PARAMETRIC CORRESPONDENCE AND CHAMFER MATCHING: TWO NEW TECHNIQUES FOR IMAGE MATCHING

H.G. Barrow J.M. Tenenbaum, R.C. Bolles, H.C. Wolf SRI International. Menlo Park. California 94025

Abstract

Parametric correspondence is a technique for matching images to a three dimensional symbolic reference map. An analytic camera model is used to predict the location and appearance of landmarks in the image, generating a projection for an assumed viewpoint. Correspondence is achieved by adjusting the parameters of the camera model until the appearances of the landmarks optimally match a symbolic description extracted from the image

The matching of image and map features is performed rapidly by a new technique, called "chamfer matching", that compares the shapes of two collections of shape fragments, at a cost proportional to linear dimension, rather than area These two techniques permit the matching of spatially extensive features on the basis of shape, which reduces the risk of ambiguous matches and the dependence on viewing conditions inherent in conventional image based correlation matching.

Introduction

Many tasks involving pictures require the ability to put a sensed image into correspondence with a reference image or map. Examples include vehicle guidance, photo interpretation (change detection and monitoring) and cartography (map updating). The conventional approach is to determine a large number of points of correspondence by correlating small patches of the reference image with the sensed image. A polynomial interpolation is then used to estimate correspondence for arbitrary intermediate points [Bernstein]. This approach is computationally expensive and limited to cases where the reference and sensed images were obtained under similar viewing conditions. In particular, it cannot match images obtained from radically different viewpoints, sensors, or seasonal or climatic conditions, and it cannot match images against symbolic maps.

Parametric correspondence matches images to a symbolic reference map, rather than a reference image. The map contains a compact three dimensional representation of the shape of major landmarks, such as coastlines, buildings and roads. An analytic camera model is used to predict the location and appearance of landmarks in the image, generating a projection for an assumed viewpoint. Correspondence is achieved by adjusting the parameters of the camera model (i.e. the assumed viewpoint) until the appearances of the landmarks optimally match a symbolic description extracted from the image

The success of this approach requires the ability to rapidly match predicted and sensed appearances after each projection The matching of image and map features is performed by a new

technique, called "chamfer matching", that compares the shapes of two collections of curve fragments at a cost proportional to linear dimension, rather than area

In principle this approach should be superior, since it exploits more knowledge of the invariant three dimensional structure of the world and of the imaging process. At a practical level, this permits matching of spatially extensive features on the basis of shape, which reduces the risk of ambiguous matches and dependence on viewing conditions.

Chamfer Matching

Point landmarks such as intersections or promontories are represented in the map with their associated three dimensional world coordinates. Linear landmarks, such as roads or coastlines are represented as curve fragments with associated ordered lists of world coordinates. Volumetric structures such as buildings or bridges, are represented as wire frame models.

From a knowledge of the expected viewpoint, a prediction of the image can be made by projecting world coordinates into corresponding image coordinates, suppressing hidden lines. The problem in matching is to determine how well the predicted features correspond with image features, such as edges and lines.

The first step is to extract image features by applying edge and line operators or tracing boundaries. Edge fragment linking [Nevatia, Perkins] or relaxation enhancement [Zucker, Barrow] is optional. The net result is a feature array each element of which records whether or not a line fragment passes through it. This process preserves shape information and discards greyscale information, which is less invariant.

To correlate the extracted feature array directly with the predicted feature array would encounter several problems: The correlation peak for two arrays depicting identical linear features is very sharp and therefore intolerant of slight misalignment or distortion (e.g., two lines, slightly rotated with respect to each other, can have at most one point of correspondence) [Andrus]; A sharply peaked correlation surface is an inappropriate optimization criterion because it provides little indication of closeness to the true match, nor of the proper direction in which to proceed; Computational cost is heavy with large feature arrays,

A more robust measure of similarity between the two sets of feature points is the sum of the distances between each predicted feature point and the nearest image point. This can be computed efficiently by transforming the image feature array into an array of numbers representing distance to the nearest image feature point. The similarity

measure is then easily computed by stepping through the list of predicted features and simply summing the distance array values at the predicted locations

The distance values can be determined in two passes through the image feature array by a process known as "chamfering" [Munson, Rosenfeld]. The feature array ($F[i\ j]\ i.j=1\ N$) is initially two-valued: 0 for feature points and infinity otherwise The forward pass modifies the feature array as follows

```
FOR i \leftarrow 2 STEP 1 UNTIL N DO

FOR j \leftarrow 2 STEP 1 UNTIL N DO

F[i.j] \leftarrow MINIMUM(F[i,j]) (F[i \ 1 \ j]+2).

(F[i-1 \ j-1]+3) (F[i,j-1]+2),

(F[i+1.J-1]+3));
```

Similarly, the backward pass operates as follows:

```
FOR i <- (N-1) STEP -1 UNTIL 1 DO

FOR j <- (N-1) STEP -1 UNTIL 1 DO

F[i,j] <- MINIMUM(F[i,j], (F[i+1,j]+2), (F[i+1,j+1]+3), (F[i,j+1]+2), (F[i-1,j+1]+3));
```

The incremental distance values of 2 and 3 provide relative distances that approximate the Euclidean distances 1 and the square root of 2

Chamfer matching provides an efficient way of computing the integral distance (i.e. area) or integral squared distance, between two curve fragments, two commonly used measures of shape similarity. Note that the distance array is computed only once, after image feature extraction.

Parametric, Correspondence

Parametric correspondence puts an image into correspondence with a three dimensional reference map by determining the parameters of an analytic camera model (3 position and 3 orientation parameters).

The traditional method of calibrating the camera model takes place in two stages: first, a number of known landmarks are independently located in the image, and second, the camera parameters are computed from the pairs of corresponding world and image locations, by solving an over-constrained set of equations [Sobel, Quara, Hannah].

The failings of the traditional method stem from the first stage. The landmarks are found individually, using only very local context (e.g. a small patch of surrounding image) and with no mutual constraints. Thus local false matches commonly occur. The restriction to small features is mandated by the high cost of area correlation, and by the fact that large image features correlate poorly over small changes in viewpoint.

Parametric correspondence overcomes these failings by integrating the landmark-matching and camera calibration stages. It operates by hill-climbing on the camera parameters. A transformation matrix is constructed for each set of parameters considered, and it is used to project landmark descriptions from the map onto the image at a particular translation, rotation, scale and

perspective. A similarity score is computed with chamfer matching and used to update parameter values. Initial parameter values are estimated from navigational data.

Integrating the two stages allows the simultaneous matching of all landmarks in their correct spatial relationships. Viewpoint problems with extended features are avoided because features are precisely projected by the camera model prior to matching. Parametric correspondence has the same advantages as rubber-sheet template matching [Fischler, Widrow] in that it obtains the best embedding of a map in an image, but avoids the combinatorics of trying arbitrary distortions by only considering those corresponding to some possible viewpoint.

An Example

The following example illustrates the major concepts in chamfer matching and parametric correspondence. A sensed image (Figure 1) was input along with manually derived initial estimates of the camera parameters. A reference map of the coastline was obtained, using a digitizing tablet to encode coordinates of a set of 51 sample points on a USGS map Elevations for the points were entered manually. Figure 2 is an orthographic projection of this three dimensional map.

A simple edge follower traced the high contrast boundary of the harbor, producing the edge picture shown in Figure 3. The chamfering algorithm was applied to this edge array to obtain a distance array. Figure U depicts this distance array; distance is encoded by brightness with maximum brightness corresponding to zero distance from an edge point.

Using the initial camera parameter estimates, the map was projected onto the sensed image (Figure 5). The average distance between projected points and the nearest edge point, as determined by chamfer matching, was 25.8 pixels.

A straightforward optimization algorithm adjusted the camera parameters, one at a time, to minimize the average distance. Figures 6 and 7 show an intermediate state and the final state, in which the average distance has been reduced to 0.8 pixels. This result, obtained with 51 sample points, compares favorably with a 1.1 pixel average distance for 19 sample points obtained using conventional image chip correlation followed by camera calibration. The curves in Figure 8 characterize the local behavior of this minimum, showing how average distance varies with variation of each parameter from its optimal value. Approximately 60 iterations (each involving a parameter adjustment and reprojection). were required for this example. The number of iterations could be reduced by using a better optimization algorithm, for example, a gradient search

Dscussion.

We have presented a scheme for establishing correspondence between an image and a reference map that integrates the processes of landmark matching and camera calibration. The potential advantages

of this approach stem from 1) matching shape, rather than brightness, 2) matching spatially extensive features, rather than small patches of image, 3) matching simultaneously to all features, rather than searching the combinatorial space of alternative local matches, 4) using a compact three dimensional model; rather than many two dimensional templates.

Snape has proved to be much easier to model and predict than brightness. Shape is a relatively invariant geometric property whose appearance from arbitrary viewpoints can be precisely predicted by the camera model. This eliminates the need for multiple descriptions, corresponding to different viewing conditions, and overcomes difficulties of matching large features over small changes of viewpoint.

The ability to treat the entirety of the relevant portion of the reference map as a single extensive feature reduces significantly the risk of ambiguous matches, and avoids the combinatorial complexity of finding the optimal embedding of multiple local features.

A number of obstacles have been encountered ir reducing the above ideas to practice. The distance metric used in chamfer matching provides a smooth, monotonia measure near the correct correspondence, and nicely interpolates over gaps in curves. However, scores can be unreliable when image and reference are badly out of alignment. In particular, discrimination is poor in textured areas, aliasing can occur with parallel linear features, a single isolated image feature can support multiple reference features.

The main problem is that edge position is not a distinguishing feature and consequently many alternative matches receive equal weight. of overcoming this problem, therefore, is to use more descriptive features brightness discontinuities can be classified, for example, by orientation, by edge or line, and by local spatial context (texture versus isolated boundary). type of feature would be separately chamfered and map features would be matched in the appropriate Similarly, features at a much higher level could be used, such as promontory or bay, area features having particular internal textures or structures, and even specific landmarks, such as "the top of the Transamerica pyramid". Ideally. with a few highly differentiated features distributed widely over the image the earanietric correspondence process would be able to home in directly on the solution regardless of initial conditions.

Another dimension for possible improvement is the chamfering process itself. Determining for each point of the array a weighted sum of distances to many features (e.g. a convolution with the feature array), instead of the distance to the nearest feature, would provide more immunity from isolated noise points. Alternatively, propagating the coordinates of the nearest point instead of merely the distance to it, it becomes possible to use characteristics of features, such as local slope or curvature, in evaluating the goodness of match. It also makes possible a more directed

search, since corresponding pairs of points are now known, an improved set of parameter estimates can be analytically determined.

Chamfer matching and parametric correspondence are separable techniques. Conceptually, parametric correspondence can be performed by re projecting image chips and evaluating the match with correlation. However, the cost of projection and matching grows with the square of the template size: The cost for chamfer matching grows linearly with the number of feature points. Chamfer matching is an alternative to other shape matching techniques, such as chain-oode correlation [Freeman], Fourier matching [Zahn], and graph matching [e.g. Davis]. Also, the smoothing obtained by transforming two edge arrays to distance arrays via chamfering can be used to improve the robustness of conventional area-based edge correlation.

Parametric correspondence, in its most general form, is a technique for matching two parametrically related representations of the same geometric structure. The representations can be two- or three-dimensional, iconic or symbolic; the parametric relation can be perspective projection, a simple similarity transformation, a polynomial warp, and so forth. This view is similar to rubber-sheet template matching as conceived by Fischler and Widrow [Fischler, Widrow]. The feasibility of the approach in any application, as Widrow points out, depends on efficient algorithms for "pattern stretching, hypothesis testing, and pattern memory", corresponding to our camera model, chamfer matching, and three dimensional map.

As an illustration of its versatility, the technique can be used with a known camera looation to find a known object whose position and orientation are known only approximately. In this case, the object's position and orientation are the parameters; the object is translated and rotated until its projection best matches the image data Such an application has a more iconic flavor, as advocated by Shepard [Shepard], and is more integrated than the traditional feature extraction and graph matching approach [Roberts, Falk and Grape].

As a final consideration, the approach is amenable to efficient hardware implementation. There already exists commercially available hardware for generating parametrically specified perspective views of wire frame models at video rates, complete with hidden line suppression. The chamfering process itself requires only two passes through an array by a local operator, and match scoring requires only summing table lookups in the resulting distance array.

Conclusion

Iconic matching techniques, such as correlation, are known for efficiency and precision obtained by exploiting all available pictorial information, especially geometry. However, they are overly sensitive to changes in viewing conditions and cannot make use of non-pictorial information. Symbolic matching techniques, on the other hand, are more robust because they rely on

invariant abstractions, but are less precise and less efficient in handling geometrical relationships. Their applicability in real scenes is limited by the difficulty of reliably extracting the invariant description. The techniques we have put forward offer a way of combining the best features of iconic and symbolic approaches.

Acknowletamenfrs

This work was supported by ARPA under contract DAAG29-76-C-0012. Additional support was provided by NASA under contract NASW-2865.

REFERENCES

- Andrus J.F.. Campbell, C.W. and Jayroe R.R., "A Digital Image Registration Method Using Boundary Maps". IEEE Trans. Corap., Sept. 1975, p. 935-939.
- Barrow, H.G., "Interactive Aids for Cartography and Photo Interpretation", Interim Report of ARPA Project DAAG29-76-C-0057, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California. Dec. 1976.
- Bernstein, R.. "Digital Image Processing of Earth Observation Sensor Data", IBM Journal of Research and Development, Vol. 20, No. 1, 1976.
- Davis, L., "Shape Matching Using Relaxation Techniques", TR-480. Computer Science Dept., University of Maryland, Sept. 1976.
- Falk, G., "Interpretation of Imperfect Line Data as a Three Dimensional Soene," Artificial Intelligence, Vol. 3, 1972, p. 101-144.
- Feder, J. and Freeman, H., "Segment Fitting of Curves in Pattern Analysis Using Chain Correlation", AD619525, March 1965.
- Fischler, M. and Elschlager, R. "The Representation and Matching of Pictorial Structures", IEEE Trans. Comp., no. 22, 1973, p.67-92.
- Grape, G.R. "Model Based (Intermediate Level) Computer Vision", Stanford Al Memo AlM-201, May 1973, Ph.D. Thesis.
- Hannah, M.J., "Computer Matching of Areas in Stereo Images", Stanford Al Memo AlM-239, July 1974, Ph.D. Thesis.
- Munson, J., Internal memo, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, Dec. 1973.
- Perkins, W.P., "Multi-level Vision Recognition System", Proc. Third IJCPR Nov. 1976, p.739-
- Quam, L.H., "Computer Comparison of Pictures", Stanford Al Project Memo AlM 144, May 1971, Ph.D. Thesis.
- Roberts, L.G., "Machine Perception of Threedimensional Objects", in: Tippet, J.T., et al. (Eds.), "Optical and Electro-Optical Information Processing", MIT Press, 1965, p.159-197.

- Rosenfeld, A. and Pflatz, J.L., "Distance Functions on Digital Pictures", Pattern Recognition, Vol. 1 No. 1, July 1968, p.33 62.
- Shepard, R.N. and Metzler, J., "Mental Rotation of Three-Dimensional Objects", Science, 1971, p.701-703.
- Sobel, I., "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes", Artificial Intelligence, Vol. 5, 1974, p.185-198.
- Widrow, B., "The 'Rubber-Mask' Technique", Pattern Recognition, Vol. 5, 1973, p.175-211.
- Zahn, C. and Roskies, R., "Fourier Descriptors for Plane Closed Curves", IEEE Trans. Comp., no. 21. 1972, p.269-281.
- Zucker, S., Hummel, R. and Rosenfeld, A., "An Application of Relaxation Labeling to Curve and Line Enhancement", IEEE Trans. Comp., no. 25, 1976.

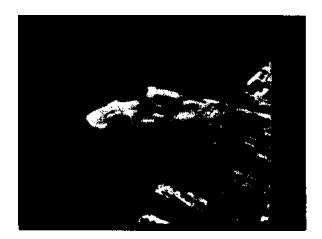


Figure 1. An aerial image of a section of coastline.

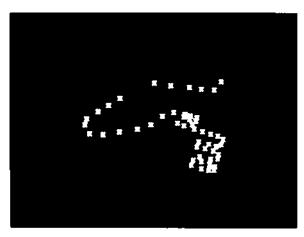


Figure 2. A set of sample points taken from a USGS map.

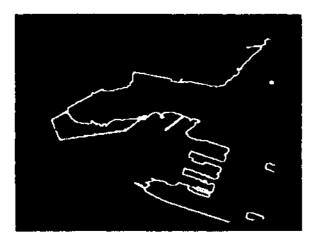


Figure 3. The traced boundary of the coastline.

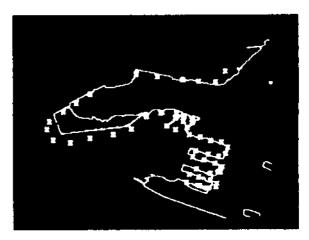


Figure 6. Projection of map points onto the image after some adjustment of camera parameters.

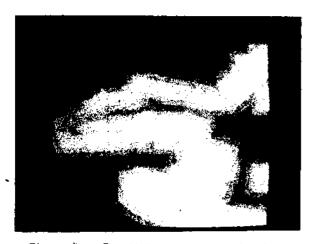


Figure 4. The distance array produced by chamfering the boundary.

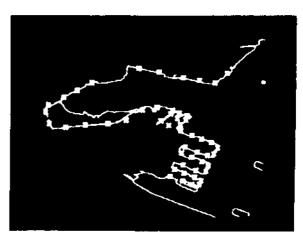


Figure 7. Projection of map points onto the image after optimization of camera parameters.

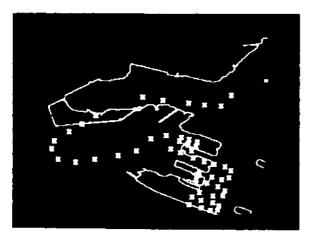


Figure 5. Initial projection of map points onto the image.

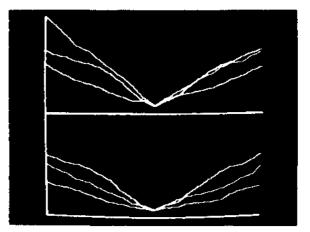


Figure 8. Behavior of average distance score with variation of the six camera parameters from their optimal values.

a query has been satisfied, the template node is represented by instantiated nodes with attached specialized location descriptions as shown in Figure 2.1. Four basic types of links provide a simple syntax to the network structure. A powerful advantage to this syntax is that the executive procedure can direct the analysis in a more general way.

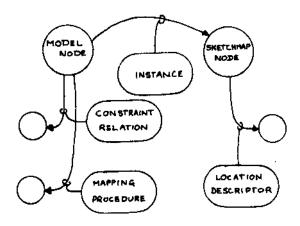


Figure 2.1 The Next Level of Detail in Model-Sketchmap Nodes

2.2 Constraints

Links to other model nodes encode (perhaps parametrized) <u>constraint relati</u>ons between model nodes. Links can encode:

- the probability that the relationship holds;
- a quantifier representing the expected value of the relationship;
- conditions under which the relationship holds.

For example, the relationship SHIP ADJACENT DOOK might have a certain probability of being true, an expected distance that the ship is from the dock, and conditions for the relation being true. We refer to the template nodes and geometric relations between them as the constraint network. This network may be interpreted like a program to find subsets of the model or the image that satisfy the constraints. Its results take account of partial or unspecified information, and it may be updated upon receipt of better data with a minimal amount of work. It is much like the graph of variable dependencies in AL [Feldman et al. 1975]. In brief, each node has a "Constraint Operation," such as Intersection, Translation, Union, or indeed any function of up to two arguments; it has two operand nodes; a father node; a status that may be "Up-To-Date" or "Out-Of-Date"; and a value that is some data structure such as a number, a list of linear objects, a region, etc.

The constraint network for the prose: "The centroids of docked ships are on lines parallel to the intersection of coastlines with dock areas at a distance of one-half a ship width" is shown

in Figure 2.2.

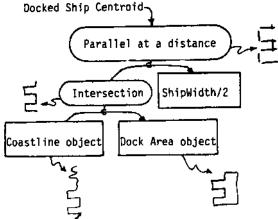


Figure 2.2 A Portion of a Constraint Network

The network starts out with data (from the model or from previous scene analysis) as the values of the tip nodes, but no values at nonterminal nodes and all nonterminals marked Out-Of-Date. Data at a tip node can have one of three statuses: it can be known that the object does not exist in the scene (so the value of the node is the null set), it can be known to some degree of accuracy where objects are in the scene (so the value of the node is a subset of image or world points), or perhaps nothing is known (in which case the object could be anywhere, and the value is implicitly the universe of image or world points).

When the constraint network is "run" to determine what is known about the location of its object, each node recursively evaluates its Out-Of-Date operand nodes, performs its operation, and stores the result in its value. It marks its status Up-To-Date. Intersection and Union work properly with the definitions of partial information of the last paragraph. When new (or better) information about an object at a tip of the graph comes in, all nodes on a path from the tip to the root are marked Out-Of-Date. Then when the graph is next run, (only) the necessary partial results are re-computed. In keeping with our philosophy, the graph is not self-activating, but is run on explicit user command.

2.3 Location Descriptors

A <u>location descriptor</u> provides information about where to find an entity. The part of the location descriptor which specifies a point set enclosing the region has been referred to as a tolerance region [Bolles 1975]. A shape location descriptor might have the structure shown in Figure 2.3.

This organization is suggestive of a framelike structure. However, not all the entries need exist; just the syntax is necessary to allow the entries to be found. In practice only the properties relevant to a particular query will be generated. Such partial instantiations are easy in our implementation language LEAP [Feldman & Rovner 1969].

[ShapeLocationDescriptor nodetype: specialization prototype instance-of: a LocationDescriptor locates: OneOf {(a ShapeObject), (a ShapeFeature)} coordsystem: a CoordinateSystem a PointSet centroid: //allows for "fuzziness" orientation: an AngleRange //...ditto tol. region: a PointSet]

...similarly for Point, Linear, and AreaLocati onDescri ptors

[CoordinateSystem

nodetype: abstract prototype units: a LengthlinitSpecification a NumberRange scale: //length units / system unit transforms: SetOf {((a Coordinate Transform) (a Coordinate System)), ...)]

Figure 2.3 Example of a Shape Location Descriptor

There are many advantages to having a standard representation for object locations:

- a. If such descriptions are data types, their computations can be separated from the procedures that use them. If they can be passed as arguments, they provide a certain "common currency" between procedures, thus simplifying and modularizing the procedures that use them.
- b. Location descriptors can represent approximate locations, which is useful for queries unconcerned with exact answers.
- c. Constraints between locations can propagate knowledge throughout the model. Location descriptors can be computed from other location descriptors via relations, or by union and intersection of the described point sets. A system which applied linear programming techniques to the problem of locating regions through constraints placed on their boundaries was developed in [Taylor 1976].
- d. Use of location descriptors is geared to an abandonment of the exhaustive segmentation paradigm wherein every region must correspond to some object. Different Iocation descriptors may refer to disjoint point sets or may overlap on the image, and different objects may have similar location descriptors.

3. Control

3.1 General Philosophy

Generally a query results in the synthesis of a sketchmap with instance nodes whose location descriptors are accurate enough for the purposes of the query. A query might also result in further refinement of location descriptors of the extension of an existing sketchmap to account for more image structure. A query-directed vision system should thus be able to use relevant information generated in successive queries. Most queries will take the form of user-written programs, since nontrivial tasks usually require fairly rigid recommendations about how the system should go about solving them. Initially the system will not attack the problem of automatically translating queries in some command language into programs. At the highest level control is embedded in the form of a user-written executive program. Figure 3.1 shows a fragment of the code used in the current executive procedure for selecting mapping procedures which identify instances of rib nodes in chest radiograph images. Each mapping procedure has pre-conditions, including an associated accuracy measure, which can depend on its neighbors, as well as a cost measure. The cheapest rib procedure which satisfies the pre-conditions is selected. The important point here is that the executive can have a relatively simple structure. This facilitates the experimentation of various control strategies other than the depth-first strategy shown in the example.

```
Recursive Procedure MatchRib(Itemvar Node);
   begin
    itemvar z, vilntager Vartif INSTANCE of Node is ANY
     then
         begin
           Print("rib ". Node. " already matched");
           returni
     e lue
     I run procedure that will do the Job with min cont;
          itempor TempProc: integer MinCost, TempCost;
MinCost: < VeryLarge;
foreach x such that
RIBIPROCEDURE of Node is x do
              begin
                 yar := GetConstraintsAndVarlance(Node.x);
if Var < Tolerance then
                    begin
                        TempCost := FINDCost(Node, x);
if TempCost < MinCost then
                          begin
                              TempCost := x;
MinCowt := TempCost;
                          end;
                    end;
              end:
         if MinCost = VeryLarge
            the n
              Print("No procedure can do job for ", Node)
           else ApplyProc(TempProc, Rode);
         foreach v such that MEICHBOR of Mode is v
                                and TYPE of w is RIB do MatchRib(w);
   end:
Figure 3.1 Executive Procedure for Ribs
```

3.2 Characterizations of Mapping Procedures

Mapping procedures have associated <u>descriptions</u> which are used by executive procedures, The descriptions contain the following:

- the slots in the data object which must be filled for the procedure to run;
- the slots the procedure can fill in;
- the cost and accuracy of the procedure in some meaningful units;

- the a priori reliability of the procedure. Examples of the kinds of facts we expect to be able to encode in this structure are (for a straight-line structure) that a Hough transform cannot find the endpoints of a line but is more reliable than the cheaper Shirai tracker, which itself needs to know the direction of a line before it can track it, and that a Heuckel operator is more expensive, but can furnish many facts about the line with little known a priori, and can rate itself on reliability of its result.

There are several advantages to separating the executive procedure from the mapping procedures and their descriptions:

- a. The executive procedure can be written without considering the implementation details of mapping procedures in great depth.
- b. Mapping procedures are similarly simplified without the burden of determining an appropriate context for their application.
- c. The executive procedure can automatically select alternative procedures in the event of mapping procedure failures.
- d. Descriptions allow a choice between methods (if several are available) based on capability, resource requirements, and a priori reliability. (Also, recovery from failure of individual routines can be automated through planning [Feldman and Sproull 1975].)
- e. If the mapping procedures can produce reliable a priori estimates of their success, the analytical results of [Bolles 1975] and [Taylor 1976] could be extended to select the procedure which produces sufficiently exact data objects.

4. Applications

4.1 Finding Docked Ships

Finding ships in a dock scene illustrates how high-level metrical knowledge about the image (such as provided by a topographic map) can make certain scene analysis problems easy.

The model contains in a Constraint Graph form (see Section 2.2) the knowledge that docked ships are in the ocean adjacent to dock areas, parallel to the dock and with centroid a distance away related to the width of the ship. In a Shape Object Descriptor, some facts about the sorts of ships we are trying to find are stored,

viz. a template for matching them (in our case, a rectangle of I's in an array for template-matching), their width, length, average brightness, etc. Template-matching is among the simplest vision primitives. Only in a context having a great deal of structure could it be expected to work in scenes as complex as Figure 4.1.

The system, under direction of the userwritten query, begins by deciding where to look by satisfying a constraint network; the more information provided, the narrower the focus of attention. In the case illustrated in this section, the constraint network looked as it does in Section 2.2. Figure 4.1 shows a halftoned version of the image, with the dock area and coastline overlaid as black lines. Recent work at SRI [Barrow and Tenenbaum 1977] has shown that map data may be automatically registered with images such as ours to within better than a pixel, so we felt comfortable about bypassing the registration problem in this study. Were the registration uncertain, the constraints would produce a more fuzzy area to search than they did. Shown in the ocean are black lines indicating the areas of search which arise from the constraint tree; their linear nature makes a simple template-matching technique a possibility (in this exercise it was the only technique, but an executive procedure might well have chosen it as applicable). The ship template is rotated to be parallel to the midline as given by the constraint graph, and templatematching is done along the line; note is taken of where the score for the match goes over threshold, and when it comes back down under threshold. The average of these two positions is taken as the location of a ship. The black crosses show the results.

Our image data is a USGS mapping photograph provided by SRI, digitized to 256 grey levels on a .007" grid at USC. The image is stored on disk in RV format, and can be sampled at integral size reductions into an integer array in core for processing (see [Maleson & Rashid 1977] re. RV format).

Linear objects are SAIL records making linked lists of (x,y) points. They can have four types at present: a list of points to be connected in order; a list of segments, or pairs of endpoints to be connected pairwise; and logically circular lists of points representing boundaries. A robust and general routine based on merging was written to compute the intersection of such linear features. Other useful geometric routines find the distance of a point from a segment (not a line), and compute a segment parallel to and some distance from another segment.

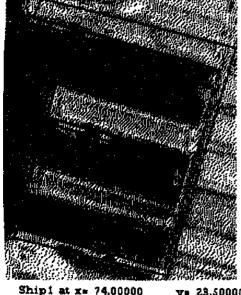
Regions (except for templates, which are arrays) are SAIL list items. A region is a list of y-lists; a y-list has a y-value followed by an even number of x-values. The first x-value is an "entering region" boundary point, the second is a "leaving region" boundary point, and so on alternately. The region: 001

101

011

would be represented as ((1 2 3)(2 1 1 3 3)(3 3 3)). Routines were written, again based on merging, to create the union and intersection of such regions, and to convert (via an asymmetric DDA algorithm [Newman and Sproull 1973]) linear objects to regions. We find multiple representations of objects simplifies the work of routines such as the constraint primitives.

Template-matching utilities can produce an array containing a rotated and scaled version of a template and can compute the correlation of a template (at some rotation and translation) with the image array.



Ship1 at x= 74,00000 y= 29,50000 Ship2 at x= 89,50000 y= 39,00000 Ship3 at x= 59,50000 y= 102,5000 Ship4 at x= 49,00000 y= 171,5000 End of SAIL execution

Figure 4.1 Harbor Scene with Dock Areas, Lines of Search, and Detected Ships

4.2 Finding Ribs in Chest Radiographs

The problem of finding ribs in chest radiographs illustrates the use of multiple procedures attached to the same template node and the use of less precise geometric constraints arising from anatomy rather than cartography.

The model contains nine right and left ribs (the maximum amount normally visible on a chest film). Presently only the lower edge of each rib is detected. Each rib is modelled as a template node with offset parameters from itself to each immediate neighbor (above, below, opposite). Additionally, three kinds of procedures are attached to each rib node as shown in Table 4.1.

LookForARib uses the Weschler parabolic model [Weschler & Sklansky 1975] to find a rib segment.

AffirmARib translates that segment using the offset parameters and attempts to verify the presence of a rib by a correlation technique. HallucinateARib instantiates a rib by translating a neighbor with no verification.

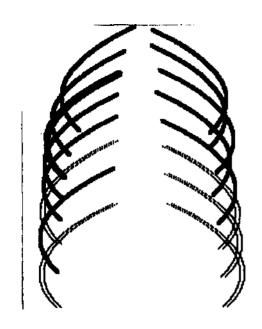
Table 4.1 RibFinding Procedures

Procedure	Precon- ditions	Cost	Var.	Postcon- ditions
LookForARib	none	20	0	instance of rib
AffirmARib	instance of neigh- bor in sketchmap	4	1	instance of rib
HallucinateARib	instance of neigh- bor in sketchmap	1	5	instance of rib

Figure 4.2 shows the result of finding ribs on a particular chest film. The ribs are labelled with special symbols to denote the procedures which located them.



Figure 4.2a Local Edges Detected by Hueckel Edge Operator



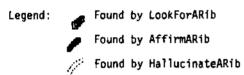


Figure 4.2b Results of Rib Finding

5. Summary

The semantic network is a kind of lumped parameter model in the spirit of [Fischler and Eschlager 1973]. The geometric constraints in the network relate template nodes whose descriptions (the "lumped parameters") are generated by attached mapping procedures. The key difference is that information found during the analysis can change the way template nodes are located.

In analyzing an image it is crucial that the generating of abstract descriptions of parts of the image, segmentation, be intimately connected with the interpretation of those parts. In our system the former operation corresponds to generating sketchmap-image links whereas the latter corresponds to generating model-sketchmap links. Interpretation and segmentation are united through multiple mapping procedures and the executive, which can efficiently change the way a part of the image is analyzed as new information about the rest of the image develops.

Finally, we want the image analysis process to do as little work as possible to satisfy a given task or query. This is attempted through the specialization of all parameters to the given task, the inclusion of performance and accuracy measures in the mapping procedure descriptions, and the use of the constraint network. All of this is just the beginning of a long term effort to

study what can be done in a general way for goal directed image understanding tasks.

References

- Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., and Wolf, H.C. "Parametric Correspondence and Chamfer Matching; Two New Techniques for Image Matching," DARPA Conference, Minneapolis, MN, April 1977.
- Bolles, R. "Verification Vision Within a Programmable Assembly System," Stanford Al Memo, AIM-275, December 1975.
- Feldman, J.A., Finkel, R., Taylor, R., Bolles, R., and Paul, R. "An Overview of AL, A Programming System for Automation," Proc. 4th IJCAI, Tbilisi, Russia, 1975.
- Feldman, J.A. and Rovner, P.D. "An Algol-Based Associative Language," CACM, Vol. 12, No. 8, pp. 439-449, August 1969.
- Feldman, J.A. and Sproull, R.F. "Decision Theory and Artificial Intelligence II: The Hungry Monkey," TR2, University of Rochester Computer Science Department, November 1975.
- Fischler, M.A. and Eschlager, R.A. "The Representation and Matching of Pictoral Patterns," IEEE Trans, on Computers, C-22, January 1973.
- Maleson, J. and Rashid, R. "The Rochester Image Protocol," University of Rochester Computer Science Department, Internal Memo, February 1977.
- Marr, D. "Analyzing Natural Images," M.I.T. Al Lab, AIM-334, June 1975.
- Newman, W.M. and Sproull, R.F. <u>Principles of Interactive Computer Graphics</u>," McGraw-Hill, Inc., 1573:
- Taylor, R.H. "Generating Al Programs from High-Level Task Descriptions," Ph.D. Thesis, Stanford Al Lab, 1976.
- Weschler, H. and Sklansky, J. "Automatic Detection of Rib Contours in Chest Radiographs," <u>Proc. 4th IJCAI</u>, Tbilisi, Russia, 1975.