[25] WPFM: The Workspace Prediction and Fast Matching System

Jonathan C Aylett and Robert B Fisher

Department of Artificial Intelligence
University of Edinburgh, Edinburgh EH1 2QL, UK

A Pat Fothergill

Department of Computer Science
University of Aberdeen, Aberdeen, UK

Reprinted, with permission of *Journal of Intelligent and Robotic Systems*, 1988, **1**, 185-201.

1. Introduction

An important area for the application of machine vision is to industrial assembly tasks, where the effective automation of such tasks is often dependent on the use of robots with sensing capabilities. The micro-world environment of a typical industrial assembly robot, although complex may also be highly constrained, with considerable information available *a priori* about objects and their disposition within a robot workcell. In order to maximise the efficiency with which a computer vision system can operate within this domain *a priori* information can be utilised to avoid extensive visual processing for constant scene features, or for objects in the scene for which there are accurate estimates of current position. The use of this information can then allow a reduction of the high computational cost of vision processing. This quantitative geometric information can be derived from a CAD based model of the robot workstation and from the known location of the camera system used to view the scene.

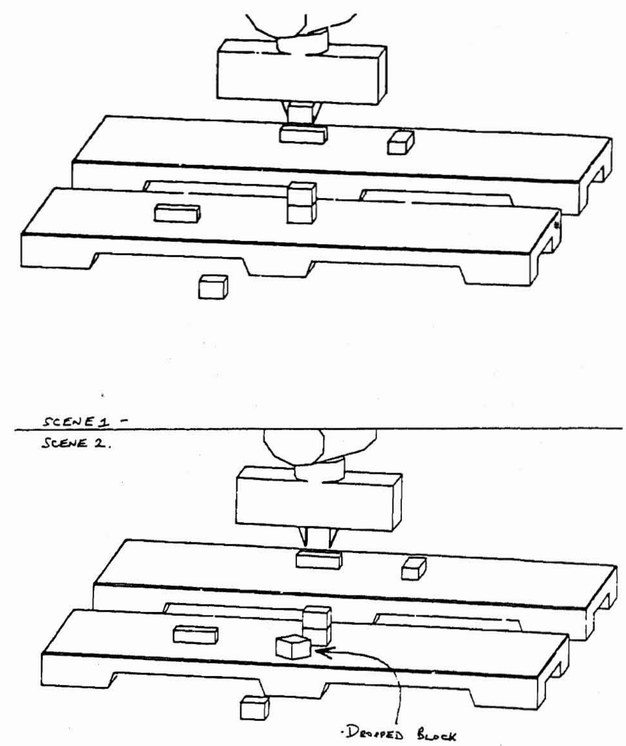

Figure 1 - A Typical Scenario

2. Workspace Prediction and Fast Matching

The *WPFM System* was developed to use *a priori* knowledge about a given scene so that some stereo data extracted from the scene could be quickly matched to model features, and then subtracted out of the scene data, leaving unknown data to be dealt with by a more comprehensive (and computationally expensive) vision matcher [9]. A schematic illustration of the type of target scenario is shown in Figure(1), where the only difference between the two modeled scenes is that in one case the robot gripper has "dropped it's block".

The *WPFM System* is composed of two major components, the off-line *workspace prediction system (WP System)* and the on-line *fast a priori matching system (FAPM System)*. The off-line prediction system was developed from the Robmod [7 & 8] solid modelling system and can produce a *workspace prediction graph (WP Graph)* for any modeled scene. The on-line fast matching phase consists of an edge based matching system, where stereo edge data derived from a real scene by a stereo vision system [5, 14], can be matched to model edge features in an appropriate *WP Graph.*

Although the *WP System* (and Robmod) is limited to a polyhedral representation of objects, curved edges can be modeled by approximating a curve with a series of linear tessellation edges. This then enables the matching of *curved* edge features extracted by the stereo process. The scene subtraction process implemented by the *WPFM System* is a "*symbolic subtraction*" of edge features, rather than an "*image subtraction*" of pixels, as an image subtraction process would generate image artifacts and other errors.

3. The Workspace Prediction Graph

Unlike boundary based modelling systems, CSG systems build objects by combining primitive solid shapes together, using boolean set operations. By including information about the way an object is constructed from these primitive solid shapes into a boundary representation, we construct a data structure containing the geometrical and topological information about an object, as well as a relational aspect which comprises a description of the *visually salient parts* that go to make up an object. This extends the concept of a boundary representation to include information specifically useful to vision processing [2] and is a development from the classical winged edge type structures used by Weiler [19] and others [4, 6, 10, 16 17, & 20]. Our variant has a hierarchical structure which composes distinct objects into *separate assemblies*, then decomposes them into *CSG primatives* (convex or concave), into *surfaces*, and then *edge boundaries* of the surfaces. The geometry is associated with edge, vertex and surface descriptions. A *Workspace Prediction Graph* can be produced by the modelling system analysing a CSG model of a scene from a particular viewpoint and adding the following visibility information to the boundary representation :

1. Total number of visible, part visible and non-visible edges.
2. Total number of visible, virtual and non-visible vertices.
3. Viewing parameters - viewing angles, viewed point and frame size.
4. Edges sorted by visibility type - visible, part-visible and non-visible. Some edges may be tagged as extremal, for "curved" primatives.
5. Vertices sorted by visibility type - visible and non-visible.
6. Virtual vertices. These are produced at T junctions where model edges are partially occluded.

Generally, the edges of the facets used to model curved surfaces by polyhedral approximation are treated as non-visible edges, as these are features which are not seen in real objects. However *extremal* edges are represented, as they correspond to a visible facet edge bounding the first non-visible facet. Although the *FAPM System* only requires the use of the edge and vertex features in the *WP Graph*, we produce a more complete representation [2] which could be adapted to be used with a more general purpose vision system, or in an enhanced *FAPM System*.

4. GDB Data

The details of the operation of the *GDB System* and the data structures produced by it can be found elsewhere [14]. The fast matching system requires the 3D geometrical descriptions of linear data segments and circular arcs contained within GDB data to perform the matching process.

5. Problems of Matching

Although we use *a priori* information about an expected scene, there will not be a perfect correspondence between the features extracted from the scene by the *GDB System* and the features produced by the *WP System* because of extra, missing or different features occurring in the *WP Graph* and the *GDB data*. These imperfections can be caused by :

Imperfections in the WP Graph

1. Simplification of the shapes of the objects when modelled in order to be able to represent them in the modelling system.
2. The WP graph may include features which are too small to recovered by the vision system.
3. Incorrect locations of a some model features, caused by objects with locations different to those predicted.
4. Missing model features, because the system will not attempt to predict features arising from objects with unknown locations.

Imperfections in GDB data

1. Features may not be appear in the data because lighting conditions can make them invisible.
2. Extra features can also be produced by the GDB system, caused by reflectance changes, shadows (shadow edges), specularities or texture.
3. Features may be occluded in the image that were predicted to be visible.
4. A single feature may be fragmented by an unpredicted partial occlusion.
5. Data can be fragmented by imperfections in an image, such as noise.
6. Tangential occluding boundaries on curved surfaces are difficult to resolve accurately by a stereo process and may be placed inaccurately, particularly in depth.
7. Linear edge features parallel to the plane of the stereo camera system are difficult to resolve stereoscopically and may also be placed inaccurately.
8. Curved edge features may be inaccurately segmented into several linear data segments.

6. Fast A Priori Matching

The *FAPM System* matches data input from a *GDB File* and model input from a *WP Graph.* Matching proceeds by pairing data to model edges using geometrical algorithms. The objective is to find all data segments that match the model, and to determine how well the data matches the model features. Tolerance values are required for this matching process and the choice of values for these parameters depends on the quality of the data. The tolerance parameters are :

1. The *maximum divergence angle* between linear model and linear data segments.
2. The *maximum tessellation divergence angle* between model tessellation and linear data segments within which the model curve (approximated by a linear tessellation) and data segment are considered to be parallel.
3. The *maximum perpendicular divergence angle* between the axis of a circular arc in the data and a "curved" model segment.
4. The minimum proportion of the data segment to model segment *overlap margin.* This defines the length of data segment allowed to fall outside a model edge.
5. The maximum *separation distance* between a linear model edge and a linear data segment at the point of closest approach.

After setting the calibration and tolerance parameters, model edges can be matched against data segments as follows :

1. For each data segment, determine if it lies within a *model circumsphere,* and if not, mark as unmatched.
2. For each model edge, scan all data segments and attempt a match. Mark paired data segments and model segments as matched, and for each model edge, record a total accumulated length of data segments matched.

Stage 1 of the process acts as a coarse filter, and thereby improves performance. The circumsphere surrounds the modelled object, and is generated by Robmod as part of the WP Graph. For speed, the test used is a *quick point test* on each data segment. Data segments have both endpoints tested to determine if one or both lie within the circumsphere. This is a conservative test as it does not remove all features located too far from the model, because the circumsphere does not "fit tightly" (it is not a convex hull), and approximate calculations are used to minimise the amount of computation required. However, the test can remove a considerable number of data segments quickly from the matching process and it becomes significant as the number of distinct objects in a scene increases.

Stage 2 is the main matching process. Each model edge is matched against all the remaining data segments. For linear model edges, there is allowed only a unique pairing of data segments to model edges, such that if one data segment is matched to a linear model edge, that data segment cannot be matched to another model feature. The data segments are paired to the first matching model edge, and a "*best fit*" test for any competing model edges is not applied, as this situation should only occur with very loose matching tolerances or with a data/model mismatch. Hence for m model edges and d data segments, this matching process is of the order of $(m*d)/2$ complexity, although this is reduced by the use of the circumsphere test or increased as the number of "curved" model edges increases (as these do not have a unique pairing relationship to data segments). The computational complexity of the matching process is kept within these bounds, as it is not necessary to estimate a reference frame transformation to register model to data [12 & 18], or to select which model to match to data by a process of model invocation [9], as this information is determined *a priori.*

6.1. Matching Linear Features

The matching process for each linear model edge is subdivided into four tests, and data is rejected at the first test in the sequence that fails. Linear edge segments in the model are represented by ($\mathbf{v_m},\mathbf{m_1},\mathbf{m_2}, M_l$) and in the data by ($\mathbf{v_d},\mathbf{d_1},\mathbf{d_2}, D_l$), where the four components are the unit direction vector $\mathbf{v_m} = \mathbf{m_2} - \mathbf{m_1}$, the two endpoints and the length. The tests are as follows :

1. *Quick point test.* Tests if either point on the data segment has an approximate separation distance from a point on the the model edge greater than the length of the model edge. The test will succeed if all the following conditions are true :

$$|m_{1i} - d_{1i}| < M_l$$

$$|m_{1i} - d_{2i}| < M_l$$

$$|m_{2i} - d_{1i}| < M_l$$

$$|m_{2i} - d_{2i}| < M_l$$

   where $i \in [1,2,3]$, the vector components.

2. *Angular deviation test.* Tests if the angle of separation of the direction vectors of the data segments and model edges is within the specified tolerance angle. For a maximum deviation angle D_a the test will succeed iff

$$|\mathbf{v_m}.\mathbf{v_d}| > \cos(D_a)$$

3. *Sub-segment test.* Tests if the projection of a data segment onto a model edge falls within that model edge. A tolerance parameter requires the data to overlap the model edge by a fractional proportion of the model edge length, and for a linear data segment both endpoints must fall within this interval. The overlap parameters are given by :

$$O_1 = (\mathbf{d_1} - \mathbf{m_2}).\mathbf{v_m}$$

and $O_2 = (\mathbf{d_2} - \mathbf{m_2}).\mathbf{v_m}$

And the test will succeed iff

$$-F_oM_l \le O_1 \le M_l + F_oM_l$$

and $-F_oM_l \le O_2 \le M_l + F_oM_l$

Where F_o is the overlap proportion tolerance parameter.

4. *Minimum distance test.* Tests to determine if the closest point between the the data segment and the model edge is within the specified separation distance limit. This test will succeed iff

$$|(O_1\, \mathbf{v_m}) + \mathbf{m_2} - \mathbf{d_1}| < S_d$$

and $|(O_2\, \mathbf{v_m}) + \mathbf{m_2} - \mathbf{d_2}| < S_d$

where S_d is the minimum separation tolerance parameter, and O_1 and O_2 are as defined above.

The algorithms used in tests (2-4) above were derived from algorithms originally developed by Watson [18] for use in a wire frame based vision matching system, and are similar to those used in other combinatorial matching systems [11 & 12]. A linear data segment that passes all of the above tests will be paired to the linear model edge and the uniqueness criterion will be enforced.

6.2. Matching Data and Model Tessellations

There are two further cases of matching, both of which involve the matching of a "*curved*" model edge to data. As curved model edges are represented in the WP Graph by a tessellation of the curve into a number of smaller linear edges, it is these *tessellation edges* that are matched to the data. The two cases differ in that curved features may be segmented by the GDB as *linear* or *circular arc* data segments. Hence in the first case, the GDB tessellates curved data into a number of linear data segments, whilst in the second case circular features are segmented into one or more circular arcs. To match model tessellations to data tessellations, we use a similar (but not identical) method as used above for the linear case. This matching process is :

1. *Quick point test,* similar to test(1) above, except that only one pair of conditions is required to be true. Hence the test will succeed if :

$$|m_{1i} - d_{1i}| < M_l$$

and $|m_{1i} - d_{2i}| < M_l$

or $|m_{2i} - d_{1i}| < M_l$

and $|m_{2i} - d_{2i}| < M_l$

2. *Angle test* as test(2) above, although the maximum deviation angle is normally set to a larger value than used for the linear case, to allow for possible extra divergence between model and data segments which can occur if the tessellation of the data and the tessellation of the model are out of step.

3. *Sub-segment test* as in test(3) above, except to allow for the possible "*stepping*" problem between tessellations, we use initially a zero overlap and require only one endpoint to overlap. If both overlap, the process is as before, otherwise we then run a subsidiary test to determine how much of the *data segment* overlaps the model edge. This subsidiary test is passed if a sufficient proportion of the data segment overlaps the model edge, (an overlap of 30% was used). In some cases, the data segment can be larger than the model edge. This is usually caused by the GDB segmenting a curve into one rather than several linear segments which will then correspond to several tessellation edges in the model. For this case, the algorithms are as described above, except that the relationship between a model edge and a data segment is reversed as several model edge segments will be paired to one data segment.

4. *Minimum distance test* as in test(4) above, unless the overlap is only partial, in which case one minimum distance is tested corresponding to the one valid overlap.

Due to the "*stepping*" problem described above, the uniqueness criteria cannot be applied in this matching process. To minimise the possibility of incorrect pairings between model and data occurring, the linear model edges are matched to the data before the tessellated model edges. The lower accuracy of this matching process is partially offset by the stronger pairing relationship implied by a match between a curved model edge and a curved data segment.

6.3. Matching Curved Features - Data Arcs

A circular arc in the GDB is represented by (R_l, $\mathbf{c_c}$, $\mathbf{d_1}$, $\mathbf{d_2}$) where R_l is the radius of the arc, $\mathbf{c_c}$ the centre point and $\mathbf{d_1}$ & $\mathbf{d_2}$ the two endpoints of the arc. To match a model tessellation edge to a circular arc, the test sequence is as follows :

1. *Quick point test* as in test(1) above, except that the point distance is the radius and the data point used is the centre of the circular arc.

2. *Angular deviation test.* The normal to the plane in which the data arc lies is compared to the direction vector of the model edge. It matches if these are perpendicular to each other to within the maximum

deviation angle.

The unit direction vectors for the two radii are given by :

$$r_1 = (d_1 - c_c) / R_1$$

$$r_2 = (d_2 - c_c) / R_1$$

and the unit normal vector n_d to the plane of the arc is given by :

$$n_d = r_1 \times r_2 / ||r_1 \times r_2||$$

For a maximum deviation angle D_a the test will succeed iff

$$|v_m . n_d| < \sin(D_a)$$

3. *Minimum distance test.* Determines if the end points of the model edge fall within the radius distance from the centre of the data arc segment, within the separation distance limit. For a separation distance limit S_d the test will succeed iff

$$|(||(c_c - m_1)|| - R_1)| < S_d$$

and $|(||(c_c - m_2)|| - R_1)| < S_d$

4. *Circular arc sub-segment test.* This determines if the model edge overlaps the data arc, by intersecting the two angle ranges of the two model edge endpoints with the angle range of the data arc. The test passes if the required proportion of the angle range of the model tessellation edge is found to overlap the data arc, (the required overlap was set to 30%). The overlap between the data and model angle ranges is given by :

$$O_a = | [0, A_a] \cap [A_{m1}, A_{m2}] |$$

and the test will succeed iff

$$|O_a / (A_{m1} - A_{m2})| > 0.3$$

(See [3] for further details of this algorithm).

In some cases, the data arc may be smaller than the model tessellation edge, usually due to data fragmentation. For this case, the test is the same, except that 30% of the *data arc* is required to overlap the model edge for the test to succeed.

Although the algorithms used for matching circular data arcs differ from the previous case of matching tessellation to tessellation, the uniqueness constraint still does not apply in the matching process, as again one data segment may be required to be matched to several model tessellation edges. However, the problem of a lower resolution in the matching process is reduced as the "*stepping*" problem encountered with tessellation to tessellation matching does not arise.

7. Results of Matching

At the end of the matching process an updated GDB file is output, with indicators of which data segments have been matched, as well as graphics indicating which model edges and data segments have been matched, (see Figures(2c-2i)). A matching "*goodness*" summary is also produced to indicate how close a match was obtained between model and data and this can then be used to determine whether a match is acceptable, and which data segments can be safely ignored in further vision processing.

8. Experimental Results

The workspace prediction and fast matching system has been implemented in C and run on our SUN computers, although not yet in conjunction with the overall vision system, which is the eventual objective. We have run the system on several test cases, the results of one of which is given below. The test data used for this was from a set of stereo images provided by AIVRU Sheffield and had an image resolution of 256 by 256 pixels. The imaged scene contained a single known test object in a cluttered environment and we attempted to match only this test object at a predicted position. The test object was a moderately complex engineering type object generally known as the "*widget*". The dimensions of the rectangular base of this test object were (20, 50, 70)mm.

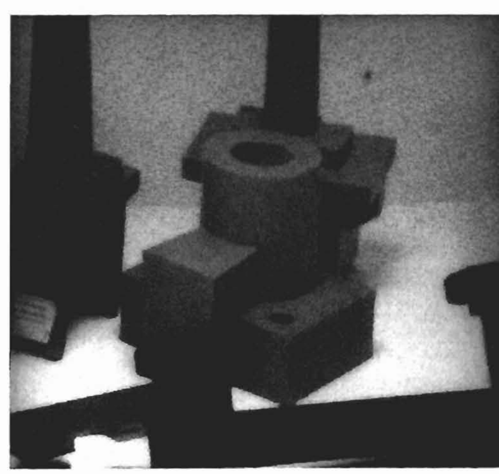

Figure 2a

One of the stereo image pairs of this scene is shown in Figure(2a) and the 3D GDB data extracted from this in Figure(2b).

Although, the matcher requires *a priori* the known location of objects, this was not available for this data and had to be estimated, so it is therefore not exact. The tolerance values used in the matching process were :

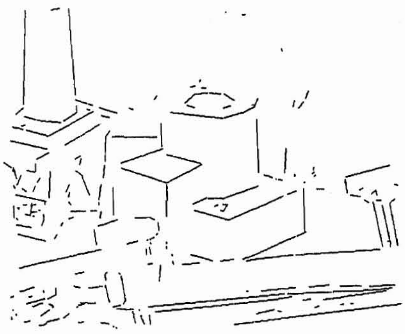

Figure 2b

Maximum divergence angle: 0.25 Radians.

Maximum tessellation divergence angle: 0.30 Radians.

Perpendicular divergence angle (circular arcs): 0.20 Radians.

Minimum Overlap Proportion: 90%.

Maximum separation distance limit: 18mm

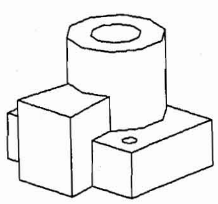

Figure 2c

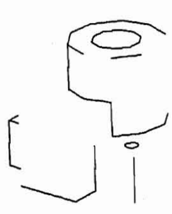

Figure 2d

The matching sequence between model and data is illustrated in figures(2c-2i), with the *WP Graph* to be matched (2c), the unmatched model edges (2d), the unmatched data segments (2e), the matched model edges (2f) and the matched data segments (2g). Lastly, the data and model are shown overlaid (2h), and the data segments outside the model circumsphere are shown (2i). This matching process ran in 0.75 seconds on a SUN-3

Figure 2e

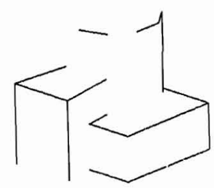

Figure 2f

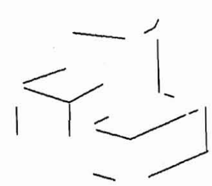

Figure 2g

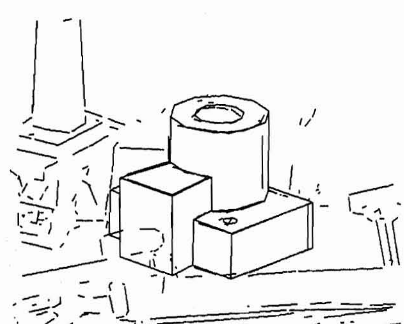

Figure 2h

Figure 2i

computer producing the following matching summary :

59 MODEL EDGES PREDICTED VISIBLE

19 MATCHED MODEL EDGES

32% OF PREDICTED MODEL EDGES MATCHED

830mm OF MODEL EDGES PREDICTED VISIBLE

464mm OF MODEL EDGES MATCHED

320mm OF DATA SEGMENTS MATCHED

69% OF MATCHED MODEL EDGE LENGTH MATCHED TO DATA

39% OF PREDICTED EDGE LENGTH MATCHED TO DATA

Of the model edges predicted to be visible 32% were matched to data, and 69% of the total length of these matched edges was accounted for by the total matched data length. The total matched data length also accounted for 39% of the total length of the model edges predicted to be visible. We can see that the circumsphere test was useful in eliminating much of the data from the matching process and enabling a faster run time (Figure(2i)).

Model matching failed in several cases because of problems with the GDB data. No data was produced by the GDB System for a large part of the top of the cylindrical projection of the widget as stereo processing failed to recover this (although edge detection did extract it). There was a similar problem for the left front corner of the base and for the vertical edge on the right front corner. The equivalent model features were therefore not matched. Data loss was also caused by the occlusion of parts of the widget by other objects and this contributes to the low figure for the proportion of model edges predicted visible and matched (32%). In practice a model of the entire scene would be constructed. This would then reduce the number of model edges and the total edge length predicted visible for the widget, increasing the figures for the proportions matched. This low figure is also explained by the failure to match some curved features. As each model curve is tessellated into several linear edges a large number of these can fail to be matched if only a few curved features are not matched. The curved features were not matched as they were either too small to be resolved by the GDB, or were segmented into tessellations which were inaccurate rather than into circular arcs. Matching with circular arcs has been successfully tested [3], and this would have avoided the problems associated with the tessellated data. Despite these problems, the matching system performed well, with nearly all (visible) linear data segments being matched.

9. Conclusion

The purpose of the *WPFM System* we have described is to identify areas of local discrepancy between the *GDB data* and *WP graph* as quickly as possible so that attention may be centred on them by a full vision system. Thus unmatched data may indicate a local area of discrepancy, which combined with information on any expected model features not matched, could indicate to the full vision system that objects have moved, or are missing. However, some unmatched data will be due to noise and other distortions in the data. The matching goodness summary, combined with the use of the full topological and geometric information in the *WP Graph*, could be used by the full vision system to disambiguate this noise and distortion from real differences in expected scenes.

The system has been implemented and run on synthetic data generated by the *Winsom* solid modelling system [15] and on real test data [3], which it can process rapidly to produce a reasonable result. Matching performance with real data will be improved as the resolution of the GDB system is increased by the use of improved calibration techniques and better camera systems. Better data is already provided by the latest version of the GDB System, and this will be used in future testing.

Acknowledgements

Funding for this work was under Alvey grant GR/D/1740.3. Vision test data was provided by T P Pridmore and S B Pollard of AIVRU Sheffield, who also provided help and advice with the GDB System.

References

[1] AIVRU (1986). GDB Release (1.0) User Documentation. AIVRU report 014, University of Sheffield.

[2] Aylett. J.C. (1987). A Boundary Representation for Machine Vision, Working Paper, University Of Edinburgh. (In preparation).

[3] Aylett. J.C. (1987). WPFM: The Workspace Prediction and Fast Matching System, Working Paper, University Of Edinburgh.

[4] Baer, A. et al. (1979). Geometric Modelling: A Survey. Computer Aided Design, Volume 11 Number 5.

[5] Blake, A. and Mayhew, J.E.W. (1985). Alvey 2-1/2D sketch project: Proposed structure for a development sys-

tem. AIVRU report, University of Sheffield.

[6] Boyse, J. W. (1982). GMSolid: Interactive Modeling for Design and Analysis of Solids. IEEE Computer Graphics and Automation, Mar. 1982, pp. 27-40.

[7] Cameron, S.A. (1984). Modelling Solids in Motion, PhD Thesis, University Of Edinburgh.

[8] Cameron, S.A., Aylett, J.C. (1986). Robmod User Guide, DAI Software Paper, University Of Edinburgh.

[9] Fisher, R.B. (1986). From Surfaces to Objects, PhD Thesis, University of Edinburgh.

[10] Gray, M. (1985). Proposal for Representing 3-D Objects. IBM UK Scientific Centre.

[11] Grimson, W. E. L. and Lozano-Perez T. Model-Based Recognition and Localisation from Sparse Range or Tactile Data. The International Journal of Robotics Research, Vol. 3, No. 3, Fall 1984.

[12] Pollard, S. B. et al. (1986). Matching Geometrical Descriptions in Three Space. AIVRU Working Paper 22, University of Sheffield.

[13] Popplestone, R.J., Ambler, A.P. and Bellos, I.M.. An Interpreter for a Language Describing Assemblies. Artificial Intelligence, Vol. 14, pp. 79, 1980.

[14] Pridmore, T.P., Bowen, J.B., Mayhew, J.E.W. (1986). Geometrical description of the connect graph (version 2). The geometrical base description : a specification. AIVRU report 012, University of Sheffield.

[15] Quarendon, P. (1984). WINSOM User's Guide (UKSC 123). IBM UK Scientific Centre.

[16] Requicha A. A. G. (1980). Representations for Rigid Solids: Theory, Methods and Systems. Computing Surveys, Volume 12, No 4.

[17] Requicha A. A. G. and Voelcker, H. B. (1982). Solid Modeling: A Historical Summary and Contemporary Assessment. Computing Surveys, Volume 12, No 4.

[18] Watson, R. (1985). An Edge-Based 3D Geometrical Model Matching System. MSc Dissertation, Department of Artificial Intelligence, University of Edinburgh.

[19] Weiler, K. (1985). Edge Based Data Structures for Solid Modeling in Curved Surface Environments. IEEE Computer Graphics and Automation, Jan. 1985, pp. 21-40.

[20] Wilson, P. R. et al (1985). Interfaces for Data Transfer Between Solid Modeling Systems. IEEE Computer Graphics and Automation, Jan. 1985, pp. 41-51.