of scale must be accounted for. Constraints based on the topology of the model and cues. Constraints that determine whether there is a viewpoint from which the group of labelled cues are all visible<sup>5</sup>. The topological and view consistency constraints provide more discriminatory power than the geometrical constraints but the overall problem is weakly constrained leading to many partial CSP-labellings.

It is very unlikely that a complete CSP-labelling will be found. This is due to several reasons. The cues are generated by vehicles (unreliably) as well as by the background scene. The vehicle may be self-occluding or occluded. Consequently the variant of the constraint satisfaction problem addressed is the identification of the maximal, consistent, partial CSP-labellings. Each of these CSP-labellings is a hypothesis for a vehicles existence that must be verified by model-based techniques. However, as the 2-D to 3-D constraints are weak, and because of the nature of the cues being used, there may a number of maximal, consistent, partial CSP-labellings. In order to reduce the model-based verification further, more heuristic knowledge must be used.

### RATIONALE FOR USING AN ATMS

While the maximal CSP-labellings are consistent according to the constraints, they are in fact ambiguous scene interpretations. The reasoning system employed to reduce such interpretations must be able to represent the ambiguity to allow the pursuit of the solution path deemed most opportune at any state.

In the identification of maximal, consistent, partial CSP-labellings, there are situations where constraints are applied to the same data in different parts of the search space. If the constraints employed are computationally expensive, such as model-based verification, then this replication of effort is undesirable. For example, consider the extracted cues shown in figure 3a.

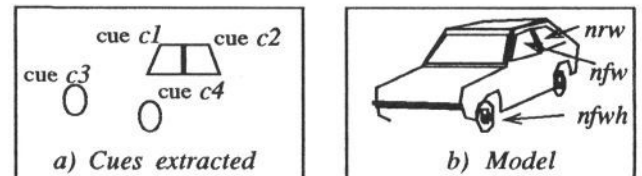
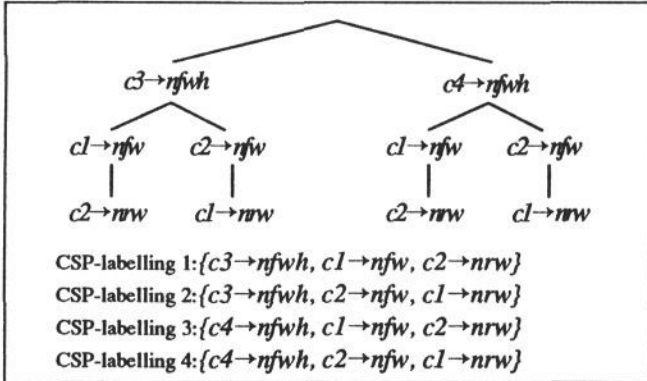


Figure 3.

For simplicity, assume that the subset of model features being used as variables are the nearside front side-window, nfw, the nearside rear side-window, nrw, and the nearside front wheel, nfw. These are shown in figure 3b. The cues available are: c1, c2, c3, c4. These are the values of the constraint satisfaction problem, the structure of which is shown in table 1. If a tree search method is used to deduce the possible CSP-labellings, then the search space explored is as shown in figure 4. An assignment of a value to a variable is represented as  $\rightarrow$ . For example:  $c3 \rightarrow nfw$ . As indicated in figure 4, there are four possible CSP-labellings.

**Table 1. Constraint Satisfaction Problem**

Variable	Domain
<i>nfwh</i>	{ <i>c3</i> , <i>c4</i> }
<i>nfw</i>	{ <i>c1</i> , <i>c2</i> }
<i>nrw</i>	{ <i>c1</i> , <i>c2</i> }



**Figure 4. Tree searched**

If there is a constraint between the two side-windows which is evaluated as the search takes place, it can be seen from the diagram that the constraint will be evaluated on the partial CSP-labelling {*c1*→*nfw*, *c2*→*nrw*} twice. Once to check the consistency of the CSP-labelling 1 and once to check the consistency of the CSP-labelling 3. Having established that constraints defining the relationship between two side windows are satisfied for the first labelling, it should not be necessary to repeat the computation for the second.

As has been said, this duplication of effort is particularly undesirable when expensive constraints are used as they are in this application. This should be contrasted to labelling approach adopted by Grimson and Lozano-Pérez<sup>9</sup>. In their application, it was possible to precompile the geometrical constraints used and store them in a lookup table. This allowed very rapid evaluation of constraints. A depth-first search was employed to identify the consistent labels. This search strategy re-evaluates constraints on the same data, however, as the evaluation of constraints is computationally cheap, there is little overhead.

The reasoning system has two requirements. It must be capable of representing multiple conflicting states, and results obtained in one part of the search space must be carried across to other parts. Part of the motivation for the development of the assumption-based truth maintenance system, ATMS<sup>2</sup>, was to provide a general solution to these two requirements. Thus the ATMS would seem an appropriate mechanism to support the reasoning required for model-based vision systems.

**THE ATMS**

The ATMS is a general purpose mechanism for recording statements of belief and reasons for belief. The ATMS will maintain multiple sets of consistent beliefs referred to as *contexts*. Each context is consistent, but any two contexts may be inconsistent. The ATMS uses a dependency network of *nodes* and *justifications* to record statements of belief. A node represents a potential problem solver belief. There are two special types of

nodes. A *premise* node represents a problem solver premise. An *assumed* node represents a problem solver assumption. The reason for belief in, or the support for a node is described by a *justification*. The node being justified is the *consequent* of the justification, and the nodes providing the support for the consequent are the *antecedents* of the justification. If all the antecedents of the justification are believed, then there is a *valid* reason for the belief of the consequent node.

Every node has an associated *ATMS-label*. An ATMS-label is a set of *environments*, where an environment is a set of assumptions providing support for the node. The environments of an ATMS-label are minimal with respect to each other. That is no environment in an ATMS-label is subsumed by any other environment in that ATMS-label. When a node is justified, a new reason for belief in a node is given. The ATMS-label of the node is updated to reflect this and the new belief is propagated to all nodes justified by this node. A node is believed in all contexts that can be derived from the environments of its ATMS-label. The *context* of an environment is the set of beliefs that are supported solely by a subset of the set of assumptions in the environment.

The ATMS records inconsistencies by the justification of a special node, the *contradiction node*. The set of environments forming ATMS-label of the contradiction node are referred to as *nogood* environments. All the contexts represented by the ATMS must be consistent. Therefore no ATMS-label of any node, apart from the contradiction node itself, can contain an environment that is subsumed by a nogood environment.

A further function performed by the ATMS is that of *interpretation construction*, where the ATMS is used to identify the set of all consistent maximal environments that can be formed from a set of assumptions. An environment is maximal if it subsumes no environment. Interpretation construction proceeds as follows. First, all the environments of size two formed from the set of assumptions and not subsumed by a nogood environment are identified. This set of environments is then used to construct environments of size three, again avoiding environments subsumed by nogoods. This process continues until the set of maximal and consistent environments has been identified.

**CONSTRAINT SATISFACTION AND THE ATMS**

Each assignment in constraint satisfaction will be represented as an assumption in the ATMS. Interpretation construction will generate all consistent environments formed by these assumptions, and so identify all the possible CSP-labels.

During interpretation construction, the consistency of each environment explored is determined by whether it is subsumed by a nogood environment. In the constraint satisfaction problem, these nogood environments are formed from a set of assumptions whose corresponding assignments fail to satisfy a constraint. The recording of the nogood ensures that the exploration of any CSP-labelling, environment, that would not satisfy the



constraint is avoided. The ATMS must be extended to provide a mechanism that will enable the evaluation of constraints during interpretation construction. Such a mechanism must prevent the re-evaluation of a constraint on the same set of assignments. The *consumer architecture* of DeKleer<sup>6</sup> provides such a mechanism. The subsequent paragraphs describe an implementation derived from the consumer architecture. This implementation is described in more detail by Bodington and Elleby<sup>7</sup>.

A *consumer* is a special ATMS node representing a problem solver procedure. In the context of this paper, a consumer contains the intended procedure for determining the consistency of a constraint. The ATMS-label of a consumer node consists of sets of assignments for which the associated constraint can be evaluated. This ATMS-label is constructed by creating a node for each variable and justifying it by every assumption representing the assignment of a value to that variable. A consumer node is justified by the set of variables involved in the corresponding constraint. An example is shown in figure 5. The consumer represents the constraint,  $a+c>4$ , between the variables  $a$  and  $c$ . The possible values assigned to  $a$  are 1 and 3, represented as assumptions  $a1$  and  $a3$ .  $c$  can be assigned the value 5. The consumer can be evaluated in the environments in the label of the consumer,  $\{a3,c5\}$  and  $\{a5,c5\}$ .

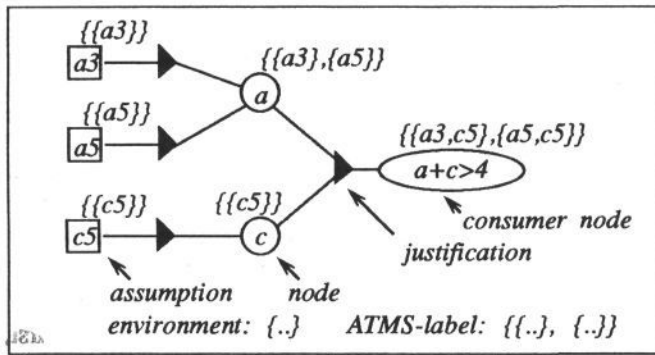


Figure 5. ATMS network for a consumer

When a consumer is evaluated in an environment, the environment is removed from the label of the consumer. This ensures that a constraint is only ever evaluated once on any set of assignments. The constraint represented by the consumer in figure 5 has not been evaluated so the ATMS-label is complete.

When consumers are used, the consistency of an environment explored by interpretation construction is determined first by checking for the subsumption by nogood environments, and then by the application of any consumers that have the environment in their label. Interpretation construction, explores the smallest environments first, so constraints that generate nogoods of low cardinality are evaluated first. This avoids the redundant constraint evaluation that can occur when a constraint is evaluated in an environment that is subsequently subsumed by a nogood environment.

The use of interpretation construction in this fashion ensures that the results discovered in one part of the search space, the application of constraints, are available throughout the rest of the search space. The minimum

amount of constraint evaluation will be used to determine the set of maximal CSP-labellings<sup>7</sup>.

## PRACTICAL INTERPRETATION CONSTRUCTION

### Identification of consistent groups of cues

In this application, it is plausible to identify all model features, and assign them cues that could be labelled as such. Interpretation construction and constraints can then be used to identify all the partially labelled cars. However, many cues may well be extracted from the image, of which only a few are generated by the vehicles. Consequently, a large number of environments will be explored, the majority of which will be inconsistent. As indicated by Provan<sup>8</sup>, the number of environments explored and the number of nogood environments will rise exponentially. Viewing nogood environments as sets of assumptions, this combinatorial explosion can be reduced by constructing constraints that generate nogoods of low cardinality. The smaller the nogood size, the more environments subsumed by it, and so the more environments pruned.

A further method of controlling the size of the search space is to formulate the problem using domain knowledge so that the areas of the search space explored, are those likely to contain solutions. Environments that are known to be nogood prior to the search commencing should be avoided. One way to achieve this is to use the concept of a *seed cue*. A seed cue is used to estimate an *area of interest* in the image. The area defines the bounds of the possible projections by the vehicle onto the 2-D image as predicted by the assignment of a model feature to the seed cue. Only cues that lie within this area could be caused by the same vehicle as the seed cue. Avoiding the consideration of cues that lie outside the area of interest prevents the exploration of environments known to be nogood.

All cues provide some evidence for a vehicle in the image and so are potential seed cues. However, certain cues enable the invocation of more restrictive 2-D constraints and are consequently more useful in limiting the search. For example, an S shape cue, caused by the roof, windscreen and bonnet, will give a rough indication of the orientation of the vehicle in the image so enabling the exclusion of certain model features due self occlusion. Other cues are more likely to be false positives. For example, closed polygons are regularly generated by windows on buildings. The usefulness of cues as seed cues can be ranked according to these criteria. Search then proceeds by selecting the seed cue that will be the most effective. All cues within the area of interest generated by the seed cue are identified and all possible assignments are made. Interpretation construction and constraint satisfaction then takes place. The interpretation construction is focussed so that the only environments that are explored are those that include an assignment for the seed cue. This leads to the identification of the set,  $MS$ , of all maximal, consistent CSP-labellings, such that:

$$MS = \{ mS \mid mS \text{ is a maximal, consistent CSP-labelling} \}$$

Each  $mS$  is equivalent to an environment in the ATMS, and refers to a set of model features that have been assigned a cue and includes an assignment to the seed cue. Each  $mS$  provides a hypothesis for the existence of the vehicle in the image as suggested by the seed cue.

### Verification of consistent groups of cues

Once the set,  $MS$ , has been identified, a position and orientation of a vehicle for each element,  $mS$ , of the set  $MS$ , can be predicted in the image<sup>5</sup>. Model-based verification can then be performed on each  $mS$  in order of the likelihood of success. Currently, the criterion on which this ordering is based, is the number of cues assigned to model features forming the set  $mS$ . Additional criteria are being considered. For example, the quality of the data forming the cues, and the number and types of constraints that the group satisfies, can be analysed.

Model-based verification returns a numerical score. If the score is of a certain level, then the assignments in the set  $mS$  are assumed to be correct, a vehicle has been identified. All the assignment assumptions in the set  $mS$  are then believed, the cues are not considered again and all other environments containing the believed assumptions are ignored. If the score is extremely low, then the assignment assumptions forming the set,  $mS$ , are considered to be incorrect and form a nogood environment. The possibility that the set,  $mS$ , is incorrect due to the inclusion of an assignment to a false positive as well as the correct partial CSP-labelling must not be excluded. So the set of maximal consistent environments that subsume this nogood environment must then be identified. If these are maximal, relative to the set  $MS$ , then they should be included in the set  $MS$ .

Consider the case (figures 3 and 4) where the CSP-labelling,  $\{c3 \rightarrow nfw, c1 \rightarrow nfw, c2 \rightarrow nrw\}$  has been identified as consistent according to 2-D constraints. The cue  $c3$  is in fact a false positive, so model-based verification techniques will indicate that the CSP-labelling is a nogood environment. The maximal environments that subsume this nogood environment are:

$$\{\{c3 \rightarrow nfw, c1 \rightarrow nfw\}, \{c3 \rightarrow nfw, c2 \rightarrow nrw\}, \{c1 \rightarrow nfw, c2 \rightarrow nrw\}\}.$$

Model-based verification on this set will indicate that  $\{c1 \rightarrow nfw, c2 \rightarrow nrw\}$  is the correct CSP-labelling.

Once all the  $mS$  sets within the set  $MS$ , have either been verified by model-based verification techniques or subsumed by assignments comprising of cues that have already been correctly assigned to model features, another seed cue is selected and the process is repeated. It is probable that the area of interest predicted by a seed cue will overlap previously explored areas, so the same cues will be re-assigned to the same model features and be reconsidered. As the ATMS has been used, the consistency of partial CSP-labellings formed from these sets will not be determined by the re-evaluation of constraints.

When all cues have been used as the seed cue, all sets of possible consistent CSP-labellings will have been explored. The scores of the maximal CSP-labellings that were not high enough to be believed are now compared in

order that further vehicles may be identified. If a maximal set  $mS_1$  has a null intersection with any other  $mS_j$ , then the assignments comprising  $mS_1$  are an indication of a vehicle. If the intersection set is not empty then the assignments in the  $mS$  with the highest score are the labels for the vehicle.

## RESULTS

The work described in the paper is still under development. The cue extraction process is not complete. At present, only closed polygons, wheel arches, and inverted bucket shapes can be extracted. A subset of the 2-D to 3-D constraints have been implemented. The labelling has not been linked to the model-based verification techniques. However, preliminary results give some indication of the effectiveness of the ATMS constraint satisfaction technique in limiting the number of constraints evaluated. Consider the image shown in figure 1. The connected edge map that is extracted using a single resolution Canny edge detector is shown in figure 6. Figures 7a and 7b show the symbolic representations of the cues that have been extracted from the Canny output using different thresholds to determine whether a group of image features are acceptable as a cue. These cues required some interactive editing of the existing cue extraction processes.



Figure 6. Output from Canny edge detector.

The performance of the ATMS labelling is summarized in table 2. The image in figure 1 has been used with three different levels of threshold. Entry a) is a measure of how many of the environments explored are members of an ATMS-label of a consumer that has already run. Such environments represent a partial CSP-labelling that has already satisfied the constraint encoded by the consumer. Entry b) is the number of environments explored that are immediately subsumed by a nogood environment, and have one or more consumers pending execution. Such environments represent partial CSP-labellings whose inconsistency has been determined elsewhere in the search space. An indication of the savings in constraint evaluation by the ATMS can be gained by comparing these two entries to the total number of constraints evaluated as shown in entry c).

The results show that the more cues there are, due to lower thresholds, the more ambiguously labelled cues are possible, so the greater the savings in constraint evaluation. This saving must obviously be countered by

the increase in number of environments explored and number of nogood environments generated.

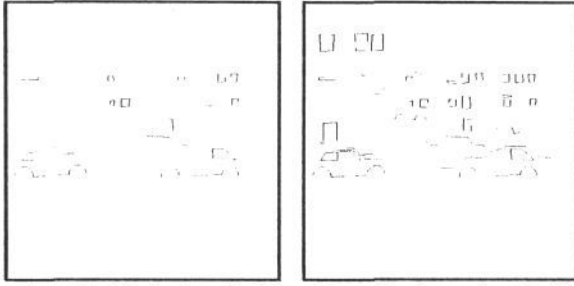


Figure 7a. High threshold      Figure 7b. Low threshold

Figure 7. Cue extracted from edges in figure 6

Table 2. Performance of ATMS labelling

	Low	Medium	High	
Cues: -----	16	26	38	
Environments explored: -----	195	402	758	
Nogood environments: -----	89	131	244	
Constraints evaluated: -----	69	113	227	
Constraints satisfied: -----	38	47	70	
Constraints failed: -----	31	66	157	
Reduction in constraints by nogoods: -----	0	18	43	a)
Reduction in constraints by goods: -----	8	43	79	b)
Percentage reduction in constraint evaluation: -----	12	54	54	c)

The constraints and cue extraction processes are being completed, and the model-based verification techniques incorporated. The technique will then be applied to a range of images enabling a more concrete measure of the effectiveness of the ATMS in limiting the amount of constraints evaluated.

## DISCUSSION AND FUTURE WORK

Currently model-based verification is only used once the set of maximal CSP-labellings consistent with the 2-D constraints have been identified. The reason being, that if a maximal set satisfies all the 2-D constraints, then it is likely to be a correct labelling. However, as discussed, these maximal sets may include false positives resulting in further model-based verification. It may prove a better policy to invoke the model-based verification techniques as soon as possible. These constrain the search more effectively than the 2-D constraints and so inconsistent partial CSP-labellings will be discovered early in the search, thus pruning large sections. A consequence of this approach, is that computationally expensive model-based verification techniques are more likely to be repeated throughout the search space on the same data. This strengthens the case for using an ATMS in the manner described. This is being investigated.

The approach described in the paper is totally dependent on the extraction of adequate cues. Currently the cues used are edge-based and can be assigned a limited number of model features. It may prove useful to initiate the search using cues of this nature and then once model-based verification has been used, predict the location of model features in the image that can be assigned more ambiguous image features such as vertices or angles. A search for these features in the image can then take place. This is similar to the approach adopted by Bolles and

Horaud<sup>10</sup>. A further consideration is to use region or colour-based cues to start the search, to predict areas of interest, and to rank the seed cues.

The ATMS has performed two roles in this application. It has been used to represent multiple, ambiguous labels allowing best first or opportunistic reasoning to take place. The ATMS has also been used to improve the efficiency of the search by reducing the number of constraints applied. If this is to result in real savings, then the improvement in search efficiency must outweigh the inherent cost of using the ATMS, which as Provan<sup>8</sup> showed may generate an unmanageable number of environments and nogoods. The problem has been controlled here by using cues and seed cues. Whilst early results are encouraging, it remains to be seen how realistic the savings offered by the ATMS are in general practice.

## ACKNOWLEDGMENTS

The authors have received significant technical help, and ideas from Peter Elleby regarding the use of the ATMS in constraint satisfaction problems. The work on the cue extraction has been done by Andrew Rydz and Tina Angelikaki. Steve Rake of the IBM Scientific Centre Winchester and members of the Intelligent Systems Group at Reading have provided many valuable discussions.

## REFERENCES

1. Baker, K.D. & G.D.Sullivan "Alvey MMI007 Vehicle Exemplar: The Knowledge-Based Approach," *Proc. Alvey Vision Conference, AVC87*, 15 Sept. 1987.
2. DeKleer, J. "An Assumption-Based TMS," *Artificial Intelligence*, vol. 28, no. 2, March 1986.
3. Worrall, A., G.D.Sullivan & K.D.Baker, "Model-based Perspective Inversion," *Proc. Alvey Vision Conference, AVC88*, Sept. 1988.
4. Brisdon, K., G.D.Sullivan & K.D.Baker, "Feature Aggregation in Iconic Model Evaluation," *Proc. Alvey Vision Conference, AVC88*, Sept. 1988.
5. Rydz, A., G.D.Sullivan & K.D.Baker, "Model-based Vision Planar Representation of the Viewsphere," *Proc. Alvey Vision Conference, AVC88*, Sept. 1988.
6. DeKleer, J. "Problem solving with the ATMS," *Artificial Intelligence*, vol. 28, no. 2, March 1986.
7. Bodington, R. & P.Elleby "Justification and Assumption-based Truth Maintenance Systems: When and How to use them for Constraint Satisfaction," *AISB workshop on Reason Maintenance Systems*, University of Leeds, 14-15 April 1988
8. Provan, G. M. "Efficiency Analysis of Multiple-Context TMSs in Scene Representation." *Proceedings AAAI-87*, 1987
9. Grimson, W.E.L. and T.Lozano-Pérez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data." *MIT internal report, A.I. Memo 738*, August 1983.
10. Bolles, R.C. and P.Horaud, "3DPO: A Three-Dimensional Part Orientation System," *The Intl. Jnl. of Robotics Research*, vol. 5, no. 3, Fall 1986.