Experiments on the use of the ATMS to label features for object recognition

R.M. Bodington† G.D. Sullivan†† K.D. Baker††

1. Abstract

Experiments are reported on the use of an Assumption-based Truth Maintenance System (ATMS) [6] to establish a match between a 3-d model and a single 2-d image. We show that the ATMS improves the efficiency of the search for maximal combinations of consistently labelled features. A memory cost is incurred, associated with the recording system of the ATMS; this can be reduced by simple heuristics. Empirical evidence is presented quantifying the costs and benefits of the method.

2. The consistent labelling of image features

Features extracted from images are usually ambiguous and uncertain. Object recognition depends on the discovery of extended combinations of image features which are mutually compatible with a known object. Grimson and Lozano-Perez [9,10] have shown that simple binary constraints between pairs of model features may be sufficient to reject most mislabellings. If the number of model features is small, it is feasible to store all binary constraints explicitly, in tables compiled in advance. Furthermore, it is only necessary to compute the measures once for a given set of image features, and these too can be stored. As the search proceeds, the consistency of a pair of labels can be tested simply by comparing the measured datum against values in the constraint table. The space of all possible combinations of labels may then be searched by a depth-first expansion of the interpretation tree.

Grimson and Lozano-Perez have shown that if the scene contains a single isolated object, very little of the interpretation tree need be explored. The strong constraints invalidate large sub-trees, and the search rapidly collapses to the single solution. However, a depth-first search with back-tracking is inherently very redundant. At each expansion of the interpretation tree, checks must be made that the newly labelled datum is pair-wise consistent with all existing labels. Identical checks are therefore duplicated throughout the tree. For example, two data features separated from each other by n levels in the tree, will be cross-checked $m^{(n-1)}$ times (where m is the number of model features - i.e. the "fan-out" of the tree). Such duplication of effort is only acceptable if the constraint checks are extremely cheap to evaluate. This is the case in [10] where the constraints compare pre-compiled geometrical measurements of 3-d model features with 3-d sensory data. The duplication is not acceptable in systems where the sensory data under-constrains the model, for example when matching 2-d data against 3-d models, as is considered in this paper.

† *British Aerospace, Sowerby Research Centre, PO Box 5, Filton, Bristol, BS12 7QW. UK*

†† *Dept. of Computer Science, University of Reading, Reading, RG6 2AX. UK.*

Two types of constraint are available between 2-d data and 3-d models:

(1) Qualitative constraints, derived from 2-d geometrical relationships between image features, such as coincidence, adjacency, enclosure, similarity. Changes of viewpoint cause large changes between image features, so these constraints are often weak, and erroneous matches may not be detected.

(2) Quantitative constraints, using 3-d knowledge, requiring all data to be geometrically compatible with a single perspective view of the object (see e.g. Lowe, [12]). This requires the viewpoint to be solved iteratively, followed by the evaluation of combinations of features. It provides strong, metrical constraints, but they are very expensive to apply.

Both types of constraint are inherently view-dependent, i.e. the exact relationship between the image features is strongly dependent on the pose of the object in front of the camera. Therefore 2-d to 3-d constraints cannot be recorded in advance in simple numerical look-up tables, and must be evaluated dynamically as the model becomes instantiated. The computational cost of applying such constraints is often large, and a highly redundant depth-first search with backtracking is impractical. Instead, a record of all partial results must be kept, so that unnecessary repetition of the work can be avoided. Truth Maintenance Systems have been proposed for this purpose in other areas of Artificial Intelligence. We demonstrate below how the ATMS can be used within a model-based vision system for recognising vehicles within unconstrained single images. The main features of the implementation have been reported elsewhere [1,2]. In this paper we report an experimental investigation into the costs and benefits of the ATMS for the consistent labelling of image cues. A more detailed report of this material is also available [3].

3. Outline of CARRS

CARRS (CAR Recognition System) is an experimental system for finding and locating vehicles in images such as Figure 1(a). CARRS adopts an "hypothesis-and-test" strategy which (in brief) consists of the following stages.

Stage 1: Data-driven determination of feature groups.

S1.1 Feature Extraction. Connected edges (Figure 1(b)) from a single scale Canny operator [5] are segmented into straight lines at curvature maxima, to form polygonal approximations.

S1.2 Cue Identification. Polygons (and fragments of polygons) are extracted to form cues for the labelling process. Cues are application-specific features which facilitate heuristic methods. At present these consist of U-shaped triples, S-shaped triples, and closed quadrilaterals. Each type of cue is associated with known components of the model which may give rise to it, e.g. U-shapes and quadrilaterals may arise at any of the windows, S-shapes may occur at the near- or off-side pillars of the windscreen. Figure 1(c) (central box) shows the cues found in Figure 1(b).

S1.3 Identification of Areas of Interest. Each cue is considered in turn as a seed

feature (SF), and is associated with a subset of all cues likely to be due to a single car. The set is based on proximity to the SF in the image, conditioned by rules dependent on the type of SF, and its size and orientation in the image. The SFs are then ordered according to the cardinality of the proximity sets (Figure 1(c)). Note the overlap between proximity sets.

Stage 2: Search for maximal consistent labellings

S2.1 Application of 2-d Constraints. Taking each SF in rank order, a search is made for maximal subsets of its proximity set, which may be labelled as model features, such that they are consistent with 2-d constraints. The constraints used are heuristic and domain-dependent; they are chosen to be fairly independent of viewpoint, and are not strongly specific to particular models of vehicles. They express requirements such as:

- The windows on each side of the car must be aligned and closely adjacent.
- A near-side window and an off-side window cannot be visible simultaneously.

These 2-d constraints use metrical concepts such as distance and orientation, which are scaled by measurements between junctions within cues.

S2.2 Hypothesis instantiation. Any maximal set of a SF which contains a sufficient number of cues is passed immediately to Stage 3 for verification. All other maximal subsets are stacked, pending the examination of the remaining SFs. When all SFs have been so treated any surviving maximal set is then passed to Stage 3.

Stage 3: Model-based verification of hypotheses.

S3.1 Viewpoint inversion. Labelled maximal sets allow the pose of a known object to be determined, using a 3-d model. This is carried out in three stages, each of which either invalidates the hypothesis, or progressively refines the viewpoint estimate.

(1) Labelled features identify a patch on the viewsphere from which all the features are visible - the "viewpatch" - which is represented as a quad-tree [14].
(2) The "roll-consistency constraint" [16] is applied, to reject parts of the viewpatch in which the angles (in the image) of labelled lines are inconsistent.
(3) The perspective transformation is inverted by iterating from the current best estimate of view [15].

If an hypothesis is rejected at any of these stages then the solution set is invalidated, and is removed from further consideration. The previously satisfied set which led to this invalid set is then reconsidered as a maximal set.

S3.2 Iconic verification. The vehicle hypothesis is evaluated by using the view estimate to project the entire model back into the image. An hypothesis-driven check is carried out for the existence of the projected model features in the image [4].

S3.3 Success Pruning. The acceptance of an hypothesis "consumes" the SF and cues in its maximal set. All constraints involving a consumed cue are declared invalid, and the cue is removed from all existing sets. This prunes the search carried out in Stage 2.

Stage 2 and Stage 3 are repeated as necessary, until all maximal partial labellings have been confirmed or invalidated.

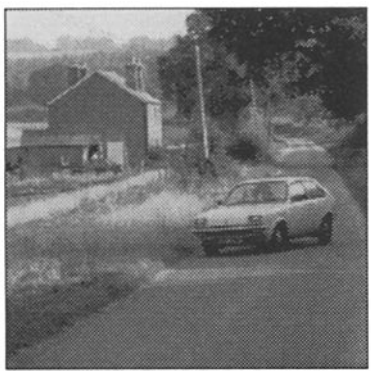

Figure 1 (a)

Figure 1 (b)

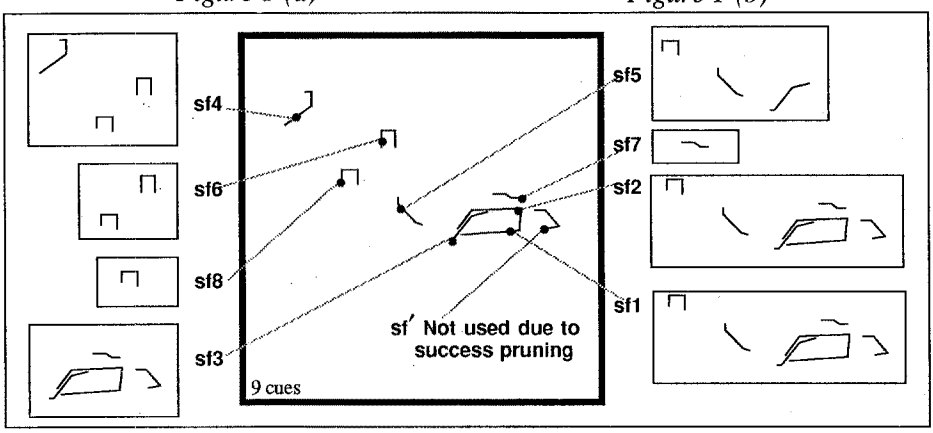

Figure 1 (c)

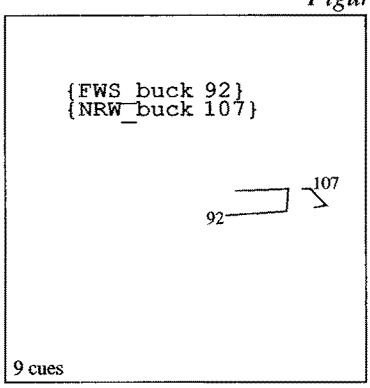

Figure 1 (d)

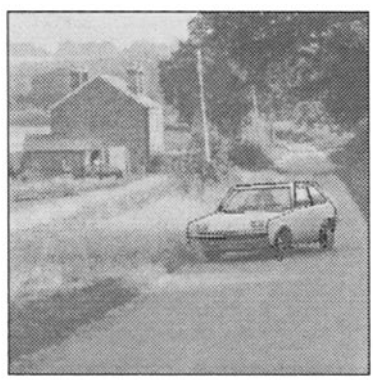

Figure 1 (e)

4. Efficient search methods

All three stages of the system impose significant computational burdens. This study concerns the search strategy used in Stage 2, and we take as fixed the cue sets delivered by Stage 1, and the verification algorithms used in Stage 3. The current problem is how to make efficient use of the weak 2-d constraints. There are two main

reasons why the methods adopted by Grimson & Lozano-Perez cannot be used.

(1) It is infeasible to pre-compile the constraints, and to pre-compute the pairwise measurements between features. The measurements between features carried out in Stage 2 are only weakly independent of view and must be parameterised by the geometry of the feature within the image.

(2) The constraints which are applied are highly specific to the particular labels being tested. An unselective initial calculation of all possible measurements between all pairs of features would be absurdly expensive.

Constraints must therefore be evaluated as and when required. Any evaluation which is repeated therefore entails a significant cost. A form of Truth Maintenance System is needed, to record results, and to resolve conflicts between partial inferences.

5. An ATMS approach to labelling

The ATMS [6] is a general purpose mechanism that supports reasoning over multiple hypotheses. We have previously shown that it can be used to support consistent labelling problems [1,2]. In brief, we represent each assignment of a model label to a feature as an assumption, and search for mutually consistent sets of assumptions (called environments). Constraints are recorded as ATMS constraint-nodes (akin to de Kleer's consumers), which are tested as new environments are explored. Invalid sets of labels are reduced to minimum inconsistent subsets, and recorded as "no-goods". All partial results of the search are stored, to avoid repeated evaluation of constraints; this is important where SFs have overlapping proximity sets. In addition, the ATMS maintains a "justification network" to record the interdependency of the data; this enforces coherence in the data when new inconsistencies are discovered (e.g. after success pruning in Stage 3).

The ATMS offers a radically different approach to that of the interpretation tree method [10] and can be used to solve a wider class of problems. However, there is a significant computational cost to pay: the recording mechanism consumes a great deal of memory, and the up-dating of the justification network is time-consuming. It is not immediately clear when the benefits outweigh the costs.

Recently, Provan [13] has criticised the use of an ATMS on the grounds that it fails to scale up to complex problems effectively. He considered the visual task of identifying the components of a humanoid puppet thrown onto a table-top. The simulated "sense-data" comprised the positions of rectangles, representing different parts of the puppet, together with additional spurious rectangles introduced to create multiple possible interpretations. Provan showed that as environments were expanded in the search, they could not be disproved until very late, when most of the components had been labelled. This resulted in large no-goods which are ineffective at invalidating other environments. The number of consistent partial labellings therefore increased exponentially, and created an unacceptable storage cost. Provan argued that the cost greatly outweighed the slight benefits, and that the ATMS approach is unsuited even to quite small labelling problems in vision.

Provan's example is only partly relevant to CARRS. It is true that strong constraints on the sensory data cannot be applied at early stages of the search. However, unlike Provan's example, CARRS can invoke strong (viewpoint-dependent) constraints at the later stages of processing (Stage 3 above). The use of "success pruning" (S3.3) then allows much of the ATMS memory load to be avoided, since it consumes features, and reduces the number of environments needing to be considered. Success pruning has an effect similar to the use by Grimson and Lozano-Perez [10] of a cut-off, when some fractional match has been found. It reduces the amount of the search tree explored and the number of environments which need to be maintained.

6. Experimental results

In this section the costs and savings of using the ATMS are considered, by using CARRS to analyse five representative images, IM1-IM5. Typical behaviour is illustrated using IM1.

6.1 Analysis of constraint savings

Where a significant portion of the search space is explored repeatedly, the encoding of constraints and recording of no-goods in the ATMS will result in a reduction in the number of constraints evaluated. The savings can be measured by recording how many constraints were necessary to check the consistency of an environment the first time it is explored. Each time an environment is re-explored, that number of constraint checks will have been avoided by the use of an ATMS. A count of the total savings is then increased by that amount.

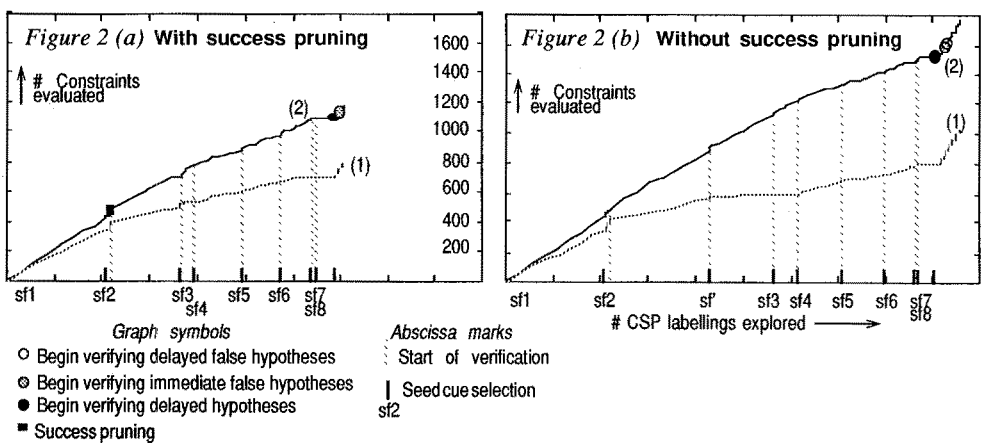

Figure 2: The affect of success pruning on the size of the search space

These savings are illustrated in Figure 2(a). The abscissa of the graph shows the number of labellings explored, i.e. the amount of the search space explored. The point where a new SF is selected is shown on the graph, as well as the points where hypotheses testing takes place. Graph (1) shows the number of constraints evaluated during the search and graph (2) shows the number that would have been evaluated if no recording mechanisms were used. The distance between the two graphs is an

indication of the savings in constraint evaluation gained by using the ATMS.

The results show that significant amounts of the search space are re-explored and that constraints are re-evaluated repeatedly. This may occur during the search within a proximity set, as happens during the search from SF1. However, significantly more re-exploration occurs when proximity sets overlap as is the case with SF2 and SF3 which have a large intersection with SF1, the first SF explored (see Figure 1(c)).

Images	IM1	IM2	IM3	IM4	IM5
Constraints					
(a.1) # evaluated:	799	467	1 007	1 344	1 126
Labellings explored					
(b.1) # explored:	2 366	3 198	3 382	5 399	5 030
(b.2) # re-explored:	1 372	2 004	1 996	3 280	3 012
(b.3) Percentage re-explored:	58%	63%	59%	61%	60%
Reduction in constraints evaluated					
(c.1) Total reduction:	374	833	655	1 460	1 356
(c.2) Percentage reduction:	32%	64%	39%	52%	55%
(c.2) = 100×(c.1) / ((c.1) + (a.1))					

Table 1 Summary of Constraint Savings

The savings in constraint evaluation gained by using the ATMS are summarised for the 5 test images in Table 1. Entry c.2 is the percentage of constraint evaluations that have been avoided by use of the ATMS encoding. The results show that using the ATMS does offer significant savings by avoiding constraint repetition. On aggregate over the 5 test images approximately 50% of the constraint evaluations are saved.

6.2 Analysis of results of success pruning

The effects of success pruning can be illustrated in CARRS by disabling the cue consumption. The result is shown graphically for image IM1 in Figure 2(b), where it is seen that a considerably greater number of constraints are evaluated.

Images		IM1	IM2	IM3	IM4	IM5
(a)	# of consumed cues:	2	-	2	2	3
(b)	# of labellings taken OUT:	66	-	55	63	106
(c)	Immediate verification reduction:	3	-	0	2	2
(d)	Delayed verification reduction:	1	-	8	7	12

Table 2 The effects of success pruning for each test image

The effects of success pruning for the 5 test images is summarised in Table 2. Entry (a) shows the reduction in the number of seed features being considered. Entry (b) shows the number of consumed image features that were previously labelled as model features. Entries (c) and (d) show the number of maximal labellings pending verification that were invalidated as a consequence of success pruning.

6.3 Memory requirements

The memory required by the ATMS increases with the search and limits the size of

application that can be considered. This limit can be estimated by measuring the size of the ATMS data structures, constraint nodes and justifications, environments and no-goods as each image is analysed. These are summarised in Table 3.

Images	IM1	IM2	IM3	IM4	IM5
Nodes,					
(a.1) # of constraint nodes:	789	927	1 093	1 219	1 155
(a.2) # of model feature nodes:	366	454	518	698	521
(a.3) # of feature labelling assumptions:	7	0	7	7	10
Justifications,					
(b.1) # stored:	2 772	2 768	3 774	5 323	4 405
Environments,					
(c.1) Final # stored:	931	3 778	2 221	2 071	2 041
(c.2) Largest # stored:	1 195	4 375	5 216	8 606	3 500
Nogoods,					
(d.1) Final # stored:	317	410	431	680	504
(d.2) Largest # stored:	317	410	431	680	504

Table 3 Storage requirement of the ATMS

Of these data structures, those that increase the most are the consistent environments and the minimal no-goods. Their growth for image IM1 is shown in the graphs in Figure 3 which shows the effect of success pruning clearly (occurring at the verification of SF1) in reducing the number of consistent environments and minimal no-goods stored. These results indicate that in CARRS the storage overhead associated with the use of an ATMS does not grow excessively.

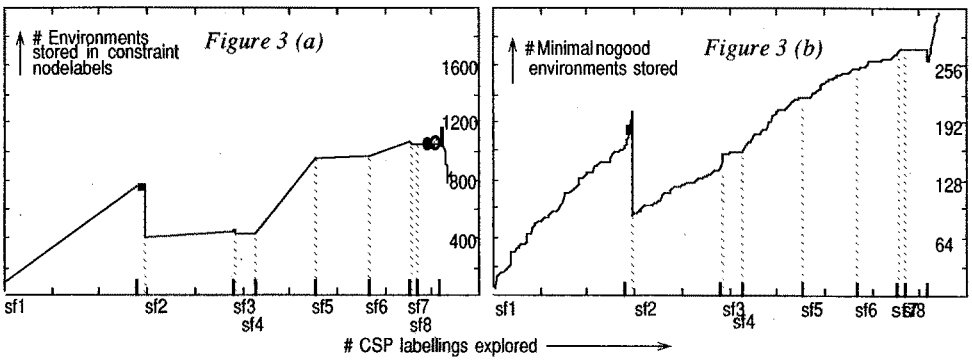

Figure 3: Storage of Environments for test image IM1

6.4 CPU costs of using the ATMS

A further disadvantage of the ATMS is the computational overhead incurred in maintaining the consistency of the justification network and environments. This can be estimated by recording the CPU time spent on ATMS operations. However, such measurements are strongly influenced by minor implementation details, such as the choice of data representation and the need for garbage collection. It is shown elsewhere [3] that in CARRS up to 73% of the time spent evaluating constraints concerns updating the ATMS to reflect newly discovered inconsistencies.

7. Conclusions

Our demonstrations based on the CARRS program have shown that an ATMS can be used effectively to determine all maximal consistent sets in a constrained labelling task. This approach overcomes a crucial defect inherent in the backtracking algorithm of Grimson and Lozano-Perez, and is therefore able to cope with the less well constrained problems of matching 2-d image features to 3-d models. In our experiments we have used relatively simple sets of features. Even here it has been shown that the recording mechanism of the ATMS allows approximately 50% of the constraint evaluations to be avoided. In more complex examples, having greater potential overlap between feature sets, the savings would be correspondingly greater.

We have also shown that the storage burden needed to maintain the ATMS is not excessive - this contradicts Provan's [13] findings for vision tasks such as that addressed by CARRS. One important factor in limiting the number of environments is the use of "success pruning", based on the ability to verify hypotheses by 3-d model-based methods. This has a dramatic effect both on the memory requirements and the up-dating costs of the ATMS. The most time-consuming component of CARRS is the maintenance of the consistency of environments. Meaningful CPU-time estimates are difficult, but our experiments suggest that the ATMS update time may completely cancel out the savings provided by the avoidance of constraint re-evaluation.

Our experiments on the consistent labelling problem using the general-purpose mechanism provided by the ATMS have allowed us to identify where significant savings can be made. Our findings show that the recording of partial results has a strong impact on the cost of the search, provided that an efficient means of maintaining a coherent record is used.

However, we note that the ATMS represents an "over-kill" for the labelling problem in CARRS. The properties of the ATMS which are important in this context may be implemented in a far more efficient way, especially if, as here, only binary constraints between labels are used. The success or failure of each constraint evaluation may be stored, as it is calculated, in a 2-d matrix allowing hashed indexing by means of the cue-labels. The breadth-first search used by the ATMS to explore the environment lattice can then be imitated by a systematic breadth-first expansion of the interpretation tree. Repeated evaluation of constraints can then be avoided by first checking each constraint against the evolving results matrix. We are currently investigating this strategy. It should be noted that this method will only work for binary constraints. To use higher order constraints a more general mechanism is needed - such as de Kleer's ATMS.

Acknowledgments

CARRS was developed under Alvey grant MMI-007, while all three authors were at the University of Reading. We are grateful to British Aerospace for continued support, which allowed this investigation to be completed.

8. References

[1] R.Bodington and P.Elleby "Justification and Assumption-based Truth Maintenance Systems: When and How to use them in Constraint Satisfaction," in *Reason Maintenance Systems and their Applications*, Ed. B.Smith and G.Kelleher, Ellis Horwood, 1988

[2] R.Bodington, G.D.Sullivan and K.D.Baker, "The Consistent Labelling of Image Features using an ATMS", *Image Vision & Computing*, vol. 7, no. 1, February 1989.

[3] R.Bodington, G.D.Sullivan and K.D.Baker, "Experiments on the use of the ATMS to label features for object recognition" *University of Reading, unpublished report* December 1989

[4] K.Brisdon, G.D.Sullivan and K.D.Baker, "Feature Aggregation in Iconic Model Evaluation," *Proceedings of the Alvey Vision Conference, AVC88*, Sept. 1988.

[5] J.Canny, "Finding edges and lines in images.", Ph.D. AI-laboratory, MIT, Cambridge, MA, 1983

[6] J.De Kleer "An Assumption-Based TMS", *Artificial Intelligence*, vol. 28, no. 2, March 1986.

[7] J.De Kleer "Extending the ATMS", *Artificial Intelligence*, vol 28, no. 2, March 1986.

[8] C.Goad "Special purpose automatic programming for 3-d model-based vision", *Proc. ARPA Image Understanding Workshop, 1983.*

[9] W.Grimson, and T.Lozano-Pérez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data." A.I. Memo 738, MIT, Cambridge, MA, August 1983.

[10] W.Grimson and T.Lozano-Pérez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, July 1987.

[11] W.Grimson, "The combinatorics of object recognition in cluttered environments using constrained search," *2nd International Conference on Computer Vision, ICCV88*, Dec. 1988.

[12] D.G.Lowe, "Three-dimensional Object Recognition from Single Two-dimensional Images.," *Artificial Intelligence*, no. 31, 1987.

[13] G.Provan, "Efficiency Analysis of Multiple-Context TMSs in Scene Representation." *Proceedings 6th National Conference on Artificial Intelligence, AAAI-87*, 1987.

[14] A.Rydz, G.D.Sullivan and K.D.Baker, "Model-based Vision Planar Representation of the Viewsphere,"*Proceedings of the Alvey Vision Conference, AVC88*, Sept. 1988.

[15] A.Worrall, G.D.Sullivan and K.D.Baker, "Model-based Perspective Inversion," *Proceedings of the Alvey Vision Conference, AVC88*, Sept. 1988.

[16] A.Worrall, G.D.Sullivan and K.D.Baker, "The Roll Angle Consistency Constraint" *Proceedings of the Alvey Vision Conference, AVC89*, Sept. 1989