Object Recognition Using Local Geometric Constraints: A Robust Alternative To Tree-Search

Alistair J Bray
University of Sussex
Janet: alib@cogs.susx.ac.uk

Abstract

A new algorithm is presented for recognising 3D polyhedral objects in a 2D segmented image using local geometric constraints between 2D line segments. Results demonstrate the success of the algorithm at coping with poorly segmented images that would cause substantial problems for many current algorithms. The algorithm adapts to use with either 3D line data or 2D polygonal objects; either case increases its efficiency. The conventional approach of searching an *interpretation* tree and pruning it using local constraints is discarded; the new approach accumulates the information available from the local constraints and forms match hypotheses subject to two global constraints that are enforced using the *competitive paradigm.* All stages of processing consist of many extremely simple and intrinsically parallel operations. This parallelism means that the algorithm is potentially very fast, and contributes to its robustness. It also means that the computation can be guaranteed to complete after a known time.

Many algorithms have been developed in recent years that rely upon dynamically growing and pruning an interpretation tree that describes the correspondence between object features and image features. The features may be 3D [1, 2, 3] or 2D [4, 5, 6]; likewise, objects may be 3D polyhedra or 2D polygons. For example, Goad's algorithm matches 2D line segments in the image to 3D segments describing the object; the algorithm is an efficient one given a low degree of noise in the image and a good segmentation, and has been well-tested in a commercial environment. This work examines such algorithms based upon the tree-search formulation with respect to *robustness* and *speed* under noisy conditions, and concludes that their performance does not *degrade gracefully.* A new formulation is presented that exploits the same information in the image, but is inherently parallel and transforms the problem from one of tree-search to signal detection.

The search-tree paradigm

In this paradigm possible matches define a *search tree* which can be pruned using local constraints. A solution-state is defined by a leaf-node of the search tree. Heuristics can provide an efficient way of searching this tree to find solutions quickly and so providing

a sophisticated *sequential* traversal of the search tree. For example, Goad describes describes various ways his algorithm can be unwound and how optimal search paths can be pre-computed for models off-line; these are all extensions and modifications to what is essentially a *depth-first sequential search* [4, 5]. There are weaknesses with this formulation in terms of the sensitivity to image noise, the speed of the search, and the unpredictability of the search times involved:

Sensitivity to Noise

Tree-search is sensitive to error in the segmentation of the image because *one* incorrect evaluation of a local constraint is sufficient to prevent a correct solution being found. Local constraints will be described later (See Section 3) that *can* cope with the type of error in the data commonly associated with occlusion, poor lighting or bad edge-detection. However, such measures are ineffective when coping with the problem of image features being wholly obscured eg. if predicted lines are not present the algorithm becomes inefficient. Solutions to this problem have been suggested such as the *Null Face Hypothesis* [4, 3]. However, these solutions lead to a large expansion of the search-tree and are therefore undesirable. Such algorithms are demonstrating *ungraceful degradation*; a single missing line causes substantial problems for the search, and multiple missing lines makes the search space hopelessly large.

Speed of Computation

Tree-search algorithms are essentially sequential and well-suited to conventional Von Neumann machines. However, given the best of optimisation techniques we can only expect a sequential algorithm to achieve certain levels of speed. On the other hand, if an algorithm is inherently parallel the potential speed of a parallel implementation is extremely high; this sort of parallelism can be exploited by many modern architectures today and it seems certain that it is the way that hardware is progressing. It is certain that a tree-search can always be performed using a parallel algorithm in such a way as to exploit parallel facilities. However, this post-hoc exploitation of potential parallelism is usually harder to control and less flexible than when dealing with an *intrinsically* parallel algorithm; time-gains are not always as great as anticipated.

Predictability

Despite the effect of pruning, the size of the search tree involved, given noisy data, can be quite substantial. The time taken for a correct solution to be found can vary enormously depending upon where in the tree it lies and how the search is guided. Heuristics can be exploited for guiding the search eg. ways of pre-computing optimal search strategies, unwinding the search tree. These can be very effective. However, they are only heuristics and, by their nature, will fail to prove effective some of the time. They are also often model-specific, and the pre-computation of the optimal search path for a new model is itself time-consuming. There is therefore an element of *uncertainty* in whether the system will arrive at a correct solution within a given time. This can cause problems in a real-time system that requires the best solution available after a given degree of processing. It is also

true that many heuristics require parameters to be set and how to determine the optimal setting for these parameters is often opaque; trial and error is often ineffective and always time consuming. *How many failures to find a predicted line should be tolerated before pruning a branch of the tree? How many matches are sufficient for a correct match? How many matches must be found before search is terminated?* The answer to these questions is not obvious and it would be far preferable if the questions could simply be side-stepped.

Aims of New Formulation

It is apparent that a serial search algorithm can be optimised, but still encounters critical problems in the above areas. It might be hoped that a new parallel formulation can be derived that exploits to the full the advantages of a parallel approach. If this can be achieved then the algorithm might possess the following properties:

1. **Graceful Degradation:** The ability to compute the correspondence set and an estimation of object orientation with a degree of accuracy comparable to the accuracy of the data.

2. **Predictability:** The ability to compute the correspondence set and an estimation of object orientation within a particular fixed time.

3. **Speed:** The ability to compute the correspondence set and an estimation of object orientation in very short periods of time.

4. **Heuristic Independent:** The ability to compute the correspondence set and an estimation of object orientation in a way that does not rely heavily on *ad hoc* parameter determination.

1 New Formulation

The new formulation consists of four stages of processing:

1. **Accumulation:** In the accumulation stage all possible binary relations between image segments are compared with all possible binary relations between model segments over all possible viewpoints. If the two relations are consistent then support is accumulated for the two assignments of model to image that the relations represent.

2. **Cooperation:** Once accumulation is complete there is a cooperative stage in which support in the accumulators is distributed between locally connected regions of the viewsphere.

3. **Competition:** In the competitive stage support is distributed *within* the accumulators for each viewpatch so as to enforce uniqueness constraints ie. only one model line can match one image line and vice versa. The result is an accumulator with a small number of high peaks that corresponds to an interpretation for that viewpoint.

4. **Signal Extraction:** The final stage of signal extraction finds the interpretation for each accumulator and finds the largest maximal clique within the interpretation that is wholly consistent in terms of the local constraints. These cliques represent the best interpretation for that viewpoint. Those of sufficient size are ordered according to the signal/noise ratio in the accumulator and then passed to the model test stage in the usual way.

1.1 Data Structures

Consider an image $[I]$ consisting of $\mathcal{I}$ 2D line segments and a model $[M]$ consisting of $\mathcal{M}$ 3D line segments. Let the Viewsphere be quantised into a set of $\mathcal{V}$ viewpatches – $[V]$. For each viewpatch a subset of $[M]$ will be visible. Consider a single viewpatch – V_v where $1 \leq v \leq \mathcal{V}$; for V_v a binary constraint array can be constructed that has dimension $\mathcal{M} \times \mathcal{M}$ – call this MA_v. An element of MA_v, $MA_{v(m,m\prime)}$ where $1 \leq m, m\prime \leq \mathcal{M}$, describes the local relationship between model line M_m and model line $M_{m\prime}$ over viewpatch V_v. If either M_m or $M_{m\prime}$ is obscured at V_v then $MA_{v(m,m\prime)}$ will be empty; otherwise it will contain the bounds upon the binary constraints between M_m and $M_{m\prime}$ over V_v. Of course all diagonal elements $MA_{v(m,m\prime)}$ where $1 \leq m = m\prime \leq \mathcal{M}$ will also be empty as they represent the relation between a model segment and itself. Hence MA_v will be a sparse array due to visibility constraints; also, if the local constraints being used are symmetric then only a triangular half of the array need be stored ie. $MA_{v(m,m\prime)}$ where $1 \leq m < m\prime \leq m$.

In a like manner a single binary constraint array can be constructed for the image segments – call this IA. This contains the actual values for the binary constraints between any pair of image segments. If the constraints being measured rely upon the segments being *directed*, then this will be of dimension $\mathcal{I} \times 2 \times \mathcal{I} \times 2$; otherwise it will have dimension of simply $\mathcal{I} \times \mathcal{I}$. This is basically decomposing an undirected image segment into two possible directed image segments of opposite directions – this effectively doubles the size of the image. Since the two constraints to be used both use directed image segments we will assume a dimension of $\mathcal{I} \times 2 \times \mathcal{I} \times 2$ from here on. This array will not be sparse except that the diagonal elements $IA_{(i,d,i\prime,d\prime)}$, where $1 \leq i = i\prime \leq \mathcal{I}$ and $d, d\prime \in [1,2]$ will be empty since they represent the relationship between an image segment and itself.

For the same viewpoint V_v it is also possible to construct a *Correspondence* array. Let us call it CA_v. This array will act as an *accumulator*. It will store information concerning possible correspondences between model and image segments for V_v. It will therefore have dimensions of $\mathcal{M} \times \mathcal{I}$. Since it acts as an accumulator each element is an accumulator cell such that a high count in $CA_{v(m,i)}$ where $1 \leq m \leq \mathcal{M}$ and $1 \leq i \leq \mathcal{I}$, implies strong support for model segment m corresponding to image segment i in Viewpatch V_v.

1.2 Updating The Accumulators

Given the above representation of data-structures it is easy to see how the $\mathcal{V}$ accumulator arrays can be updated. Given a particular viewpatch V_v, each non-empty element of IA – $IA_{(i,d,i\prime,d\prime)}$ – can be checked against each non-empty element of MA_v – $MA_{v(m,m\prime)}$ – to give a boolean output that will depend upon whether the values of $IA_{(i,d,i\prime,d\prime)}$ fall inside the bounds defined by $MA_{v(m,m\prime)}$. If $IA_{(i,d,i\prime,d\prime)}$ is consistent with $MA_{v(m,m\prime)}$ then the two cells $CA_{v(m,i)}$ and $CA_{v(m\prime,i\prime)}$ in the accumulator array CA_v can be incremented. If it is

inconsistent then no update occurs. Therefore for V_v the number of possible updates to CA_v is equal to the twice the product of the number of elements in MA_v and the number of elements in IA. In practice it will be much less than this since updates only occur for consistent matches.

1.3 Competition and Co-operation

Let us assume that the accumulation stage is complete on the $\mathcal{V}$ accumulator arrays. The problem is how to extract the correspondence set and an estimation of viewpoint from these arrays. This is a signal detection problem. Each accumulator array consists of a mixture of signal and noise. At viewpoints far away from the correct one the signal is more or less non-existent; as the viewpoint tends towards the correct one the signal becomes much stronger and it is expected that it should be detectable from the background noise. In this sort of problem it is the ratio of signal to noise that is significant. The signal is defined by correct correspondences satisfying the local constraints, and the noise is defined by incorrect correspondences satisfying the constraints by chance. For instance, error in sensor data will lead to a reduction in *signal*; irrelevant image segments on the other hand lead to and increase in *noise*; missing image lines will again lead to signal reduction.

Competition

However, there is further information that can be applied to the accumulators that helps to amplify the signal and reduce the noise. Two assumptions can be made which allow the competitive paradigm to be applied within each accumulator array. The assumptions are:

- **No image segment can correspond to more than one model segment**
- **No model segment can correspond to more than one image segment**

The first of these assumptions is certainly one that we would wish to incorporate in a system, since it is only under very exceptional circumstances that an image line corresponds to more than a single model line. The second is not so certain since a model line can produce more than a single image line; this may occur in cases of occlusion or poor edge-detection. As will be seen, this formulation provides an elegant way of incorporating the two different constraints in terms of competition in two orthogonal directions.

The accumulator has both a model and an image dimension. It is simple to implement a form of competition independently in each of these dimensions. Consider the raster in the accumulator defined by $CA_{v_{(m,K)}}$ where K is a constant ($1 \leq K \leq \mathcal{I}$) and $1 \leq m \leq \mathcal{M}$. This raster represents the support for the various model lines matching image line K at viewpoint v. Since only one model line is allowed to match that image line competition can be introduced between the elements of this raster. That is to say, one of these cells will *win* at the expense on the other $\mathcal{M} - 1$ lines. If competition was in the model dimension alone this competition would be simple to implement. Obviously the element in the raster with the highest count would win, and maybe if this count was insufficient then there would be no winner. However, there is also similar competition in the image dimension ie. those rasters defined by $CA_{v_{(K,i)}}$ where K is a constant ($1 \leq K \leq \mathcal{M}$)

and $1 \leq i \leq \mathcal{I}$. Therefore a way on implementing this competition in both dimensions simultaneously must be adopted that provides a solution consistent with both the above assumptions.

Co-operation

There is one more assumption that can be incorporated into this formulation. So far we have quantised the viewsphere into $\mathcal{V}$ viewpatches. We have then considered each of these independently. In doing this we are ignoring the topology of the viewsphere. As stated above, we expect the degree to which the signal will dominate over the noise in a particular viewpoints accumulator to be correlated with how close that viewpoint is to the correct viewpoint. It is therefore to be expected that traces of the signal will not only show up in the correct viewpatch but also in neighbouring viewpatches. It is expected that the signal will trail off as the viewpoint moves away from the correct one. It is therefore possible to introduce a degree of topographical cooperation between like cells of the accumulator for neighbouring viewpatches. In this way the signal is being picked up over a larger portion of the viewsphere.

1.4 Global Consistency

One of the main problems with the approach as formulated is that of *global inconsistency.* There can be two types of global inconsistency:

- **Within Local Constraints:** An interpretation is globally inconsistent *within* local constraints when one or more of the correspondences is inconsistent with one or more different correspondences in terms of the local constraints used.
- **Beyond Local Constraints:** An interpretation is globally inconsistent *beyond* local constraints when all the local constraints between correspondences are satisfied, and yet there is still no single 3D position of the object that will map each model feature onto each image feature.

In the tree search formulation, the interpretation is guaranteed to be globally consistent within the local constraints, and consistency beyond local constraints is guaranteed by some form of model test. In the formulation proposed here local consistency within constraints is *not* guaranteed; inconsistent correspondences may result from incorrect peaks resulting from noise. In such interpretations it may be that many of the correspondences are correct but a minority are not (eg. along rasters corresponding to missing model lines). This minority is easily sufficient to force the model test to fail since these matches may be radically wrong in terms of the geometry of 3 space [1]. One solution to this problem is to have some form of filtering in the model test that removes matches that are preventing convergence. It is to be expected that this would be computationally expensive. A better solution is to ensure consistency within local constraints by finding *maximal cliques.*

[1] A model test performing a least squares solution can cope with error as long as incorrect matches are "close" to correct. Interpretations that are consistent within constraints usually meet this criterion of closeness. However, those that are inconsistent in terms of local constraints often fail to meet this criterion and subsequent *convergence* is impossible

The result of the competitive layer is an interpretation that is expected to be largely consistent within constraints but not totally so. This can be represented as a graph in which each node is a correspondence and a connection represents consistency. The problem of ensuring global consistency within constraints is reduced to that of finding maximal cliques in the graph. The study of this problem is well-developed, and given that the interpretation in expected not to be that large (it is bounded by $max(\mathcal{I}, \mathcal{M})$), there are very efficient algorithms for finding these cliques. Bolles & Cain [7, 8] discuss this problem in the domain of model-based vision, and state that they have found Johnston's algorithm [9] to be quite satisfactory. It is to be expected that the largest maximal clique will represent the correct interpretation, although smaller cliques could be considered worth investigating if the model test still fails.

1.5 Diagrammatic Representation

The above formulation is represented in Figure 1. The lowest layer represents the input or stimulus – this is an $\mathcal{I} \times \mathcal{I}$ array of binary relations generated from the image segments. The second layer represents the constraints generated from the model – an $\mathcal{M} \times \mathcal{M}$ array for each of the $\mathcal{V}$ viewpoints. There is a mapping (or *connection*) between each element in Layer 1 and each element in Layer 2. The third layer represents the accumulator space. It consists of an $\mathcal{M} \times \mathcal{I}$ array for each of the $\mathcal{V}$ viewpoints. For each connection between the first and second layers there is a corresponding pair of connections between the second layer and two elements in the third (given binary constraints). These connections will "fire" depending upon the consistency of model with image. The fourth layer has the same dimensions as the third and represents the co-operative process. Each element in the fourth layer is connected to the set of similar elements for neighbouring viewpatches in the third layer. The final layer again has the same dimensions as the previous and contains the results of the competitive stage. From these $\mathcal{V}$ arrays of dimension $\mathcal{M} \times \mathcal{I}$ the best globally consistent signal is extracted.

2 Algorithm

The first stage is to perform pre-computation of the model bounds array for each viewpoint on those model lines visible at that viewpoint [2]. This can be computed off-line using the chosen set of constraints. Obviously, the model constraints need only be computed once and from then on can be loaded up before run-time. Likewise it is possible to compute the image constraint array that describes the relationship between any two image lines in terms of the local constraints chosen.

2.1 Accumulation

The iterative procedure for doing this is outlined in Figure 2. As can be seen it is wholly parallel in nature – a simple outline of the procedure described in Section 1.2. Figure 3 shows a typical set of accumulator arrays after the accumulation process.

[2]It is assumed that the visibility of model lines over the viewpatches has already been pre-computed

2.2 Co-operation

The co-operative activity between neighbouring patches on the viewsphere is simulated by simply making the counts in an accumulator array for a viewpatch a function of both themselves and their corresponding counts over neighbouring viewpatches ie. a weighted average over neighbouring viewpatches. An example result is shown in Figure 4 of a typical set of accumulator arrays after this cooperative stage has been executed.

2.3 Competition

The competitive activity along the model and image dimensions of the accumulator arrays is simulated by re-adjusting counts within a raster line in favour of the winning cell. That is to say, given a particular raster line the counts are adjusted so that the winning count takes a small amount from each of its losing competitors. This adjustment is performed iteratively, adjustments alternatively along the model rasters and then along each of the image rasters. This procedure iterates until the adjustments being made are insignificantly small. The usual result is a new accumulator array with a few very large peaks, and these peaks tend to be the only peaks along any raster line (ie. along model or image raster lines). An example result is demonstrated in Figure 5 of a typical accumulator after the competitive stage. It is easy to see that it is very simple using this technique to relax the assumptions that the competition is representing; for example, allowing competition only in the image dimension removes the assumption that a model line can match only a single image segment. It is also easy to see how the assumptions can be incorporated to different degrees by the amount of *relative* competition in the two dimensions. Both the rules governing the re-distribution of the counts along rasters, and the integration of the iteration between the two dimensions can easily be adjusted to favour either of the two assumptions.

2.4 Signal Extraction: Maximal Cliques

The final stage of the algorithm consists of extracting the signal from the accumulators; that is to say, deciding which viewpatch is displaying the strongest consistent pattern of activation and what that pattern means in terms of the correspondence set. This is achieved in a sequence of three stages:

In the first stage of **Basic Extraction** the accumulator after competition is examined; all cells that are peaks in *both* their corresponding model and image dimensions are taken to represent correct correspondences for the given viewpoint ie. they are considered "on". The correspondence is obviously denoted by the indices of the accumulator cell. This gives an interpretation that meets both the assumptions that a model line matches a single image line and vice versa. However, this interpretation can still be globally inconsistent in terms of the local constraints used.

In the second stage **Maximal Cliques** are extracted. The interpretation arising from basic extraction can be considered to be a graph in which a node represents a correspondence and connectivity is defined by consistency in terms of local constraints. The maximal clique algorithm will list all combinations of correspondences that are mutually consistent. The largest maximal clique is the largest set of combinations. Applying this

algorithm is therefore enforcing global consistency within constraints in that each correspondence must be locally consistent with every other. The algorithm used for maximal clique extraction is that of Johnston [9], and as used by Bolles & Cain [8] in their *Local Feature Focus* method for locating 2D objects [3]. The algorithm is highly recursive and has been found to run very quickly on the short interpretations commonly found. The result is that the smallest number of nodes in the original interpretation are turned "off" to make the interpretation globally consistent within local constraints.

The final stage is that of **Ordering Viewpoints**. The signal strength for a viewpoint is measured as a function of the mean value and the standard deviation of the cells in the interpretation that remain "on" after Stage 2 (ie. are part of the largest maximal clique). The viewpatches are ordered in terms of the number of "on" cells (if less than 3 cells are "on" for a viewpoint it is ignored since at least 3 correspondences are necessary to invert the perspective transform); those viewpoints with the same number of "on" cells are ordered in terms of signal strength. The result is an ordered set of viewpoints and their corresponding interpretations that can be passed to a further model test stage to check for global consistency *beyond* local constraints and to extract the 6 parameters that determine object position. Due to the global consistency already enforced this set of viewpoints usually represents a small proportion of the whole viewsphere.

3 Local Constraints

In the implementation that produced the results in the next section the scale factor is unknown; the object may therefore appear at any size in the image. There are two approaches to coping with this problem in the formulation proposed: either accumulation operates over a set of likely scales as well as the set of viewpoints, or else the local constraints used are *scale independent*. The latter approach was adopted. The two geometric constraints between segments used were:

- **The Angle Constraint** states that for two image lines to match two model lines at a particular viewpatch the angle between the image lines must be within the range defined the model lines over that viewpatch (within a certain error bounds)
- **The Direction Constraint** states that for two image lines to match two model lines at a particular viewpatch the bounds on the angle defined by the arbitrary vector connecting the first image segment to the second, relative to the first segment, must be within the similar bounds defined by the model lines for that viewpatch.

These two constraints are illustrated in Figure 6. It can be seen that both these constraints are scale independent, since the constraints are defined by angles that are wholly independent of segment size. It can also be seen that both constraints are insensitive to error in the length of the image segments. The Angle constraint is wholly independent of segment length. The Direction constraint is independent as long as the error makes the segment too small rather than too large; that is to say, the range of angles defined by any subsegments of the true image segments will be a subset of the range of angles defined by the true image segments (or model segments). The result is that another dimension

[3]The actual algorithm is listed in the Appendix of [8] and described in greater detail in [7]

of scale does not need to be added to the accumulator, and the constraints used will also be robust to errors in segmentation caused by occlusion, poor edge-detection, or poor lighting.

4 Results

The results were obtained from an implementation of the above algorithm. It has been run on noisy simulated data to illustrate its potential. The implementation is actually an iterative serial one, the iteration reflecting its parallel nature, and is written in the high-level language POP11, part of the Sussex University POPLOG system.

Two models were used. The first consists of 6 3D segments connected as an irregular tetrahedron; the second consists of 10 randomly generated 3D segments that are unconnected. No account is taken in the model concerning the visibility of the segments; all lines are considered visible. In this formulation invisible lines should provide low counts in the accumulator and therefore be ignored. Visibility can be incorporated very effectively into this formulation when the model constraints are compiled; constraints are only computed for $MA_{v_{(m,m\prime)}}$ if both model lines M_m and $M_{m\prime}$ are visible at V_v. When accumulation occurs, connections involving elements of $MA_{v_{(m,m\prime)}}$ which are empty are simply ignored. This way the two rasters in CA_v corresponding to m and $m\prime$ are guaranteed to be totally empty and will be ignored in further computation.

In this implementation a very weak form of the above was incorporated. It was considered that if two model lines were projected into the image and found to intersect at a particular viewpoint then one of them was actually invisible at this viewpoint. Therefore the constraint box corresponding to these lines at this viewpoint was left empty and ignored in subsequent accumulation [4].

Noisy Images were generated from the model. The model was projected at random orientations to produce a basic image and then noise would be added. Three types of noise have been added to varying degrees:

- **Segment Reduction:** As in poor feature extraction or occlusion [5].
- **Segment Addition:** As in non-object features.
- **Segment Removal:** As in missing object features.

The two Scale Independent and Noise Robust constraints discussed in the previous section – the Angle and the Direction constraint – were used. These were both implemented as Direction Dependent and Symmetric.

Figures

The 5 figures 7 to 11 illustrate some results obtained. In the top half of the figure the accumulator arrays are displayed for the final viewpoints that provide an interpretation

[4]Intersection is easy to detect since when two segments intersect the computed bounds on the Direction Constraint will be greater than 180°

[5]Segmentation reduction was set throughout these examples at 40% ie. each segment was reduced to 60% of its true length.

and that can be verified by a model test. A maximum of 25 viewpoints are displayed; at the course resolution this corresponds to all viewpatches except one; at the fine resolution it corresponds to about $11\frac{1}{2}\%$ of the viewsphere. For display purposes the values in *each* accumulator have been normalised to take advantage of the full range of grey-values available; therefore absolute intensities are lost, but relative intensities *within* accumulators are preserved.

In the examples using the coarsely quantised viewsphere the cooperative stage was left out since it was considered that the signal would not be strongly present in neighbouring viewpatches as they are so far away. The left-hand side displays the accumulator cells for the final viewpatches in ascending order *before* competition. The right-hand side displays the same accumulators for the same viewpatches *after* competition. The horizontal represents the *model* dimension, and the vertical represents the *image* dimension. As can be seen, no peak in the competitive layer shares a raster line with another peak, so enforcing the two assumptions described above. The cells displayed in the competitive layer are not necessarily all used in the model test since some are filtered out by the maximal clique test.

The lower half of the Figure displays the results of the model test using an interpretation extracted from one of the viewpatches. Firstly the perfect image is shown – this consists of the projection of the model at the appropriate viewpoint (ignoring visibility), along with the display of the extra noise segments added to the image. Secondly, the image is degraded by reducing the length of all segments (both those that are part of the object and those that are not) – this is the actual image that is given to the matching algorithm. In the third portion the model is displayed from the viewpoint selected after matching, but having undergone a 2D rotation about this viewpoint that makes a rough approximation at the missing degree of freedom. The final projection shows the position of the model after Lowe's iterative model test [10, 11, 12] has been applied to the interpretation to match the model as accurately as possible to the corresponding line segments.

- **Figure 7:** This first figure displays results for the *ideal* situation using the model of random lines and the fine quantisation. All model lines are present but no extraneous image lines have been added. The model segments have been degraded by up to 40%.

- **Figure 8:** The small viewsphere has been used (only twenty six) viewpatches). Only five of the six model lines are present along with nine extra noisy image lines. The correct interpretation has been found in only a single viewpatch. This is not perfect since the solution should be found in at least two patches - that corresponding to the correct viewpoint, and that on the opposite side of the viewsphere corresponding to the Necker reversal [6]. The interpretation found for this viewpoint (view 17) is correct and seen to be very close to the actual viewpoint. The model test positions the model in perfect position.

- **Figure 9:** This Figure uses the small viewsphere and the random model. Only

[6]Since near parallel projection has been assumed throughout there is very minimal difference between projection at one viewpoint and that on the opposite side of the viewsphere. Since visibility constraints have been ignored the system cannot discriminate between the two possibilities.

nine of the ten model lines are present and nine extra image lines have been added. Two interpretations are found, the first of which is correct, despite the fact that the signal is masked considerably (as can be seen from the accumulators displayed). The model test is successful and the model is correctly located.

- **Figure 10:** The fine quantisation of the viewsphere has been used. Only four of the six model lines are present, and five extra image lines have been added. Three solutions are found and the model test succeeds on the first of these.

- **Figure 11:** This demonstrates this simple implementation of the algorithm being pushed to its limits. The random model has been used and the fine quantisation of the viewsphere. Five model lines have been removed and five extra image lines have been added. It can be seen from the co-operative layer that the signal/noise ratio is very weak. Seven interpretations are found and the third one is correct and passes the model test; the model test had failed on the two previous attempts using the "better" interpretations.

5 Discussion

5.1 Signal/Noise Ratio

The main factor determining the success of the algorithm described above is the signal/noise ratio; increasing its strength has a very positive effect upon the efficiency of the algorithm. This can be done only by altering the constraints checked in the accumulation stage. In the tree-search formulation the computation of the bounds on the local constraints is determined by the quantisation of the viewsphere and the degree of noise anticipated (eg. [4, 5]). In this formulation similar bounds may be computed but not all of them *need necessarily* be used. Given a correct match of two model lines to two image lines at a particular viewpoint, it is not essential that the corresponding accumulations $CA_{v_{(m,i)}}$ and $CA_{v_{(m',i')}}$ need either be considered or, if considered, that they should pass the constraint check. What is essential is the signal/noise ratio.

For instance, a local constraint bound corresponding to $MA_{v_{(m,m')}}$ may be so large that it not only lets in the signal but also a large degree of noise; in this case it might be considered that it is not worth being considered in the accumulation stage. In other words the constraint bounds may be *tightened* so that both more of the signal and more of the noise is excluded from the accumulators; it is the ratio of this exclusion that is significant. This method of exclusion could be used as a method of introducing a measure of *saliency* in that it is not accumulating upon constraints that could easily be met by chance, but only on those that are likely to be met by correct correspondences.

Of course, another method of changing the signal/noise ratio is to alter the constraints considered. The advantage of the constraints used in this implementation (Direction and Angle) is that they are Scale Independent, and so they can detect the object at any distance or scale without having to have a Scale dimension in the accumulator. If the scale/distance of the object was already known in advance then stronger constraints could be used that would substantially decrease the amount of noise in the accumulator. The signal/noise detection problem would therefore be simplified.

5.2 Advantages

The approach outlined above possesses three advantages over the original search-tree formulation of the matching problem:

1. **Parallelism:** The main advantage offered by this this approach is its inherent parallelism. The original tree-formulation is naturally serial, although such a search can be implemented to some degree in parallel. This parallelism means:
   - **Graceful Degradation:** The algorithm has robust qualities, meaning that no particular parts of the image are of especial significance. This is in distinct contrast to the tree-search formulation that has problems coping with either occlusion or degraded data.
   - **Predictable Speed:** Given a degree of serialism, it is possible to predict quite accurately how long the process will take to complete (Given *complete* serialism this may be a very long time).
   - **Potential Speed:** There is no reason why, given appropriate hardware facilities, such an algorithm could not be made to run as fast as required.
2. **Scale Independence:** The system of constraints used allows the algorithm to cope satisfactorily with the scale problem without extending the accumulation space into another dimension. This is at the expense of a weaker signal/noise ratio. If scale is already determined, the best of both worlds can be had.
3. **Reduced "Ad Hoc" Parameter Settings:** The tree-search formulation is heavily reliant on "ad hoc" parameter settings that control the search strategies and how the tree can grow. This formulation is less dependent upon such settings.

5.3 Coarse-to-Fine Strategy

It is very simple to see how this new formulation could be redesigned to employ a coarse-to-fine resolution strategy. The results above show that a resolution of 26 viewpatches is sufficient in some cases to yield a correct solution; even if it does not give a perfect interpretation it often points to the correct viewpatch. The correct viewpatch could be quantised to a finer resolution and then the accumulation repeated anew over this limited portion of the viewsphere. Extraction of the interpretation need only take place at the end. For example, given the two resolutions of $\mathcal{V}_1$ and $\mathcal{V}_2$, where $\mathcal{V}_1 < \mathcal{V}_2$, the ratio of the accumulations necessary for a coarse-to-fine strategy to the accumulations necessary for the finer resolution would be:

$$(\mathcal{V}_1 + \frac{\mathcal{V}_2}{\mathcal{V}_1}) : \mathcal{V}_2$$

In the case of the two resolutions 26 and 218 this represents savings of about 85%; this must be offset against time for dynamic configuration and potential error [7].

[7]The recursive quantisation of the viewsphere necessary for this sort of strategy comes naturally from the description of the viewsphere in terms of the icosahedron

Acknowledgements

Thanks to Dave Hogg, Vaclav Hlavac and Kelvin Yuen for their help with this work. Thanks also to Peter North for a critical insight.

References

[1] **Grimson W E L and Lozano-Perez T.** Model-based recognition and localisation from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[2] **Grimson W E L.** The combinatorics of object recognition in cluttered environments using constrained search. *Second International Conference on Computer Vision,IEEE*, pages 218–227, 1988.

[3] **Grimson W E L.** On the recognition of curved objects. *IEEE Pattern Analysis and Machine Intelligence PAMI*, 11(6):632–643, 1989.

[4] **Goad C.** Special purpose automatic programming for 3d model-based vision. *Proc. Image Understanding Workshop, Virginia, USA*, pages 94–104, 1983.

[5] **Goad C.** Fast 3d model-based vision. In Pentland A P, editor, *From Pixels to Predicates*, pages 371–391. 1984.

[6] **Bray A J.** Tracking objects using image disparities. *Proceedings of the 6th Alvey Vision Conference, Reading*, 1989.

[7] **Bolles R C.** Robust feature matching through maximal cliques. *Proc. SPIE Technical Symposium on Imaging and Assembly*, April, 1979.

[8] **Bolles R C and Cain R A.** Recognizing and locating partially visible objects, the local-feature-focus method. *International Journal of Robotics Research*, 1(3):57–82, 1982.

[9] **Johnston H C.** Cliques of a graph - large or small. *Draft, Queen's University of Belfast*, 1975.

[10] **Lowe D G.** *Perceptual Organisation and Visual Recognition.* PhD thesis, Stanford University, Dept. of Computer Science, 1984.

[11] **Lowe D G.** *Perceptual Organisation and Visual Recognition.* Boston: Kluwer, 1985.

[12] **Lowe D G.** Three-dimensional object recognition from two-dimensional images. *Artificial Intelligence*, 31(3), 1987.

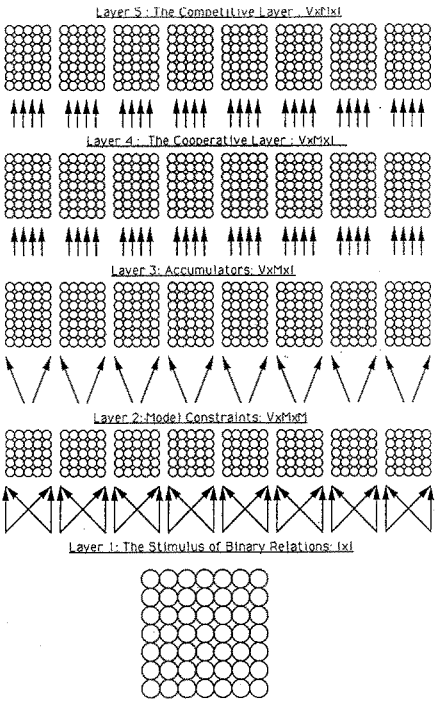

Figure 1: Diagrammatic Formulation

```
For v from 1 to V
Do  For i from 1 to I
    Do  For d from 1 to 2
        Do  For i' from 1 to I
            Do  For d' from 1 to 2
                Do  For m from 1 to M
                    Do  For m' from 1 to M
                        Do  If      consistent(IA(i,d,,i',d'),MA(v)(m,m'))
                            Then    CA(v)(m,i)   = CA(v)(m,i)   + 1 ;
                                    CA(v)(m',i') = CA(v)(m',i') + 1 ;
                            Endif
                        Endfor
                    Endfor
                Endfor
            Endfor
        Endfor
    Endfor
Endfor
```

Figure 2: Iterative Procedure

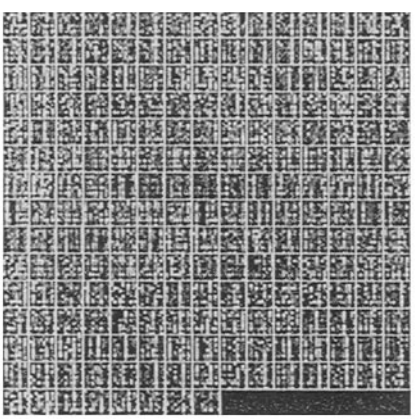

Figure 3: Typical Accumulator

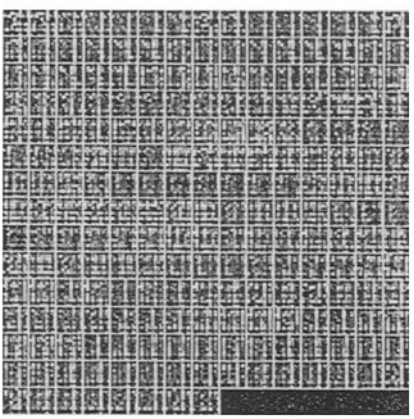

Figure 4: Accumulator after Cooperation

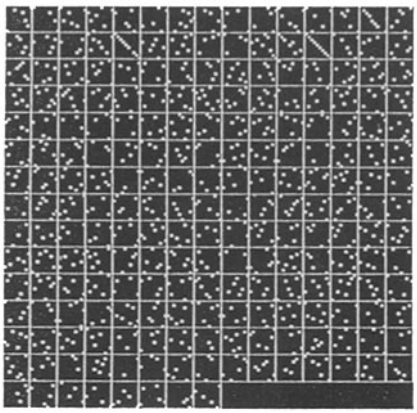

Figure 5: Accumulator after Competition

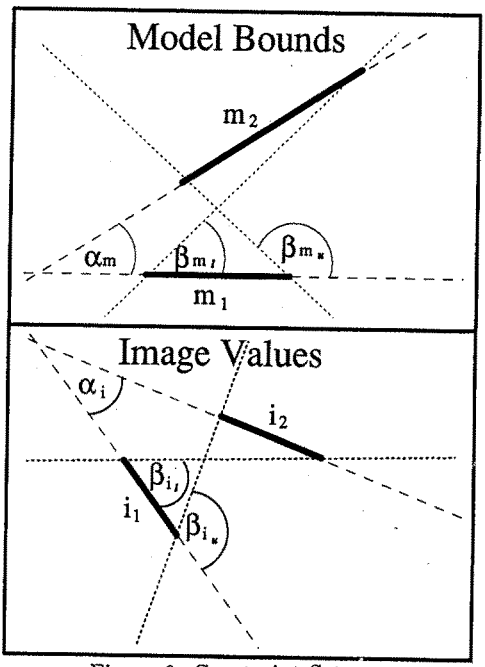

Figure 6: Constraint Set

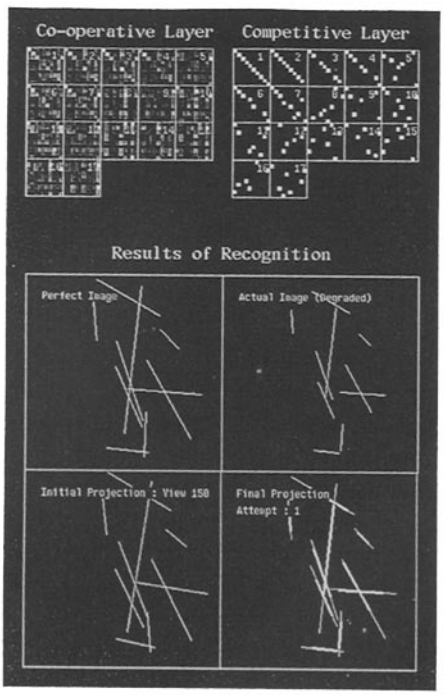

Figure 7: Perfect Example

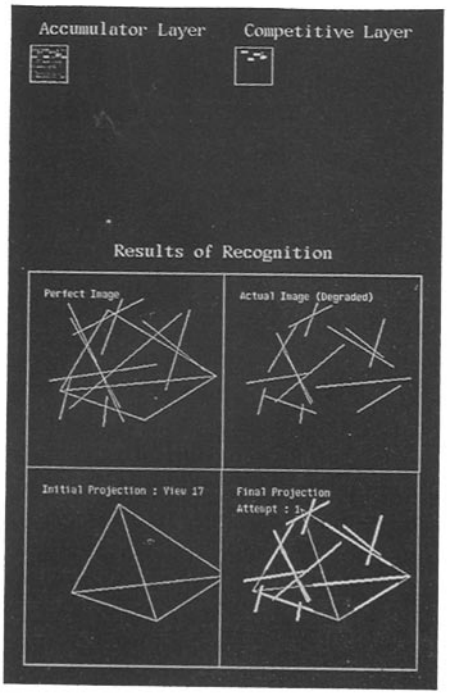

Figure 8: Small Viewsphere 1

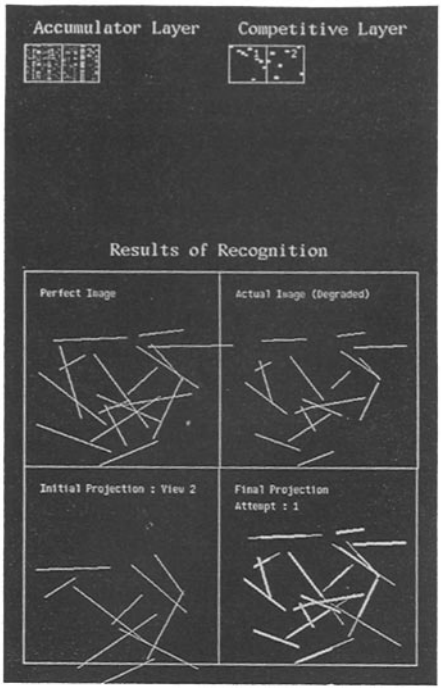

Figure 9: Small Viewsphere 2

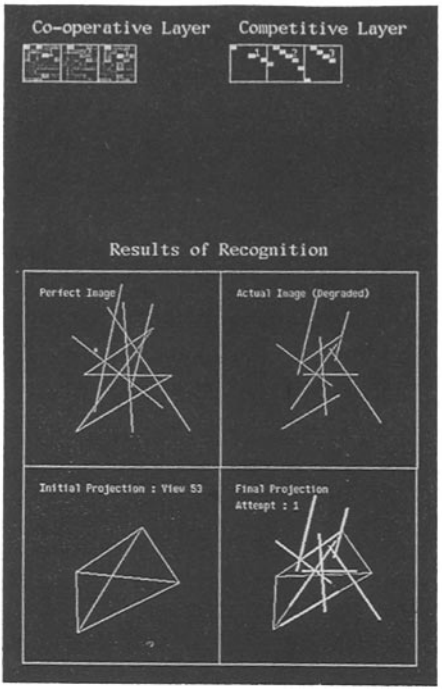

Figure 10: Large Viewsphere 1

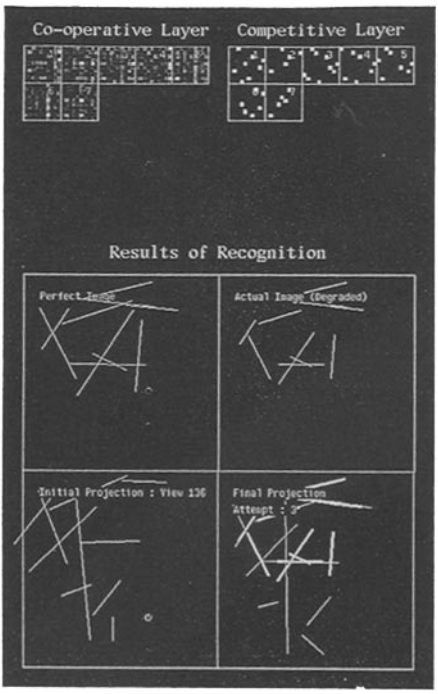

Figure 11: Large Viewsphere 2