DETECTION OF LINES AND CIRCLES IN MAPS AND ENGINEERING DRAWINGS

D. P. Argialas
Remote Sensing Laboratory
Dept. of Surveying Engineering
National Technical University
9 Heroon Polytechniou Str.
Zografos 15773, Greece

S. Krishnamurthy
Dept. of Computer Science
Louisiana State University
Baton Rouge, LA 70803
USA

Commission: III

Purpose:

The development of semi-automated scanning/conversion systems can reduce the cost of the manual intensive process of converting maps and engineering drawings into a form usable by the CAD/GIS systems. A number of scanning/conversion algorithms have been developed that manipulate raster data and extract meaningful shapes such as lines, circles, arcs, and other geometric elements that appear in engineering drawings. In this research effort, a hybrid method for detecting both lines and circles on maps and engineering drawings was implemented and tested. This hybrid method uses the parameter space of the Muff transform for detecting lines and it uses the parameter space of the Hough transform for detecting circles. For every pixel of the scanned image, the algorithm determines two more pixels in its neighborhood and checks whether these three pixels are on a line or on a circle. If they form a line then the parameter space of the Muff transform is updated. If they lie on a circle then the parameter space of the Hough transform is updated. After processing all pixels, the parameter space is checked for the presence of lines and circles. Lines and circles found in the parameter space are transformed back to the image plane. The method was tested on several maps and engineering drawing, and the results were encouraging.

KEY WORDS: Line Detection, Hough Transform, Muff Transform, Image analysis, scanning/conversion.

INTRODUCTION

When existing as-built engineering structures are expanded, modified or upgraded, it is generally necessary to determine the physical and operational characteristics of the structure using old plans and specifications. It is estimated that for certain projects information has to be extracted manually from over 30,000 drawings (E size). In addition, once the appropriate information is extracted, it is not in a form that can readily be utilized in a modern computer aided design and drafting system (CADD), which provides an efficient means for creating, storing and updating engineering drawings. Manual entry of drawings into a computer database is a slow, expensive, and tedious process. Thus, there is a need for automated or computer-assisted systems for transforming existing drawings to a form suitable for CADD/GIS.

Scanning/conversions systems attempt to capture information in an engineering drawing or map and convert it to a format compatible with that of the CAD/GIS systems. However, building such an automated system is by far a complex task because of the abundant information present in the drawings and the difficulty in extracting adequate descriptors of the relationships between the various entities of engineering drawings. In the raster format a drawing is represented by a rectangular array of pixels, while in the vector format a drawing is represented by geometric elements, such as lines, circles, arcs, arrows, etc. Thus one of the objectives of a conversion system is to detect lines and circles appearing in engineering drawings.

There has been a tremendous growth in algorithms for extracting lines and circles in recent years (Wallace 1985). The line and circle algorithms are based either on tracing contiguous edge pixels which might fall on a line or circles [Joseph 1989] or by transforming the image to a parameter space and check peaks in the parameter space to detect the parameters of line and circle. The former methods are known as line tracing methods, while the later method is known as the Hough Transform method. Line tracing methods do not perform well in the presence of noise in the image. On the other hand, the Hough transform is less sensitive to the presence of noise and occlusion in images [Kimme 1972]. However, the Hough transform requires enormous storage space for computation, and detects only the parameters of lines and not their end points. Additional processing is therefore required to detect the end points.

There are many algorithms that use the Hough Transform as a basis to detect lines and circles [Yuen et al 1990, Iannino and Shapiro, 1978]. While most of these methods have met with considerable success, each of them have some drawbacks that prevents them from being totally effective in a general setting. One of the methods is known as Muff transform [Wallace 1987]. The Muff transform sub-divides the image into a set of rectangular regions, and checks for lines that cut the boundary of the rectangular region. One of the advantages of the Muff transform is that it detects the end points of a line. But it is not applicable for the detection of circles.

The "Cord method" by Amir [1990] determines the centers of circles by drawing a semi-circle of arbitrary radius around each point in the image, finding all points that lie on this semi-circle, drawing lines from each original point to all other points that lie on the semi-circle, then computing the equations of the perpendicular bisectors of each of these lines and finally finding candidate circle centers defined by the solution of every possible pair of these equations. By employing an accumulator array, of similar dimensions with the image plane, for every solution of a pair of equations, the value of the appropriate accumulator cell is incremented by one. When all points are processed, the accumulator array is searched for peak values which will be the centers of circles. The Cord method provides the coordinates of the centers of circles, but not the radius of circles. However, if the radii of the circles in the image are known, a significant reduction in the computations is achieved [Amir 1990].

The method by Cao and Deravi [1990] is a modified Hough transform method that assumes every point lies on a circle and searches two more points and in the neighborhood of the original point by employing some heuristic. The method is based on the principle that at most three points are needed for the computation of the center coordinates and radius of a circle. The x and y coordinate values of the three points are substituted in the equation of a circle to yield three equations which can be solved to compute the center coordinates and radius of a potential circle present in the image. The accumulator array is similar to the 3-dimensional array used in the Hough transform for circle detection and it consists of layers. Each layer is associated with a particular radius and center of possible circles. The computed coordinates of the center and radius of the circle are used as an index to the accumulator cell and the corresponding cell is incremented by a unit value. All pairs of points that are present in the neighborhood of the original point are considered for each and every original point. In the last step, the accumulator array is searched for peak values for the detection of circles.

In this research effort, a modification of the Hough/Muff/Amir/ and Cao and Deravi transform methods was designed and implemented for the detection of lines and circles appearing in engineering drawings and maps.

METHODOLOGY

Hough transform provides the parameters associated with a geometric element such as a line or a circle. But it does not provide the end points of lines. On the other hand, Muff transform provides the end points of a line. However, Muff transform cannot detect circles and other geometric elements that are defined by more than two parameters. In order to make use of the advantages of both the Muff and Hough transforms and to reduce the problems encountered in both Hough and Muff transforms, a Hybrid method for detecting both lines and circles was designed. This Hybrid method uses Hough and Muff transform techniques as a basis together with the techniques developed by Amir (1990) and Cao and Deravi (1990). The Cord method [Amir 1990] described in the previous section draw a semi-circle around each point in the image, checks for points on the semi-circle and uses these points to detect the center of circles. The Hybrid method considers a full circle to be drawn around each point and tries to detect three points necessary to determine both lines and circles.

Since a binary image will usually be represented by a sparse two-dimensional array, which unnecessarily will waste a lot of memory space, a binary image is represented as a set of "points" with coordinate pairs (x_i,y_i) where i=1, M, M being the total number of pixels whose value is one.

Use of Accumulator Arrays in the Hybrid Method

The accumulator array used in the Hough transform for the detection of lines is not suitable for detecting the end points of lines because the accumulator array is not compatible with the image array. Also, considerable amount of computation has to be carried out following the Hough transform method to detect the end points of lines. Muff transform, on the other hand, directly provides the end points of lines in the accumulator array itself because the accumulator array chosen in the Muff transform is compatible with the image array. Therefore, the proposed Hybrid method chooses the accumulator array used in Muff transform for the detection of lines. The Hough transform detects circles, and directly provides the coordinates of the circle center and the radius because the three-dimensional accumulator array matches with the image array. Hence in the Hybrid method the accumulator array of the Hough transform was used for detecting circles. Thus, the Hybrid method maintains separate accumulator arrays for lines and circles, and takes advantage of both the Hough and Muff transforms. Additionally, the Hybrid method, similar to the Muff transform, divides the image into a number of sub-images in order to have a small accumulator array to detect short lines.

Computation of three points

The Hybrid method is based on the principle that at most three points are needed for the computation of the center coordinates and radius of a circle. The equation of a circle is given by

$$(x-a)^2+(y-b)^2=r^2$$

where (a, b) are the coordinate of the center and r the radius of the circle. In this equation, there are three unknown variables: a, b and r. If there are three points (x_1, y_1), (x_2, y_2), (x_0, y_0) that lie on circle, the center coordinates (a, b) and the radius r can be computed by solving simultaneously the three equations.

$$(x_1-a)^2+(y_1-b)^2=r^2 \quad (1)$$
$$(x_2-a)^2+(y_2-b)^2=r^2 \quad (2)$$
$$(x_0-a)^2+(y_0-b)^2=r^2 \quad (3)$$

To simplify the solution of the above equations an easy method is to expand them resulting in

$$x_1^2+y_1^2-2ax_1-2by_1+a^2+b^2=r^2$$
$$x_2^2+y_2^2-2ax_2-2by_2+a^2+b^2=r^2$$
$$x_0^2+y_0^2-2ax_0-2by_0+a^2+b^2=r^2$$

Again, these equations are of the form

$$2ax+2by+c=(x^2+y^2) \quad (4a)$$

$$\text{where } c=r^2-a^2-b^2 \quad (4b)$$

When the three points (x_1, y_1), (x_2, y_2) and (x_0, y_0) are substituted in (4a), they result in

$$2ax_1+2by_1+c=(x_1^2+y_1^2) \quad (5)$$
$$2ax_2+2by_2+c=(x_2^2+y_2^2) \quad (6)$$
$$2ax_0+2by_0+c=(x_0^2+y_0^2) \quad (7)$$

The above three equations constitute a system of linear equations with three unknown variables, namely, a, b and c. The three equations can be solved by Crammers rule to obtain the coordinates of the center (a, b). The radius r of the circle is given from equation (4b), i.e.,

$$r = (a^2 + b^2 - c)^{1/2}$$

From the above discussion, it is seen that with a set of three points, the center coordinates and the radius of a circle on which the three points lie can be computed by solving the system of linear equations (5), (6) and (7). To determine the three points that may lie on a circle present in the image, for every point $P_0(x_0, y_0)$ considered in the sub-image, the Hybrid method initially assumes that point to be on a circle. This point, by assumption, may be one of the three needed points in order to solve equations (5), (6) and (7.) Thus, the problem is simplified to the determination of two more points in the vicinity of the point P_0.

The Cord Method [Amir 1990], described earlier, can not be used as is for the determination of circles and lines. It can, however, be used, with a modification, in the determination of two or more points needed in solving equations (5), (6) and (7) to compute the center and radius of a circle as well as to check for presence of lines. When a semi-circle is drawn around the point P_0 instead of a full circle, only one point P_1 (where the semi-circle intersects the actual circle present in the image) is obtained (Figure 1). If a full circle is drawn around each point, another point P_2 can be determined as shown in Figure 1. Hence, the system of linear equations can be solved by substituting the x and y coordinates of the three points P_0, P_1 and P_2. The important consideration in choosing the points P_1 and P_2 is that the distance between the points P_1 and P_2 should be sufficient enough such that short distances between them should be avoided for the computation of circle centers [Cao and Deravi 1990]. For example pairs of points (P_2, P_5), (P_1, P_3) and (P_4, P_5) in the Figure 3 can be omitted because it is impossible for a circle to contain any pair of the points and point P_0. The Hybrid method recommends that the distances between them should be more than the radius of the circle drawn around the point and less than the diameter of the circle. If short distances in-between the points exist, i.e., when the distance is less than the radius of the circle drawn, then the points may not be suitable for detecting the center coordinates of circles because of the digitization errors involved in the position of the points [Cao and Deravi, 1990]. As a result of the digitization errors, the computed center coordinates may be shifted from the actual center coordinates of the circles. In order to avoid such short distances in-between points, the Hybrid method draws a circle (instead of a semicircle of the cord method) around each point P_0 (Figure 1) of the sub-image. The circle drawn around each point in the image can be called a "ring" in order to distinguish it from the actual circles present in the image. The Hybrid method uses the ring to detect the center and radius of possible presence of circles and also uses it checks for the presence of lines as explained in the following.

Detection of lines and circles

Let the image be divided into a number of sub-images of equal size, and let one of these sub-images be considered (Figure 2). The accumulator array for the detection of lines is same as the accumulator array in the Muff transform [Wallace 1987]. In other words, the bottom, right and top sides of the sub-image are chained together to map onto the rows of the accumulator array and the right, top and left sides of the sub-image is chained together to map onto the columns of the accumulator array. This accumulator array for line detection is called the Muff array.

The accumulator array for the detection of the circles in the sub-image is similar to the accumulator array in the Hough transform for circle detection. In other words, the accumulator array can be visualized as a stack of two-dimensional arrays, with each two-dimensional array indexed by a single radius. Thus, the whole stack is a three-dimensional array wherein each cell in the array is indexed by three parameters, namely, the center coordinates (a, b) and the radius r. Both the

Muff and Hough arrays are initialized to zero when a new sub-image is considered.

For each point P_0 in the sub-image, the Hybrid method draws a ring with P_0 as center (Figure 3) Let the radius of the ring be "R". The points on this sub-image that cut this ring are stored in an one-dimensional array called "Bound." In the worst case, the number of points in the array Bound may vary up to a maximum of approximately 6 times R (i.e., circumference length of the circle: 2 times π times R). Usually, the number of image points on the ring will not be greater than 10 points because the engineering drawings contain widely spaced long and thin lines. However this number may vary from one scanned drawing to another. Let the number of points in the array Bound be *NUM*.

Every possible pair of points P_1 and P_2 in the array bound are considered for possible presence of lines or circles. There are at most *NUM*(*NUM*-1)/2 combinations of pairs of points from the array bound. If the distance between any pair of points is less than the radius of the *ring* then the pair may be ignored for subsequent analysis because neither a line could pass through them nor a circle could contain them. Using the three points P_0, P_1 and P_2, the Hybrid method determines whether the points are colinear or not. A simple heuristic was constructed to check for colinearity by employing the distance between the points P_1 and P_2. If the distance was equal to twice the radius of the ring, then the points were considered colinear (points P_0, P_1 and P_2 in Figure 4). In this case the two points where this line intersects the four sides of the sub-image were determined as follows.

Let the coordinates of points P_0, P_1 and P_2 be (x_0, y_0), (x_1, y_1) and (x_2, y_2) respectively. The point P_0 can be ignored for further analysis because P_0 is in the center of line joining of P_1 and P_2. Moreover, the points P_1 and P_2 are sufficient to compute the point of intersection. Now, the task is to compute the points of intersection of the line segment joining P_1 and P_2 with any two sides of the sub-image considered. The parametric equation of the line segment joining P_1 and P_2 can be represented [Hill 1990] as

$$\mathbf{T}=P_1(1-r)+r*P_2 \tag{8}$$

where $\mathbf{T}$ is any point on the line joining the points P_1 and P_2 and "r" is a number varying from zero to one. When r is zero and it is substituted in (8) point P_1 is obtained. When r is one and it is substituted in (8), P_2 is obtained. when r = 0.5 then P_0 is obtained because P_0 is the middle point of line joining P_1 and P_2 (Figure 4). If r is varied from 0 to 1, then the coordinates of any point can be obtained by substituting the x and y coordinates in (8) resulting in

$$\mathbf{T}_x=P_{1x}*(1-r)+r*P_{2x} \tag{9}$$

$$\mathbf{T}_y=P_{1y}*(1-r)+r*P_{2y} \tag{10}$$

where P_{1x}, P_{1y}, P_{2x} and P_{2y} are the x and y coordinates of P_1 and P_2. $\mathbf{T}_x$ and $\mathbf{T}_y$ are the x and y coordinates of any point on the line. Similarly the equations of the four lines (or sides of the sub-image in parametric form) were represented (Figure 2) as follows

$$\mathbf{B} = \boldsymbol{BL}\ (1-b) + b*\boldsymbol{BR} \tag{11}$$
$$\mathbf{R} = \boldsymbol{BR}\ (1-r) + r*\boldsymbol{TR} \tag{12}$$
$$\mathbf{T} = \boldsymbol{R}\ (1-t) + t*\boldsymbol{TL} \tag{13}$$
$$\mathbf{L} = \boldsymbol{L}\ (1-l) + l*\boldsymbol{BL} \tag{14}$$

BL, ***BR***, ***TL*** and ***TR*** are the bottom left, bottom right, top left and top right corner points of the sub-image respectively. Equation (11) represents the bottom side of the sub-image, equation (12) represents the right side of the sub-image, equation (13) represents the top side of the sub-image, equation (14) represents the left side of the sub-image. **B** is any point on the bottom side, R is any point on the right side, **T** is any point on the top side and **L** is any point on the left side of the sub-image.

All of the above equations are used to determine on which two of the sides the line joining P_1 and P_2 intersects (Figure 5). Let the points of intersection be P_{d1} and P_{d2}. The distances of these two points along the boundary of the sub-image from the bottom left corner of the sub-image is computed directly. Let them be d_1 and d_2. These two distances are used as an index in the Muff array, and the corresponding cell is incremented by unity.

If the points P_0, P_1 and P_2 do not lie on a line (i.e., they are not colinear) then the three points are checked if they lie on a circle (Figure 6). In other words, these points are used to determine the center coordinates and radius of the circle that contains these three points. This is done by substituting the (x, y) coordinates of the three points in the equations (5), (6) and (7). Then the equations are solved by Crammer's rule to compute the center (a, b) and radius r (Figure 7). The parameter space is the same as the parameter space of the Hough Transform for circle detection, i.e., a three dimensional array having a, b and r as one of the axis. The accumulator cell corresponding to the value (a, b, r) is incremented by unity.

Similarly, another two pair of points in the array bound is considered and the center and radius are computed, and the accumulator cell in the Muff or Hough array is incremented by one count based on the whether a line or a circle is found. After all points from the array bound are considered, the next image point in the sub-image is considered and the process is repeated.

After processing all points in the sub-image, the Muff and Hough arrays are inspected to determine the peaks or high values. If a high value is found in the Muff array, then the indices of the cells in the array are used for the determination of end points of the line that intersect two of the sides of the sub-image. For example, if d_1 and d_2 are two distances (Figure 5), then the coordinates of the end points of a line present in the sub-image are given by (d_1, 0), (2L + B - d_2, B). Likewise, for a high value in the Hough array, the indices of the Hough array directly provides the center coordinates and radius of a circle present in the sub-image.

The end points of lines and center coordinates and radius of circles that are determined in the sub-image are stored in a data structure. Then the next adjacent sub-image is considered. Again both the Muff and Hough arrays are initialized to zero and the Hybrid method is applied over the new adjacent sub-image.

SYSTEM IMPLEMENTATION

The Hybrid method was implemented on the X11 version of X windows loaded on the SUN 3/260 workstations, and the computer program was coded in the "C" programming language. The SUN workstation had 16 Mbytes of main memory and 280 Mbytes of hard disk storage. The SUN workstations employed a high-resolution bit-mapped screen capable of displaying both raster and vector images. The display and manipulation of image data was accomplished using some of the X windows graphic utilities.

A schematic diagram of the system architecture is shown in Figure 8. A user sends requests to the X server through the Xlib graphic utilities provided by X windows. These requests are processed appropriately by the X server and the actions are returned to the user. The Image processing/analysis routines such as enhancement of images, histogram, edge detection, line and circle detections are coded in a separate library called "Image-lib". These routines use some of the routines of Xlib for accessing the pixel values, manipulating and storing them. Thus, the Image processing and analysis routines are not portable because the image processing routines can be executed only on systems which run the X server.

RESULTS AND DISCUSSION

The performance of the Hybrid method was tested on a number of engineering drawings and maps. To conserve space, the illustrations shown in this following, display a composite for each drawing, whose components are both the original drawings/maps and the identified end points of lines or centers of circles, which are shown as "thick spots." The engineering drawing of Figure 9 has been scanned using a Hewlett Packard scanner connected to an Apple Macintosh computer. The resolution selected was 75 DPI. The digital image was stored in PICT format which was converted into "MCID" format and was ported to SUN using "Host Access" running on the Macintosh which was connected to SUN through a network. The image processing routines written to implement the Hybrid method were developed under SUN and they could access the MCID format.

The scanned image was preprocessed with a 3x3 spatial template to enhance the background of the image. The preprocessed image was then converted into a binary image by a simple thresholding process. The radius of the ring selected in the hybrid method was 10 pixels. The points on the ring were stored as a sequence of x and y coordinates. The maximum number of points in the array Bound was set to 56. This is because at most 56 points could be on the ring. The binary image was then sub-divided into a number of subimages of equal width and height (e.g. white lines in Figure 9). The size of the sub-image was 100 x 100 points. Each one of the sub-images was considered one at a time and they were processed using the designed Hybrid method. The size of the Muff array and the Hough array depends directly on the size of the sub-image selected. For a sub-image of size w by w pixels, the Muff array size of 3*w by 3*w is sufficient because only one third of the Muff array is searched for peak values. The size of the sub-image selected was 100 x 100, so the size of the Muff array was 300 x 300. The size of each layer in the Hough array was the same as the sub-image. The cells of the accumulator array which had high values were distinguished from the cells that had low values by choosing a threshold value. The threshold value depends on the number of points in each of the sub-images. Therefore, the threshold value was specified interactively for each of the sub-images. The detected end points of the line segments in each of the sub-images are shown by dark spots or dots (Figure 9). A synthetic image was constructed for testing the algorithm for circles. The detected centers of circles are shown by a thick white spot at the center of the detected circles (Figure 10). Two more pictures, one of a drainage pattern (Figure 11) and the other of a map (Figure 12) are shown together with the detected end points of their lines.

The Hybrid method detected some of the lines and circles in the drawings and maps. The performance of the hybrid method could be improved by selecting an optimal threshold value used to detect the peak values in both the Muff and Hough arrays. Even though a simple threshold value provided good results, clustering techniques in detecting peaks may provide better results. The simple threshold values were selected interactively for each of the sub-images and thus it was not optimized.

The Hybrid method selected the size of the sub-images having some a prior information about the drawing. An effort to study the effects of the sizes of the sub-images on the performance of the Hybrid method was not considered. But, it is expected that as the size of the sub-image is increased, the chances of missing short lines that do not intersect the boundary of the sub-image will also increase. On the other hand, as the size of the sub-image selected is decreased, the circles in the sub-images will be broken into short arcs which are difficult to detect. But, short lines could be detected by the Hybrid method. Moreover, both the Muff and Hough array have to be searched very often which is very time consuming. Thus, a optimal size of the sub-image has to be selected in order to detect short lines and full circles.

The Hybrid method was tested with the radius of the ring equal to 10 pixels. An effort to study the effects of various radii of the ring on the performance of the Hybrid method was not considered. If a larger radius of the ring say R = 20, is selected then approximately 120 image points ($2\pi R$) have to be checked if they lie on the ring. This is a very time consuming process. If a small radius of the ring is selected, say 5, then it is impossible to detect circles because of the short distances present in the points that lie on the ring [Cao and deravi 1990]. Thus, an optimal size of the ring has to be selected in order to minimize the computation time and avoid missing full circles.

One of the disadvantages of the Hybrid method, just like Hough or Muff transform, is that when it divided the image into a number of sub-images, it actually broke the linear and circular features in the image into smaller linear features and arcs respectively. Once these broken linear features are detected by the Hybrid method, a considerable amount of post-processing has to performed to unite the detected lines. Similarly, circles broken into arcs were not detected. However, the chances of loosing information such as shape, is more when the image is divided into sub-image. Thus more research need to be performed to remedy such information loss.

CONCLUSIONS

The major goal of this research effort was to detect lines and circles appearing in engineering drawings. To do so the Hybrid method first sub-divided the image into smaller sub-images. By dividing the image into a number of sub-images, the Hybrid method detected the points of intersections of lines with any two sides of the sub-image. These points of intersection were actually the end points of line segments present in the sub-image. In each one of these sub-images, the Hybrid method selected a point and draw a *ring* with that point as center. Then it searched and found all image points that lie on this ring. It considered combinations of pairs of these points along with the center point and checked if they were colinear. If the three points were colinear, the points of intersections of the identified line with the sub-image were computed. If the points were not colinear, the Hybrid method checked for the presence of a circle that contained the three points. If a circle contained them, the coordinates of the center and radius of the circle were computed. The Hybrid method maintained a 2D Muff accumulator array similar to the Muff transform for identifying the end points of line segments, and also maintained another 3D Hough accumulator array for finding the coordinates of centers of circles. After processing all points in the sub-image, peak values in both the Muff and Hough arrays were found for the presence of actual lines and circles in the image. The method was tested on several maps and engineering drawing, and the results were encouraging. To alleviate some of the problems encountered with the Hybrid method more research needs to be performed.

REFERENCES

Amir. I, 1990. " Algorithm for the center of circular fiducials", Computer, Vision, Graphics and Image Processing, 49: 398-406.

Cao, X., and F. Deravi, 1990. "An efficient method for multiple-circle detection." IEEE, 744-747.

Hill, F, 1990. "Computer Graphics", Macmillan Publishing Company.

Hough P.V.C., December 1962. "Method and means for Recognizing complex patterns", US patent 3, 069, 654.

Iannino. A, and Shapiro, S.D., 1978. " A survey of the hough transform and its extensions for curve detection". IEEE, 32-38.

Joseph, S., 1989. "Processing of Engineering Line Drawings for Automatic Input to CAD". Pattern Recognition, 22(1):1-10.

Kimme, C., D. Ballard, and J. Sklansky, 1975. "Finding Circles by an Array of Accumulators". Communications of the ACM, 18(2):120-122.

Wallace, R., 1987. "A Modified Hough Transform for Lines". IEEE, 665-667.

Yuen, H., J. Princen, J. Illingworth, and J. Kittler, 1990. "Comparative study of Hough Transform methods for circle finding". Image and Vision Computing, 8(1):71-77.

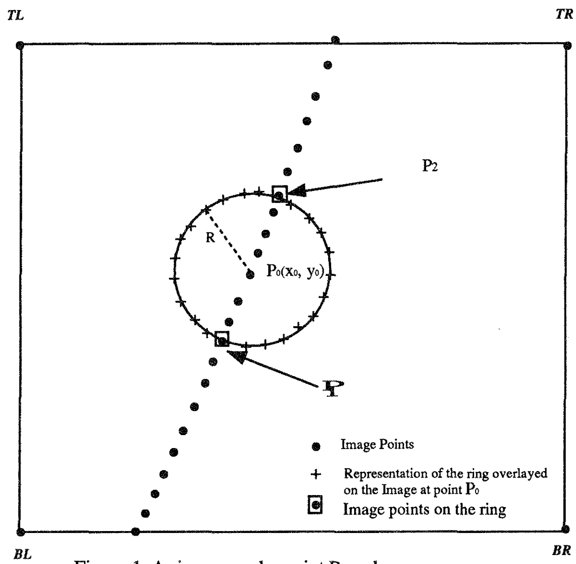

Figure 1. A ring around a point P and a line crossing the ring at two points

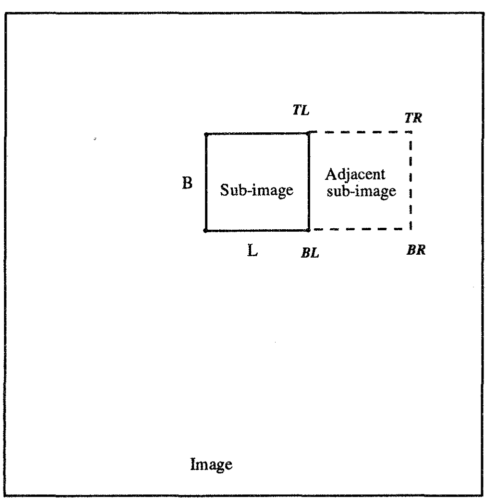

Figure 2. A sub-image of size L x B and an adjacent sub-image

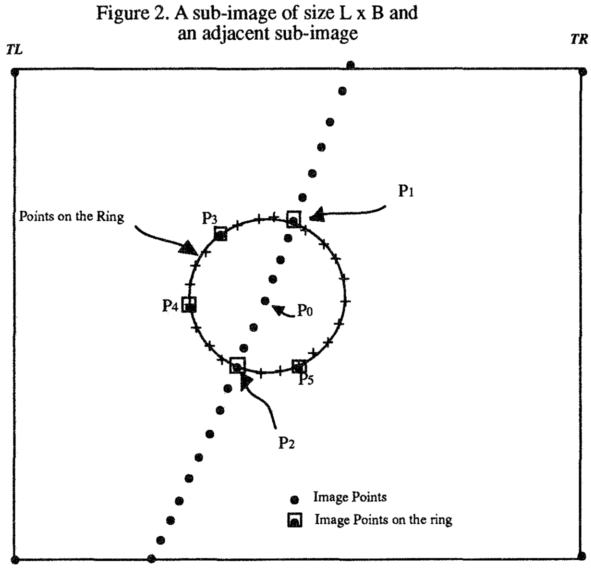

Figure 3. A ring around an image point P. Points that are on the ring are stored in array Bound

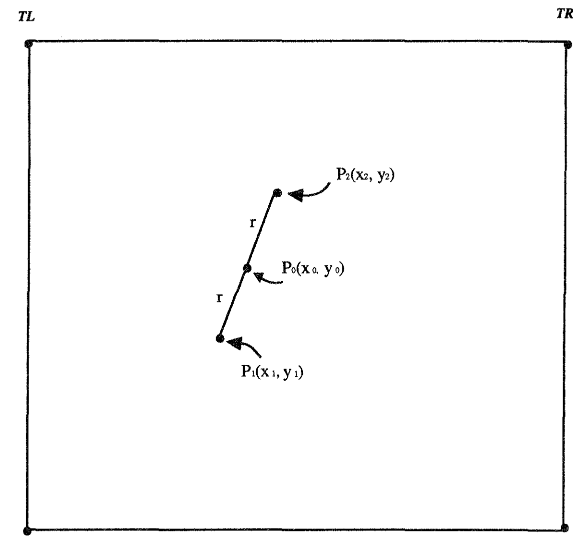

Figure 4. Colinear points on the Ring

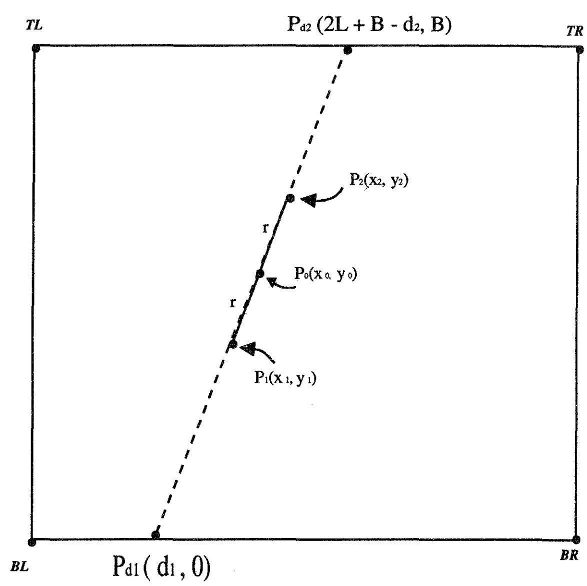

Figure 5. Intersection points (Pd1, Pd2) of a detected line and the boundaries of the sub-image.

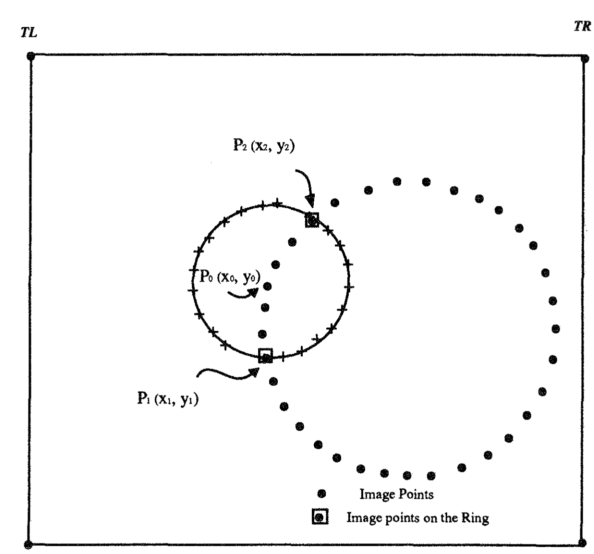

Figure 6. Three non-colinear image points on the ring.

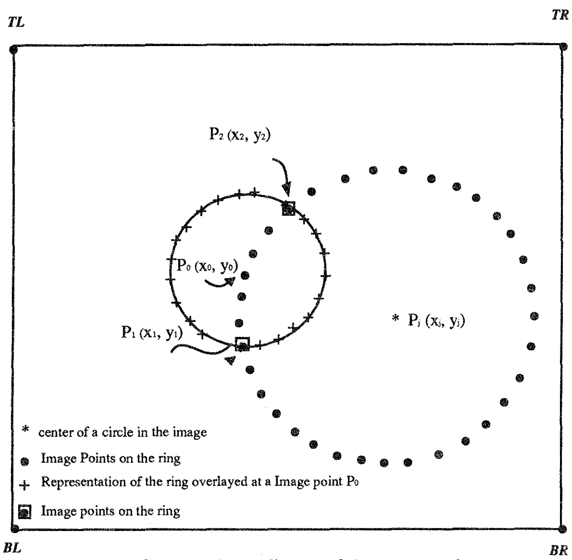

Figure 7. Computed coordiantes of the center and radius of a circle.

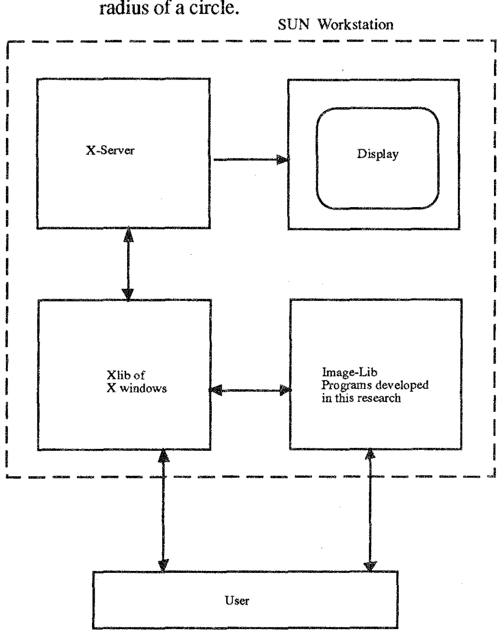

Figure 8. Overview of the system

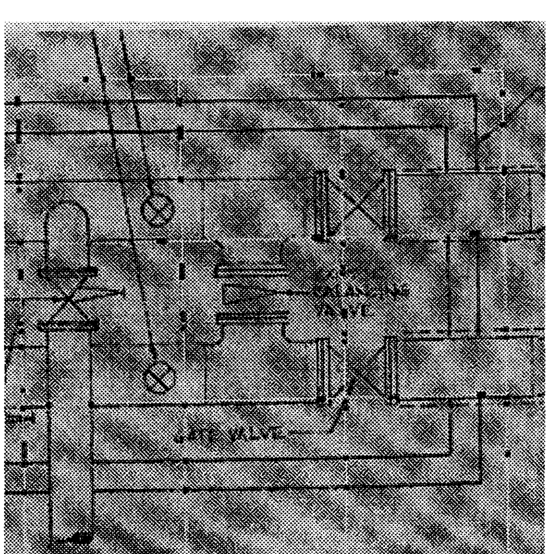

Figure 9. Detected end points of lines in an engineering drawing

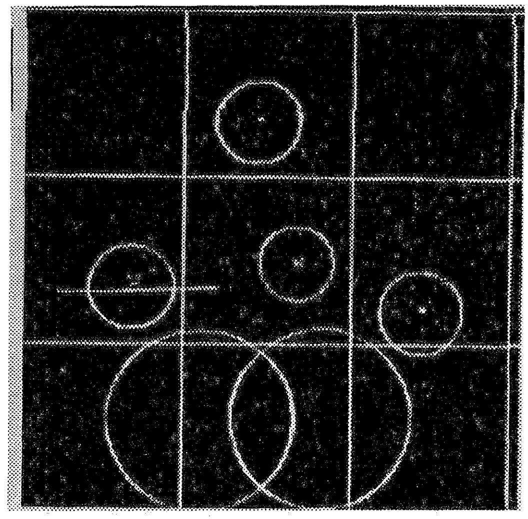

Figure 10. Detected centers of circles.

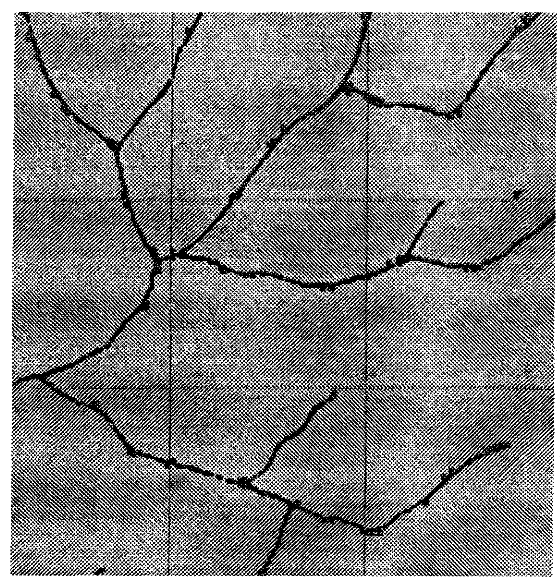

Figure 11. Detected end points of lines for a rectangular drainage pattern

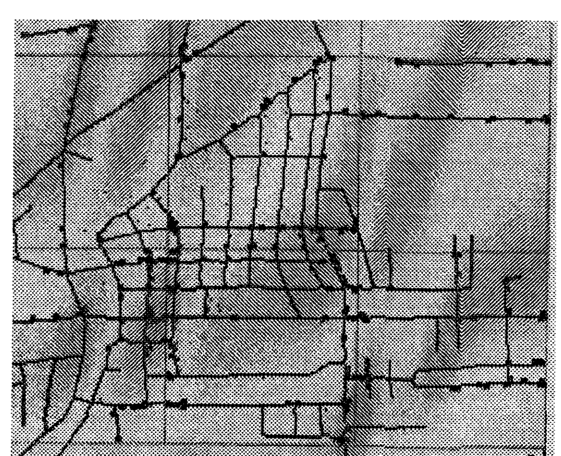

Figure 12. Detected end points of lines on a map.