Feature Extraction Using the Hough Transform

Tara Ferguson

***Abstract*-This paper contains a brief literature survey of applications and improvements of the Hough transform, a description of the Hough transform and a few of its algorithms, and simulation examples of line and curve detection using the Hough transform.**

I. Introduction

The Hough transform is a technique used to detect lines, curves, and objects in an image using the concept of parameter space. It has been widely used in a variety of applications, some of which are described below.

The Hough transform is used to extract linear features from geoscientific datasets [1]. In the estimation of moving lateral vehicle locations for driving assistance using wheel shape information, the projected wheel shape, which is an ellipse, is detected using the Hough transform [2]. The method for adaptive time-varying cancellation of wideband interferences in spread-spectrum communications assumes a parametric interference model to estimate the interference parameters by means of the generalized Wigner-Hough transform [3]. Implementation of a Hough transform for the characterization of rock features in acoustic borehole images is considered because the planar fractures ideally result in sinusoidal curves having a period equal to the perimeter of the borehole [4]. An automatic map-based road detection algorithm for spaceborne synthetic aperature radar images uses the Hough transform [5]. A method for producing a geometric model of the objects in a robotic system for the purpose of real-time planning of collision-free trajectories is constructed, and the modeling process is based on the Hough transform [6]. Photographs were digitized and analyzed to characterize lens and lens nucleus shape as a function of age by the Hough transform and other image analysis methods [7].

A large amount of research also has been done on ways to improve the Hough transform because it requires a lot of storage and computation. Here are a few examples of these improvements. A model for object recognition using the genetic Hough technique decreases the huge amount of storage needed for the Hough space [8]. Curve detection techniques are implemented faster and more accurately using the Hough transform [9]. The progressive probabilistic Hough transform (PPHT) and the probabilistic Hough transform are used to minimize the amount of computation needed to detect lines [10].

In section IV, I used the classical Hough transform to implement the detection of a line and circles that are imperfect in images that contain noise and other features.

II. Hough Transform [11]

A simple way to understand the concept of the Hough transform is by looking at an example of detecting a straight line in an image. Consider the point (x_1,y_1) on the line $y=mx+b$ in Fig. 1(a).

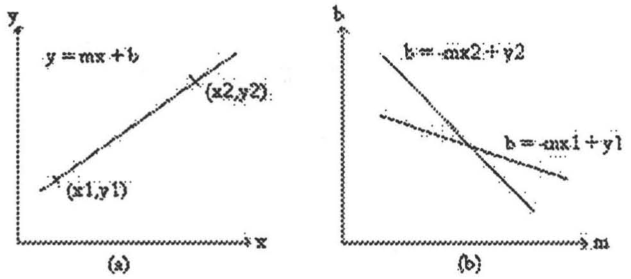

Fig. 1. A line (a) in image space (b) in parameter space

All lines going through this point can be described by the equation $y_1=mx_1+b$ which can be interpreted as an equation in the parameter space *m,b* shown in Fig. 1(b). Doing the same transformation, all straight lines going through the point (x_2,y_2) can be described by the equation $y_2=mx_2+b$. The common point of both lines in the *m,b* parameter space is the line in the image space that contains both the points (x_1,y_1) and (x_2,y_2). This shows that all points on a straight line in the original image space are represented by lines which all cross at the same point in the parameter space. For practical implementation purposes, the parameter space must be quantized into a two-dimensional array of accumulator cells. For each point on the line in parameter space, the accumulator cell is incremented. Then all accumulator cells above a certain threshold will represent lines in the original image.

Generalization of this procedure to more complex curves and objects is shown in the algorithms in the following section.

III. Classical and Generalized Hough Transform Algorithms [11]

This section will present the algorithms for the classical Hough transform, which is used when an analytic equation of the object or curve to detect is known, and the generalized Hough transform, which is used when an analytic equation of the object is not known.

A. Classical Hough Transform

Let $\mathbf{a}$ be the vector that contains the parameters of the analytic equation, and $\mathbf{x}$ be the vector containing the parameters of the image. For example, if the analytic expression $f(\mathbf{x},\mathbf{a})$ is a line with the equation,

$$s = x_1\cos\theta + x_2\cos\theta \tag{1}$$

then $\mathbf{a} = [s \ \ \theta]$ and $\mathbf{x} = [x_1 \ \ x_2]$. Or, if the analytic equation $f(\mathbf{x},\mathbf{a})$ is a circle with the equation

$$(x_1-a)^2 + (x_2-b)^2 = r^2 \tag{2}$$

then $\mathbf{a} = [a \ \ b \ \ r]$.

First, quantize the parameter space within the limits of the parameters $\mathbf{a}$. The dimensionality *n* of the parameter space is given by the number of parameters of the vector $\mathbf{a}$.

Second, form an n-dimensional accumulator array $A(\mathbf{a})$ that has the same structure of the quantized parameter space. Initialize all structure values to zero.

Third, for each point in the image, increase all accumulator cells $A(\mathbf{a})$ if $f(\mathbf{x},\mathbf{a}) = 0$

$$A(\mathbf{a}) = A(\mathbf{a}) + \Delta A \tag{3}$$

Finally, the local maxima in the accumulator array $A(\mathbf{a})$ correspond to realizations of curves $f(\mathbf{x},\mathbf{a})$ that are present in the original image.

B. Generalized Hough Transform

Let an R-table be a look-up table that defines the relationship between the boundary positions and orientations and the Hough parameters. To construct an R-table, specify an arbitrary reference point (x_{1ref}, x_{2ref}) within the feature, with respect to which the shape (the distance r and angle α of normal lines drawn from the boundary to this reference point) of the feature is defined. (See Fig. 2). The look-up table will consist of these distance and direction pairs, indexed by the orientation ϕ of the boundary. An example R-table is given in Table I. The Hough transform space is now defined in terms of the possible positions of the shape in the image, i.e. the possible ranges of (x_{1ref}, x_{2ref}). The transformation is defined by:

$$x_{1ref} = x_1 + r(\phi)\cos(\alpha(\phi)) \tag{4}$$

$$x_{2ref} = x_2 + r(\phi)\cos(\alpha(\phi)) \tag{5}$$

If the orientation ϕ is unknown, and/or the size S of the object is unknown, the number of parameters increases to four.

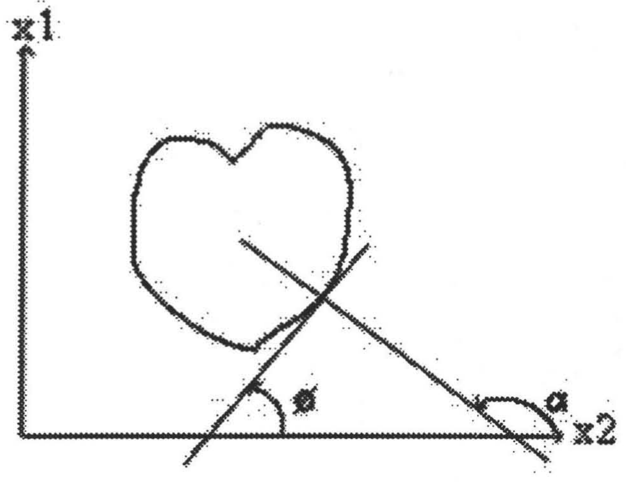

Fig. 2. Geometry used to form the R-table

TABLE I
EXAMPLE R-TABLE

Orientation angle	Set of intersection parameters
ϕ_1	$(r_1^1, \alpha_1^1), (r_1^2, \alpha_1^2), \ldots (r_1^{n1}, \alpha_1^{n1})$
ϕ_2	$(r_2^1, \alpha_2^1), (r_2^2, \alpha_2^2), \ldots (r_2^{n2}, \alpha_2^{n2})$
...	...
ϕ_k	$(r_k^1, \alpha_k^1), (r_k^2, \alpha_k^2), \ldots (r_k^{nk}, \alpha_k^{nk})$

First, construct an R-table description of the desired object.

Second, form a data structure $A(x_1, x_2, S, \phi)$ that represents the potential reference points. Initialize all structure values to zero.

Third, for each pixel (x_1, x_2) in the original image, determine the edge direction ϕ. Find all potential reference points (x_{1ref}, x_{2ref}) and increase all $A(x_{1ref}, x_{2ref}, S, \phi)$,

$$A(x_{1ref}, x_{2ref}, S, \phi) = A(x_{1ref}, x_{2ref}, S, \phi) + \Delta A \quad (6)$$

for all possible values of rotation and size change,

$$x_{1ref} = x_1 + r(\phi)S\cos(\alpha(\phi)+\phi) \quad (7)$$

$$x_{2ref} = x_2 + r(\phi)S\cos(\alpha(\phi)+\phi) \quad (8)$$

Finally, the local maxima in the accumulator data structure A correspond to realizations of the objects that are present in the original image.

IV. SIMULATION RESULTS

I implemented the classical Hough transform to detect a line in an image with additional features (see Fig. 3), four circles in a noisy image with missing parts (see Fig. 4), and a circle (of different radius than the four circles) in an image with features (see Fig. 5). In each case, I used the Canny edge detector to get the edges of the image, and then I took the Hough transform of that edge image.

As shown in Fig. 3, there are many edges that don't belong to the lines, but using the Hough transform allowed me to find just the long lines. I could have decreased the threshold when I was transforming the Hough image to the detected lines image if I wanted to see a few more lines from the letters, but I chose not to.

When I detected the circles, I found the radius before I took the Hough transform because this allowed for a two-dimensional Hough image rather than a three-dimensional Hough image, which greatly decreases the number of computations and storage. As shown in Fig. 4, the Hough transform detected four circles in the presence of noise and gaps in the circle boundaries. As shown in Fig. 5, the Hough transform detected a circle of larger radius than Fig. 4 from an image with additional features.

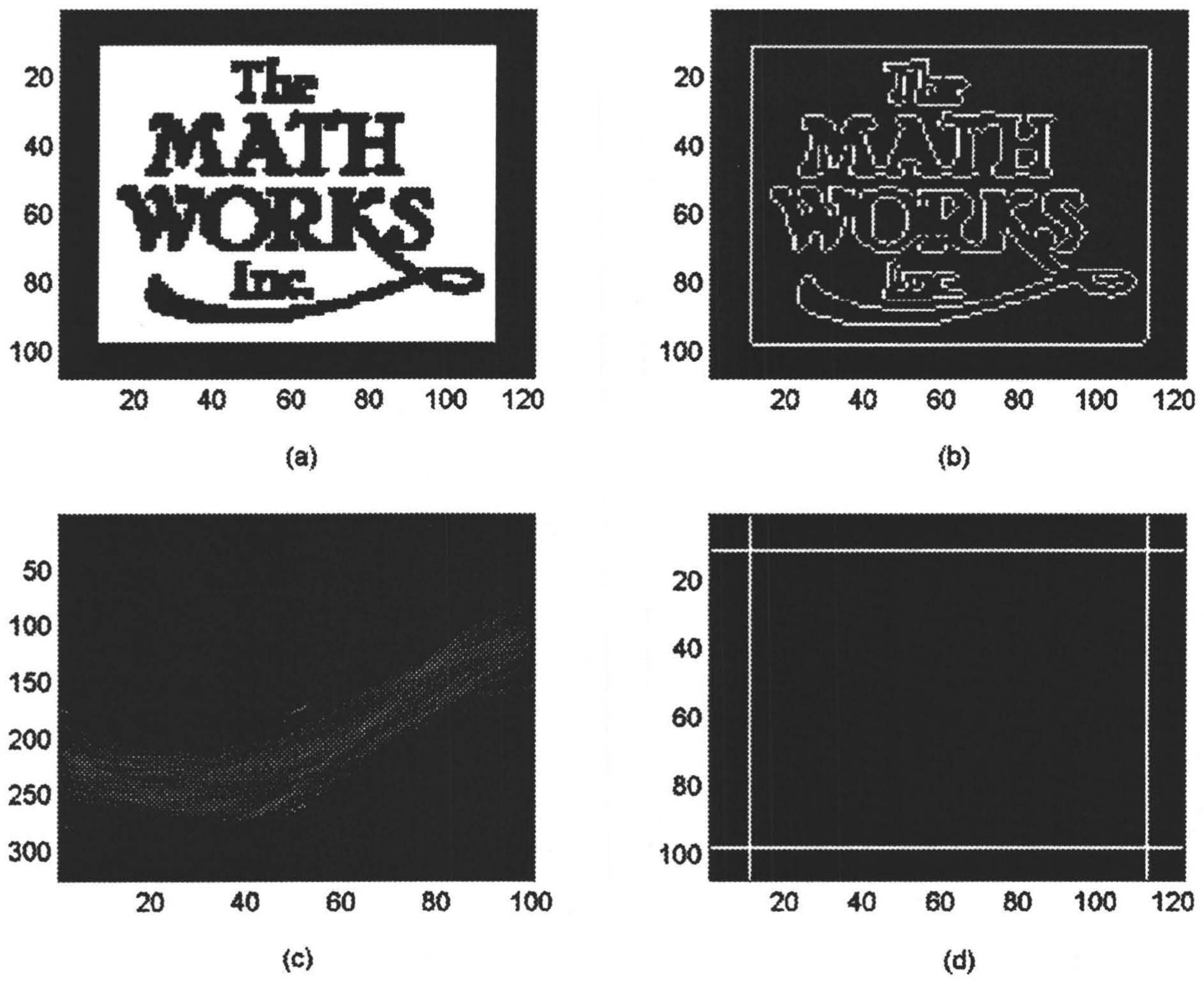

Fig. 3. Line detection using the Hough Transform (a) original image, (b) edges using the Canny edge detector, (c) Hough image, (d) detected lines

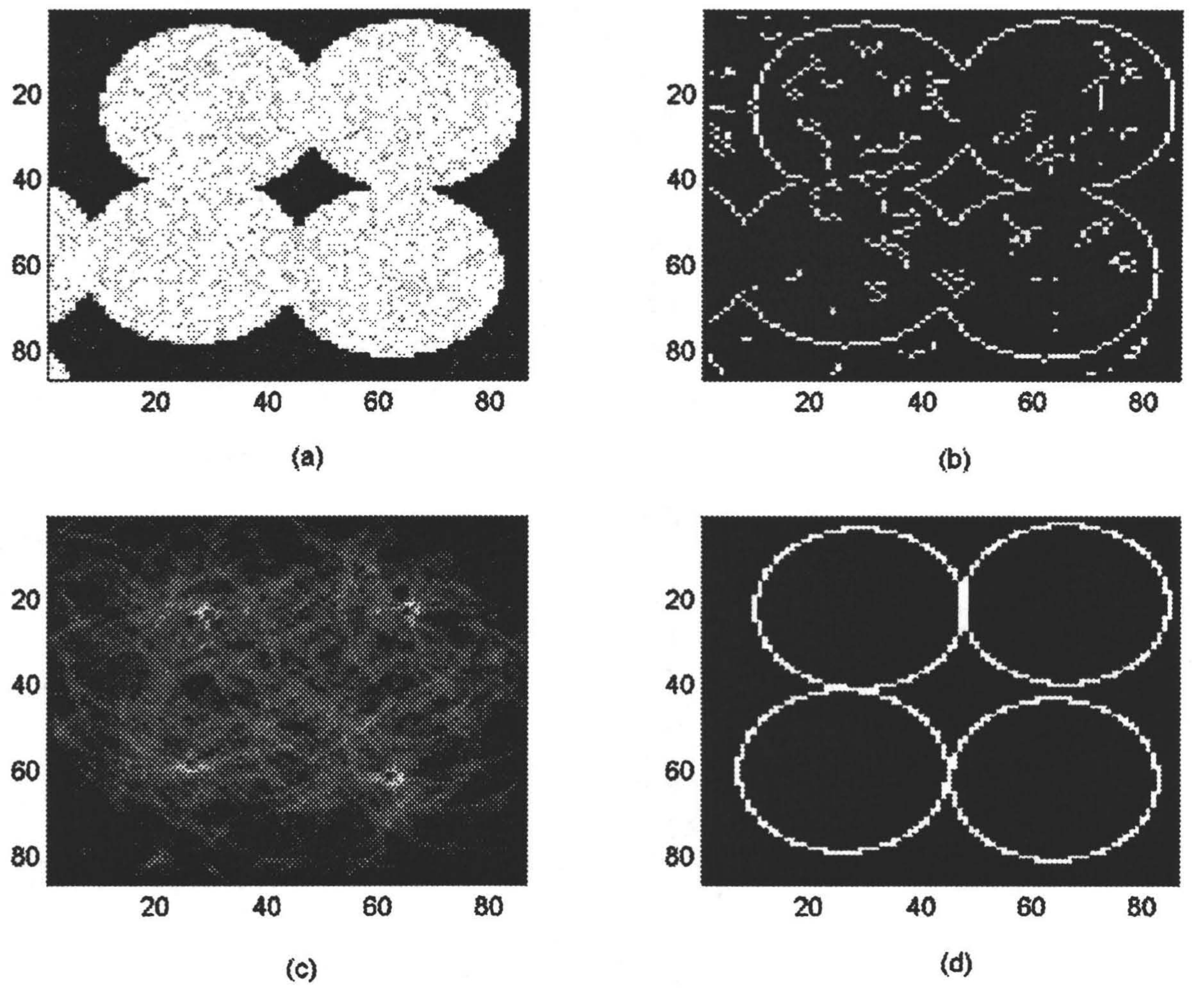

Fig. 4. Circle detection using the Hough Transform on image with multiple degraded circles which have a radius of 18.75 pixels (a) original image, (b) edges using the Canny edge detector, (c) Hough image, (d) detected circles

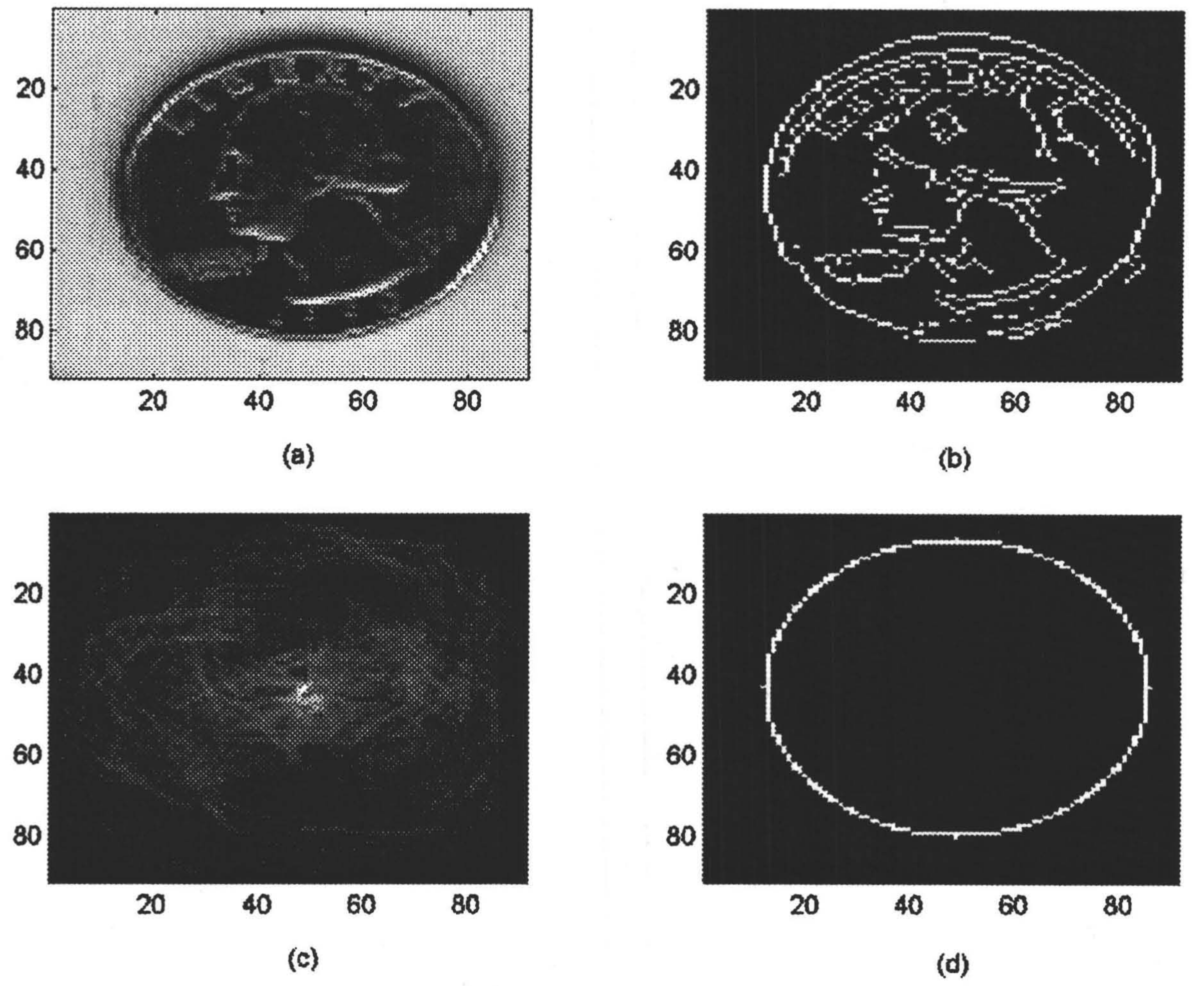

Fig. 5. Circle detection using the Hough Transform on image with a radius of 36.5 pixels (a) original image, (b) edges using the Canny edge detector, (c) Hough image, (d) detected circles

V. Conclusions

I have presented some of the applications and uses of the Hough transform, two algorithms describing its process, and simulation examples showing its detection of lines and circles. Through this research and implementation process, I found that the Hough transform has many advantages including its ability to detect objects in noisy images, its ability to find objects that may have deformations or missing parts, its ability to detect objects of a particular shape but not necessarily the same size or orientation, and its ability to find objects in the presence of overlapping or additional structures in the image.

Unfortunately, the algorithms that I have described and implemented in this paper require a lot of storage and computation unless the object can be represented as a curve with few parameters or if prior knowledge of border direction, size, and/or shape is used to limit the number of accumulator parameters. Because of these time and space requirements, effort has been devoted to hierarchical approaches; fast algorithms were developed; gray-scale Hough transforms working directly in image data were presented; methods combining the Hough transform and automated line tracing were studied; and many parallel implementations were tested [11].

REFERENCES

[1] N. C. Fitton and S. J. D. Cox, "Optimising the application of the Hough transform for automatic feature extraction from geoscientific images," *Computers and Geosciences*, vol. 24, pp. 933-51, Dec. 1998.

[2] C. C. Lai and W. H. Tsai, "Estimation of moving vehicle locations using wheel shape information in single 2-D lateral vehicle images by 3-D computer vision techniques," *Robotics and Computer Integrated Maufacturing*, vol. 15, pp. 111-20, Apr. 1999.

[3] S. Barbarossa and A. Scaglione, "Adaptive time-varying cancellation of wideband interferences in spread-spectrum communications based on time-frequency distributions," *IEEE Transactions on Signal Processing*, vol. 47, pp. 957-65, Apr. 1999.

[4] K. Glossop and P. J. G. Lisboa and P. C. Russel, "An implementation of the Hough transformation for the identification and labeling of fixed period sinusoidal curves," *Computer Vision and Image Understanding*, vol. 74, pp. 96-100, Apr. 1999.

[5] B. K. Jeon and J. H. Jang and K. S. Hong, "Map-based road detection in spaceborne synthetic aperture radar images based on curvilinear structure extraction," *Optical Engineering*, vol. 39, pp. 2413-21, Sept. 2000.

[6] E. J. Bernabeu and J. Tornero, "Optimal geometric modeler for robot motion planning," *Journal of Robotic Systems*, vol. 17, pp. 595-608, Nov. 2000.

[7] J. F. Koretz and C. A. Cook and P. L. Kaufman, "Aging of the human lens: changes in lens shape at zero-diopter accommodation," *Journal of the Optical Society of America A, Optics, Image Science, and Vision*, vol. 18, pp. 265-72, Feb. 2001.

[8] P. K. Ser and C. S. T. Choy and W. C. Siu, "Genetic algorithm for the extraction of nonanalytic objects from multiple dimensional parameter space," *Computer Vision and Image Understanding*, vol. 73, pp. 1-13, Jan. 1999.

[9] C. F. Olson, "Constrained Hough transforms for curve detection," *Computer Vision and Image Understanding*, vol. 73, pp. 329-45, Mar. 1999.

[10] J. Matas and C. Galambos and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Computer Vision and Image Understanding*, vol. 78, pp. 119-37, Apr. 2000.

[11] M. Sonka and V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Pacific Grove, CA: Brooks/Cole Publishing Compane, 1999.

APPENDIX

A. Matlab code for line detection

```
%Final Project Line Detection
clear, close all

I = imread('ic.tif');
I = im2double(I);
I=I(50:150,50:150);
subplot(2,2,1)
colormap('gray')
imagesc(I) %clean image

E = edge(I,'sobel');
subplot(2,2,2)
imagesc(E)

[Y,X] = size(E);
maxHI = ceil(.70710678*(Y+X));
minHI = -maxHI;
N = 100;
HI = zeros(2*maxHI+1,N);
rad = linspace(0,pi,N+1);
for j=1:Y,
   for i=1:X,
      if E(j,i)==1,
         for k=1:N,
            r = round(i*cos(rad(k))+j*sin(rad(k)));
            newr = r+maxHI+1;
            HI(newr,k)=HI(newr,k)+1;
         end
      end
   end
end

subplot(2,2,3)
imagesc(HI)
thresh = 25;
L = zeros(Y,X);
for newr=1:2*maxHI+1,
   for k=1:N,
      if HI(newr,k)>thresh,
         for j=1:Y,
            for i=1:X,
               r=newr-maxHI-1;
               if r==round(i*cos(rad(k))+j*sin(rad(k))),
                  L(j,i)=1;
               end
            end
         end
      end
   end
end

subplot(2,2,4)
imagesc(L)
```

B. Matlab code for circle detection:

```
%Final Project Circle Detection
clear, close all

I = imread('eight.tif');
I = im2double(I);
I=I(100:180,20:100);
subplot(2,2,1)
colormap('gray')
imagesc(I) %clean image

E = edge(I,'canny');
subplot(2,2,2)
imagesc(E)

r=36.5;

[Y,X] = size(E);
R=r;
maxHIx = X;
maxHIy = Y;

HI = zeros(maxHIy,maxHIx);
for j=1:Y,
   for i=1:X,
      if E(j,i)==1,
         for a=1:maxHIx,
            for b=1:maxHIy,
               if round((i-a)^2+(j-b)^2)>=floor((R-.5)^2),
                  if round((i-a)^2+(j-b)^2)<=ceil((R+.5)^2),
                     HI(b,a)=HI(b,a)+1;
                  end
               end
            end
         end
      end
   end
end

subplot(2,2,3)
imagesc(HI)

HIthresh=100;
C=zeros(Y,X);
for b=1:Y,
   for a=1:X,
      if HI(b,a)>=HIthresh & HI(b,a)==max(max(HI(b-2:b+2,a-2:a+2))),
         for j=floor(b-R):ceil(b+R),
            for i=floor(a-R):ceil(a+R),
               if round((i-a)^2+(j-b)^2)>=floor((R-.5)^2),
                  if round((i-a)^2+(j-b)^2)<=ceil((R+.5)^2),
                     C(j,i)=1;
                  end
               end
            end
         end
      end
   end
end

subplot(2,2,4)
imagesc(C)
```