**Methods for graphical feature extraction:**

*One of the techniques described by Muchnik, describes a method for “forming the simplest features” and hypothesizes that they may be used as a basis for constructing more intricate features. He defines the term “informative fragments” as portions of an image which can be described in terms of simple words such as comers, curvature, and intersection. A major assumption is that the fragments are spaced well apart. Using a gradient search technique, a local extremum of the distance function is found and thus isolates an “informative fragment.” Hence the data will yield many reference attributes which may then be organized using a clustering program.*

*Similarly, a predictive coding experiment, concerned mainly with the transmission of pictorial data, presented by Wholey. A statistical survey of a representative class of weather maps, traced onto a 70 x 100 matrix (7000 elements), was made to determine the most likely grey level (black or white) to follow an aperture with a particular grey level distribution. Given the statistics and an image to be coded, an error matrix can be generated which can later be used to regenerate the original picture. The method did not prove to be very successful.*

*Kamentsky and Liu discuss a design and search procedure for multi-font print recognition logic, which is also based on an analysis of the statistics of the characters which are representative of those to be used. The computer program chooses the best logic configuration where the latter operates on a subset of the scanned image matrix and therefore represents the attributes which are employed as an input to the classification stage.*

**Methods of logic generation**

To reduce the number of logics considered, the following three constraints have been imposed on the generation of recognition switching functions:

1. Only recognition logics which are invariant to translation of the input characters with respect to the logic were considered.
2. Recognition logics consisting of only certain types of n-tuples, with conditions on the positions of each of the *n* points with respect to each other were considered.
3. Specific switching functions on these n-tuples such as AND-ing only or majority logics of the n points were used

Even with these constraints the number of logics generated is still large, in order to prevent it, we must use a selective approach. Also we have to take into consideration the significance of the generated logics, which is then used to determine the redundancies of the set of logics.

**Information measure:** An ideal observer can be postulated for a character reader. The ideal observer specifies the distribution of the states of the input signal to a physical system, based on the state of the output. In this case the input is a character pattern and the output the state of a set of parameters. The ideal observer conserves all information relevant to specifying the input, but no more. Consider a character reader with ***M*** parameters, x1, x2 . . xM used to classify m different characters, c1, c2 . . . cm***.*** Then the conditional probability distribution of the character set **P {cilx},** given the particular state of the ***M*** parameters x, completely describes the input for any state of the ***M*** parameters. If for any probable **x,** this distribution is peaked, that is, one of the characters has probability near one and the other ***m*** - **1** characters have probability near zero, then this is a good set of parameters. If on the other hand, the probabilities are all nearly equal, then the parameter set is poor.



**P {x}**is the probability of the parameter state x and the first sum is taken over all states of the parameter set. This measure is applicable for determining the value of the complete set of parameters **x.**

**Redundancy measure:** When using the previous measure there is no assurance that each of the measurements will not divide the alphabet into the same two parts. During experiments it is shown that the random choice of masks did lead to a random partitioning. Another issue is the number of logics that must be selected to meet a certain specified error rate. Theoretically, if *m* characters are to be classified, log2*m* binary logics which correctly partition the character alphabet into two parts will be necessary and sufficient. However, since the data samples may have many unpredictable variations, some redundancy must be provided if confusions between classes are to be eliminated. Thus, feature code representations of different classes must maintain a certain distance. Following this line of reasoning we have used the following method for minimizing a large number of logic circuits to yield a smaller set with a given minimum distance between pairs of characters.

For a set of ***N*** logical conditions that have been designed to recognize a set of *m* classes successfully. We can compute the pairwise information measure ***lik, j*** for every pair of classes cl and ck and the parameter of each L.C. ***xj***



The values of ***I lk,j*** may be viewed as the elements of a ***C2m*** by ***N*** matrix, ***( Iij )*** where each information value, ***Iij = I lk,j*** indicates the separation power of the **jth** measurement on the **ith** pair of characters *cl* and ck.

The elements of (fij) are now quantized into **ZEROS** and **ONES.** Each element **fij** is set equal to one if it is greater than a certain threshold value **0** and is made zero otherwise. Let us assume that *r* measurements are required to distinguish each pair, that is, a minimum separation distance of *r* is needed. A threshold value **0** is chosen that will produce at least *r* **ONES** in each row of the matrix. Next, the rows are rearranged such that the number of **ONES** in each row increases as I increases, and the number of **ONES** in each column decreases as *j* increases. Each row in the matrix is then checked and the columns marked that will produce *r* **ONES** in each row. Only the marked measurements are preserved.

*Another approach is to utilize random patterns. Uhr and Vossler [119] generate these in a 5 x 5 aperture and then invoke correlation to determine whether the patterns exist in the image.*

The input pattern is transformed into four 3-bit characteristics by each of a set of 5x5 matrix operators, each cell of which may be visualized as containing either a 0, 1, or blank. These small matrices which measure local characteristics of the pattern are translated, one at a time, across and then down that part of the matrix which lies within the mask. The operator is considered to match the input matrix whenever the 0's and 1's in the operator correspond to identical values in the pattern, and for each match the location of the center cell of the 5x5 matrix operator is temporarily recorded. This information is then summarized and scaled from 0 to 7 to form four 3-"bit characteristics for the operator. These represent 1) the number of matches, 2) the average horizontal position of the matches within the rectangular mask, 3) "the average vertical position of the matches, and 4)the average value of the square of the radial distance from the center of the mask.

A variable number of operators can be used in any machine run. This can mean either a number pre-set for that specific run, or a number that begins at zero and expands, under one of the rules described below, up to a maximum of 40. The string of 25 numbers which defines a 5x5 matrix operator can be generated in any of the following ways:

1. A pre-programmed string can be fed in by the experimenter.
2. A random string can be generated; this string can be restricted as to the number of "ones" it will contain, and as to whether these "ones" must be connected in the 5x5 matrix. (We have not actually tested this method as yet.)
3. A random string can be "extracted" from the present input matrix and modified by the following procedure (which in effect is imitating a certain part of the matrix). The process of inserting blanks in the extracted operator allows for minor distortions in the local characteristics which the operator matches.
4. A 5x5 matrix is extracted from a random position in the input matrix.
5. All "zero" cells connected to "one" cells are then replaced by blanks.
6. Each of the remaining cells, both "zeros" and "ones," are then replaced by a blank with a probability of ^.*(k)* Tests are made to insure that the operator does not have "ones" in the same cells as any other currently used operator or any operator in a list of those recently rejected by the program. If the operator is similar to one of these in this respect a new operator is generated by starting over at step 1.

*Bledsoe and Browning [2] choose 75 pairs of elements at random each of which may take on either one of four states. The characters are analyzed in this way and correlation then used.*

*It is possible to find representative features or attributes using the concept of learning machines [88], [7], [1]. Block et al. [3] incorporates a perceptron in the pattern recognition scheme and states the problem as: “Given a set of patterns, determine a set of features, minimal in number, such that each pattern can be formed by the superposition of a subset of these features.” A 5 x 5 pattern array was chosen.*

**Contouring**: Most of the algorithms operate by estimating the value of the surface at a regular grid work of points across the map area. Contour lines are then drawn through this grid by mathematical interpolation.

Computer programs for drawing isoline or contour maps from scattered data points can be categorized into three general groups. These three basic approaches are embellished almost endlessly, however, making direct comparisons between different programs difficult.

The most obvious computer contouring approach is incorporated in various triangulation procedures, which simulate the process of manual contouring. Lines are projected from each data point to the nearest three points, dividing the map areainto triangles. The points where contour lines cross the triangles are established by linear interpolation down the sides of the triangles. The final step is to connect points of intersection that have equal value to form contours. Essentially this process represents the surface as a "geodesic dome" composed of flat triangular plates (IBM, 1965). The principal advantages of this procedure are the directness of the methodology and the fact that all control points must lie on the contoured surface. The principal drawbacks are the non-uniqueness of the triangular mesh, which can result in different patterns of contour lines for the same data, and the extreme slowness of the procedure as compared to gridding routines. An alternative methodology includes the global fit procedures, in which a complex mathematical function of the geographic coordinates is fitted to the control point values. Polynomial trend surfaces and double Fourier surfaces are examples. Basically, global fit methods are an extension of statistical regression procedures into two-dimensional space. The equation which represents the surface is usually calculated so the sum of the squared deviations of the surface from the control point values is a minimum.

Local Fit methods estimate values at the nodes of a regular grid across the map from a weighted average of the control points nearest each grid node. Contours are laced through the grid work by linear interpolation between the nodes to find the points of intersection of the contour levels with the grid lines. Points of common elevation are then connected to form the contour lines. Estimating the regular grid of values is called "gridding" and consists of two steps. First, the nearest neighbors must be found. The simplest procedure is to take the n nearest points to the grid node being estimated. With certain distribution of control points, this may result in unconstrained estimates of the surface, if all the nearest points lie on one side of the node to be estimated. Constraints may be introduced to insure some equitable radial distribution of the nearest neighbors used. These include a quadrant search, where n points must be found in each of four quadrants around the estimated points, and the octant search which carries the concept of radial constraint one step further. The second step is the estimation of grid values from control points that have been located in the first step. The estimates may simply be weighted averages, where the control points are weighted by a function of their distance D from the grid node.

**Skeletonization:** A skeleton is a lower dimensional object that essentially represents the shape of its target object. Because a skeleton is simpler than the original object, many operations, e.g., shape recognition and deformation, can be performed more efficiently on the skeleton than on the full object. The process of generating such a skeleton is called skeleton ex-traction or skeletonization. Skeletons have been extracted from different sources, such as voxel (image) based data, boundary represented, and scattered, and for different purposes, such as shape description, shape approximation, similarity estimation collision, biological applications, navigation in virtual environments and animation.



Local skeletons can be connected to form a global skeleton of the input model. The centroid method is a simple approach that can result in skeletons that do not represent the shape of the object. The second method, based on the principal axis of a component, is slightly more expensive to compute, but leads to improved skeletons in some cases.

**Using Centroids:** One of the easiest ways to construct a skeleton for a component C (in a decomposition) is to connect the centroids of the openings, called opening centroids, on @C to the centroid of C. These openings are generated when a component is split into subcomponents during the decomposition process. Although this approach is simple and generates fairly good results one of the major drawbacks of this type of skeleton is its inability to represent some types of shapes. For example, the skeleton of a cross-like model extracted using its centroids are only a line segment instead of two crossing line segments. The method described next attempts to address this problem.

**Using The Principal Axis:** In this method, we extract a skeleton from a component C (in a decomposition) using the principal axis of the convex hull HC of C. Instead of connecting the centroids of C's openings to the center of mass of C, we connect these centroids to the principal axis enclosed in HC.



Let PA(HC) be a line through the center of mass of HC, parallel to the principal axis of HC. Our method connects an opening centroid to one of the k points on PA(HC)$∩$HC. These k points, denoted by P, evenly subdivide PA (HC) $∩$HC into k+1 line segments. The selection of the value of k is based on the desired minimum skeleton link length. Let P’ $⊂$ P be a set of points to which the opening centroids connect. Figure 5 illustrates P and P’ with circles along PA(HC). Then, the final skeleton S of C contains line segments that connect the opening centroids to P0 and line segments that connect the P’. To minimize the chance of getting a long skeleton with many joints, we match the opening centroids to P so that the cardinality of P’ and the distances from the opening centroids to P’ are minimized. We solve this optimization matching problem using dynamic programming. Details of how we find the optimal solution are discussed in Appendix A. In cases where all the points in P’ lie only on one side of the center of mass c of HC, e.g., P’ in Figure 5(b), line segments that connect to the points in P’ are not enough to represent the entire component. In such cases, the skeleton will connect P’  with the end point of P on the other side of the center of mass c. Similarly, when P’ contains only c, the skeleton will connect c with the end points of P on both sides of c, e.g., the skeleton of the component P1 in Figure 4 (using the principal axis).



The skeleton should reside in the interior of the model and it should encode the “topology" of the model's shape. Thus, using these general criteria, the strategy to compute the quality of a skeleton S is to compare S with its associated component C. This first method checks whether S intersects $∂$C and the second method checks the topological representation of S w.r.t. C. In the third method, we propose an adaptive measurement based on the volume of the component. An important property of these three methods is that the error of the skeleton becomes smaller as the decomposition becomes finer.

**Checking penetration.** Our first method measures the quality of S by checking whether S intersects the component boundary $∂$C. If so, the function Error(C, S) returns a large number (larger than the tolerable value τ). Otherwise, zero will be returned. The consequence is that C will be decomposed if $∂$C $∩$ S $\ne ∅$.

**Measuring centeredness.** In the second method, we measure the offsets of S from the level sets of the geodesic distance map on $∂$C. The value for each point in this map is the shortest distance to its closest opening of C. Ideally; a skeleton should pass through all connected components in all level sets. Therefore, this measurement method simply checks the number of times that S does not do so. Let LC be all the connected components in the level sets of C. We define the error of a skeleton S as: 

where f(lc; S) returns 0 if S intersects component lc, and 1 otherwise, and |LC|is the total number of the connected components in C.

**Measuring convexity.** Our idea for the last quality measurement comes from the observation that in many cases the significance of a feature depends on its volumetric proportion to its “base". For example, a 5 cm stick attached to a ball with 5 cm radius is a more significant feature than a 5 cm stick attached to a ball with 5 km radius. This intuition can be captured by the concept of the convexity of a component C defined as **convexity(C) = vol(C)/vol(HC)** , where vol(X) is the volume of a set X. Thus, we can define the error measurement as:

**Err(C; S) = 1 - convexity(C) …. (4)**

Assume that the skeleton S is a good representation of the convex hull HC. Then, a smaller difference between HC and C means that S is a better representation of C. Thus, although the skeleton S is not included in Equation 4, S is implicitly considered in terms of HC.

**Thinning:** it involves the process of removing of points or layers of outline from a pattern until all lines or curves are of unit width or a single pixel wide. One of the techniques involves the generation of a medial line in which every point is equi-distant from at least two points on the edge of the pattern. In cases of a digital approach a two dimensional array of pixels are considered. Constraints are included so that contour pixels of the skeleton either touch or are on the medial line.

Many thinning algorithms are iterative. In an iteration the edge points are examined against a set of criteria to decide whether the edge point should be removed or not. The algorithms are classified as parallel or sequential. With a parallel algorithm, only the result from the previous iteration takes part in the decision to remove a point in the current iteration, which is suitable for processing by parallel hardware such as an array processor. Whereas the sequential algorithms, on the other hand make use of the result from the previous pass and those results obtained so far in the current pass to process the current pixel.

**Parallel algorithms:** Most of the algorithms use the same approach of visiting all the pixels in the bit-map to identify the dark points. The dark points are then classified into edge points and non-edge points. Only the edge points need to be considered. Tests are conducted on their 8-neighbours to determine wheteher they are break points, end points or non-safe points.

Time complexity of parallel algorithms consists of three components:

1. In every pass in every subiteration, every pixel in the bitmap has to be examined once to identify the dark pixels. The number of operations is proportional to the area of the bitmap.
2. Every dark pixel has to be examined for edge points. The number of operations is proportional to the area of the objects in every pass.
3. The number of passes is related to the “thickness” of the object.

Where the total number of operations in (1) is a product of the number of passes, the number of subiterations per pass and the size of the bitmap, while in (2) the total number of operations is a sum of the sizes of the objects in all the subiterations and in all the passes. Even though the size of the objects reduces progressively after each pass, the total number of operations to determine whether a pixel is an edge point or not is still very substantial. The complexity of (2) increases sharply when the resolution of the image or the thickness of the objects increases.

**Serial algorithms:** in sequential algorithms less memory is required. It has to examine every pixel in the bitmap to distinguish the foreground from the background. Thus time complexity still depends on the size of the bitmap. While a significant reduction in time complexity can be achieved by examining only those points which belongs to the outline of an object. One of the serial algorithms is contour tracing technique which was introduced to deal with nearly thinned objects or thick objects. The contour is a sequence or chain of edge points p0,p1,…,pn, where p0$≡$pn. For a pixel pi, the two pixels just prior to or following it in the chain, namely pi-1 and pi+1 respectively are called the C-neighbors of pi (with p-1$≡$pn-1). The sequence of pixels is usually represented by a chain code, which is a sequence of directions dir0, dir1, …, dirn-1, pointing to the next point in the sequence. For 8-connected contours, diri is in the range between 0 and 7 inclusive, representing the 8 directions. Another variation is to record the change in direction, instead of the absolute angle with respect to the x-axis. The set of chains, together with the coordinates of the starting point’s p0 and a vector describing the direction of the interior, will completely define the bitmap. In contour tracing thinning algorithms, the term “multiple pixel” was defined to describe edge points that satisfies one or more of the following:

1. It is traversed more than once when tracing the set of contours.
2. It has no neighbors in the interior of Q
3. It has at least one D-neighbor which belongs to the contour, but which is not one of its C-neighbors.

After the contour has been traced and the sequence of pixels examined to determine whether it is multiple or not, the contour is removed. Multiple pixels are skeletal pixels and are copied to a bitmap where the skeleton is formed progressively. To ensure connectivity, pixels which are neighbors to the new contour is traced and the operation repeats until all the dark points are removed.

**Thinning by contour generation:** before all the iterations, the bitmap is recoded into chain codes. A chain code is generated for every closed contour describing the outlines of the object. The resulting set of chains represents the outer-most layer of the objects. Also available after the trace are the coordinates of the heads of the chains and a parameter describing the direction of the interior. This parameter *dir\_int* is equal to +1 for a counterclockwise scan of the exterior of an object or for a clockwise scan of an interior hole of the object. It is equal to -1 for a clockwise scan of the exterior or for a counterclockwise scan of an interior hole of the object. The chains, the coordinates of the heads of chains and *dir\_int* will defined the bitmap completely. This is therefore a recoding step, after which the original bitmap is no longer needed.

With the chain codes the outline is plotted on a bitmap S. Every pixel visited will have its value incremented. If S begins with all the pixels having a value of 0, a pixel visited more than once will have a value greater than or equal to 2 and is therefore a break point. When the operation completes the skeleton is formed in S. A chain code describing the skeleton is also available. After plotting the first contour on S, the algorithm goes through a number of iterations. The iteration terminates for a particular contour when there are no more non-safe points in that contour.

**Normalization:**  is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, e.g. the full range of pixel values that the image type concerned allows. It differs from the more sophisticated histogram equalization in that it can only apply a linear scaling function to the image pixel values. As a result the `enhancement' is less harsh.

Before the stretching can be performed it is necessary to specify the upper and lower pixel value limits over which the image is to be normalized. Often these limits will just be the minimum and maximum pixel values that the image type concerned allows. For example for 8-bit graylevel images the lower and upper limits might be 0 and 255. Call the lower and the upper limits a and b respectively.

The simplest sort of normalization then scans the image to find the lowest and highest pixel values currently present in the image. Call these c and d. Then each pixel P is scaled using the following function:



Values below 0 are set to 0 and values about 255 are set to 255.

The problem with this is that a single outlying pixel with either a very high or very low value can severely affect the value of c or d and this could lead to very unrepresentative scaling. Therefore a more robust approach is to first take a [histogram](http://homepages.inf.ed.ac.uk/rbf/HIPR2/histgram.htm) of the image, and then select c and d at, say, the 5th and 95th percentile in the histogram (that is, 5% of the pixel in the histogram will have values lower than c, and 5% of the pixels will have values higher than d). This prevents outliers affecting the scaling so much.

Another common technique for dealing with outliers is to use the intensity histogram to find the most popular intensity level in an image (i.e. the histogram peak) and then define a cutoff fraction which is the minimum fraction of this peak magnitude below which data will be ignored. The intensity histogram is then scanned upward from 0 until the first intensity value with contents above the cutoff fraction. This defines c. Similarly, the intensity histogram is then scanned downward from 255 until the first intensity value with contents above the cutoff fraction. This defines d.

**Vectorization:** i.e. raster–to–vector conversion is of course a central part of graphics recognition problems, as it deals with converting the scanned image to a vector form suitable for further analysis. Raster–to–vector methods know that there are several steps in this process.

The first step is to find the lines in the original raster image. Whereas the most common approach for this is to compute the skeleton of the image, a number of other methods have been proposed. The next step is to approximate the lines found into a set of vectors. This is performed by some polygonal approximation method, and there are many of these around as well, using different approximation criteria. After approximation, it is often necessary to perform some post-processing, to find better positions for the junction points, to merge some vectors and remove some others, etc. A last step sometimes performed is to find the circular arcs.

Chain Codes: Chain codes describe an object by a sequence of unit-size line segments with a given orientation (see Section 4.2.2). The first element of such a sequence must bear information about its position to permit the region to be reconstructed. The process results in a sequence of numbers (see Figure 8.6); to exploit the position invariance of chain codes the first element, which contains the position information, is omitted. This definition of the chain code is known as Freeman's code [Freeman, 1961]. Note that a chain code object description may easily be obtained as a by-product of border detection; see Section 6.2.3 for a description of border detection algorithms. If the chain code is used for matching, it must be independent of the choice of the first border pixel in the sequence. One possibility for normalizing the chain code is to find the pixel in the border sequence which results in the minimum integer number if the description chain is interpreted as a base 4 number—that pixel is then used as the starting pixel [Tsai and Yu, 1985]. A 'mod 4 or mod 8 difference code, called a chain code derivative, is another numbered sequence that represents relative directions of region boundary elements, measured as multiples of counter-clockwise 90° or 45° direction changes (Figure 8.6). A chain code is very sensitive to noise, and arbitrary changes in scale and rotation may cause problems if used for recognition. The smoothed version of the chain code (averaged directions along a specified path length) is less noise sensitive.

**Distance transforms:** A distance transformation is an operation that converts this binary image to a grey-level image where all pixels have a value corresponding to the distance to the nearest feature pixel. Computing the distance from a pixel to a set of feature pixels is essentially a global operation. Unless the digital image is very small, all global operations are prohibitively costly. Therefore algorithms that consider only a small neighborhood at a time, but still give a reasonable approximation to the Euclidean distance are necessary.

An original binary image, to which the DT is to be applied, consists of feature pixels with the initial value zero, and non-feature pixels with the initial value infinity, i.e., a suitably large number.

The computation of the DT is either parallel or sequential. In the parallel case the center of the mask is placed over each pixel in the image. The local distance in each mask-pixel cn, is added to the value of the image pixel "below" it (including the central zero). The new value of the image pixel is the minimum of all the sums. The process is repeated until no pixel value changes, i.e., the number of iterations is proportional to the largest distance in the image. The parallel algorithm is thus:



where $v\_{i,j}^{m}$ is the value of the pixel in position (i, j) in the image at iteration m(k, l) is the position in the mask (the center being (0,0)), and *c(k, l)* is the local distance from the mask.





The sequential algorithm also starts from the zero/infinity image. The masks are passed over the image once each: the forward one from left to right, and from top to bottom, and the backward one from right to left and from bottom to top. The new value of the "central" image pixel is the minimum of the sums of the image pixel values and the local distances cn as before. After these two passes the distance transform is computed. The sequential algorithm is thus:

