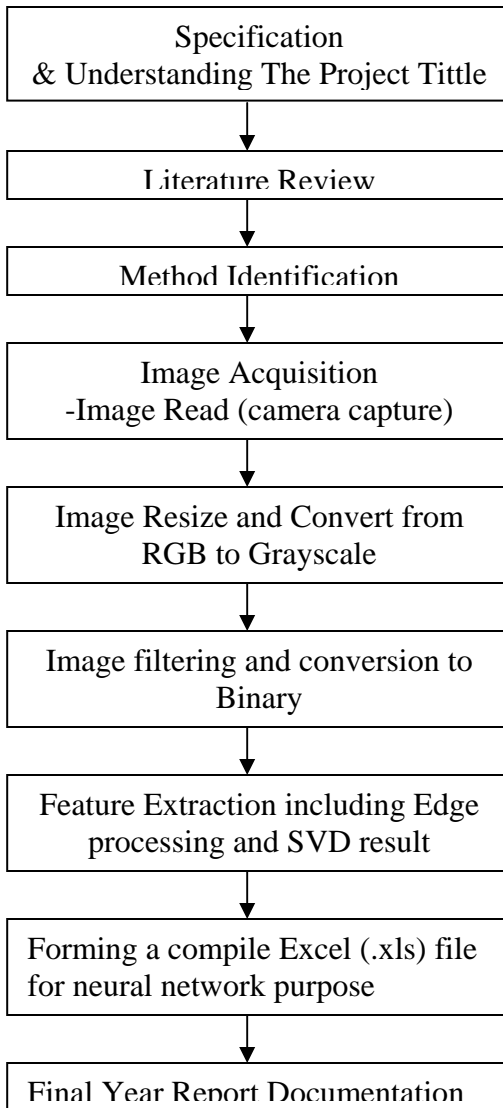


## **CHAPTER 3**

### **METHODOLOGY**

#### 3.1 Flow Chart of Project

Figure 3.1 explains thoroughly on how the image processing is obtained. The title which is “Shape recognition Using Image Processing” means that mainly all the system in this project is done using the image processing. It is recommend to use MATLAB program to process all the obtained images because the MATLAB software are very good in performing the evaluation of image processing and directly connected to the neural networks division.



**Figure 3.1** Flow Chart

### 3.2 Commands

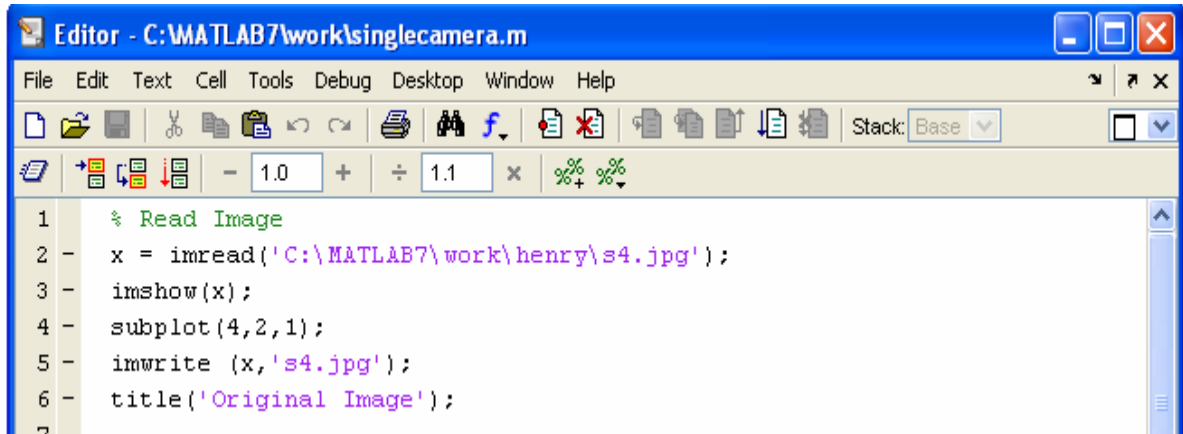
There are several commands that had been used inside the program for the image processing. All the command however is a straight output where firstly, it will read the important file from the current directory (default is work folder in MATLAB program) and from there it will be process using image show. The output is the same image only being converted into various forms. Below is the command that been used to make the image acquisition and image processing program.

### 3.2.1 Image Acquisition

The image acquisition involves capturing and storing digital images from vision sensors like Color Digital cameras, Monochrome and Color CCD Cameras, IR Cameras, Video cameras, Scanners etc. Images at different illumination levels or lighting conditions are acquired to analyze the effect of lighting on image processing.

#### 3.2.1.1 Image Acquisition ('imread')

'imread' is the command to call out the .jpg images from the image storing folder. MATLAB supports several graphics file formats, such as HDF and TIFF that can contain multiple images. By default, imread imports only the first image from a file. To import additional images from the file and use the syntax supported by the file format, which in this case is the .jpg file.

A screenshot of the MATLAB Editor window. The title bar reads "Editor - C:\MATLAB7\work\singlecamera.m". The menu bar includes "File", "Edit", "Text", "Cell", "Tools", "Debug", "Desktop", "Window", and "Help". The toolbar contains various icons for file operations and editing. Below the toolbar, there is a zoom control showing "1.0" and "1.1". The main editor area contains the following MATLAB code:

```
1 % Read Image
2 - x = imread('C:\MATLAB7\work\henry\s4.jpg');
3 - imshow(x);
4 - subplot(4,2,1);
5 - imwrite(x,'s4.jpg');
6 - title('Original Image');
```

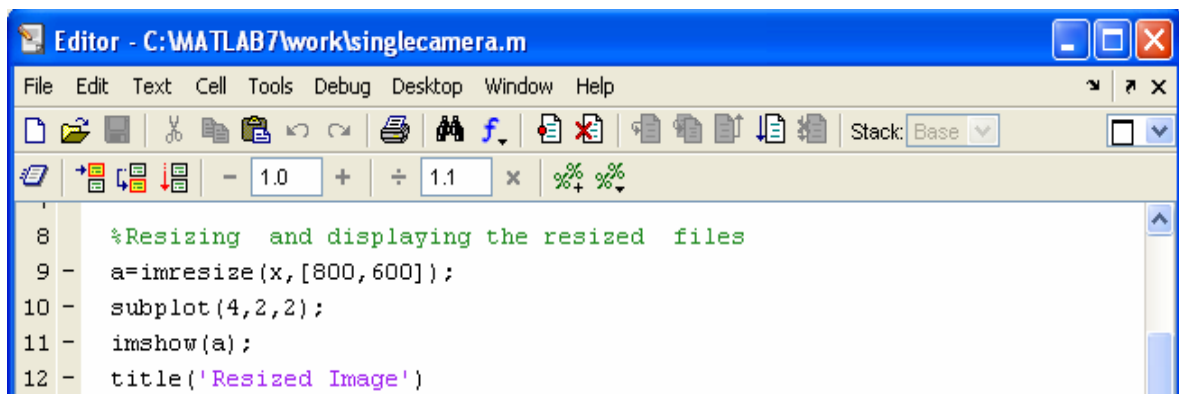
**Figure 3.2** Read Image command

Figure 3.2 is the part where the read command image from the folder and placed all the images in different folder depending on the model that have been selected. There are 4

models that were taken to make the database for the system. There are cylinder, rectangle, cubic and pyramid.

The images have it own distance. There is one single measurement of distance during image acquisition. For each distance, there are actual positions of the camera to take as to make the system robust. The camera that used is storing all the images in .jpg format. The command 'imwrite' is use as to change the image acquisition from the previous image taken to the new one without any stacking images. After that the image of the original that were taken by the cameras is displayed with a name of Original 1 if captured by the single camera.

### 3.2.1.2 Resizing Image ('imresize')



```
Editor - C:\MATLAB7\work\singlecamera.m
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + 1.1 x % %
8 %Resizing and displaying the resized files
9 - a=imresize(x,[800,600]);
10 - subplot(4,2,2);
11 - imshow(a);
12 - title('Resized Image')
```

**Figure 3.3** Resizing Command

Figure 3.3 is the command that been used to resize the images from original. The purpose of resizing is to reduce the pixel values so that the processing will be faster and the Singular Value Decomposition (SVD) value will be limited. Resizing to a proper resolution is important because over-resized will make the image broke. Less resizing however will make the SVD data a lot of values and here the database will be too much. Hence, the

system will become slower. The unknown 'a' refers to the left original image and 'b' refers to the right original image. Also pick the resolution of 80x60 because of the sum of pixels are small. Therefore, with a small pixel value, the SVD of the system can be taken less than expected to make the database without losing any identification of the image.

### 3.2.2 Image Storing

There are various format stored for the images such as .jpeg, .tiff, .png, .gif and .bmp. The captured images are stored in one of the following formats on the computer for possible image processing:

#### 3.2.2.1 Joint Photographic Experts Group (JPEG)

Jpeg format is designed to transfer graphic data and images via digital telecommunication networking and is generally used to hold and transfer full color photorealistic images. Before Jpeg, there were very few formats, which supported 24 bit halftone images. TIFF and BMP formats allowed holding 24 bit data, but they failed to perform a lossless compression of the data, which contained thousand colors from the real world, on the high quality level. Jpeg compresses photos though with quality loss. Photos and multi-color images, transferred in this format, are ideal for networking. Jpeg format is primarily used to hold photorealistic graphic images with a large number of colors.

### 3.2.2.2 Graphics Interchange Format (GIF)

Gif file format is used to hold and transfer images in index color mode (not more than 256). The format also supports a lossless LZW compression algorithm (a single file may contain a set of images) and interlaced mode (interfaced image load). The format is one of the most popular graphic formats. Though it not suited to hold a photorealistic images, for it cannot contain more than 256 colors. It is mostly used to display animation and drawing pictures without blend. Gif format file has got a set of compression algorithm features that allows compressing images with sequential horizontal colors. That means that a file, which contains horizontal bars, will take fewer places than the same file with vertical bars. Gif format, unlike Jpeg format, use a lossless compression algorithm. That means that while compression/decompression bits are not deleted. Gif uses a patent compression /decompression algorithm, which is called LZW (Lempel Zev Welch). Interlaced mode makes possible to refine images, as they are loaded. It is very convenient for big files, as user can see the main image features in several seconds after loading.

### 3.2.2.3 Microsoft Windows Bitmap (Bitmap)

Bitmap is a home Windows raster format, which is used practically for all possible raster data storage. All BMP versions were designed for computers with Intel processors. The current format version is device in dependable (that means that Bitmap determines the pixel's color without reference to display device) and makes possible to record images of a different quality level (to the point of 32 bits). After revising, the format was used to hold color and black and white images, so it became general. The main advantage of the format is considered to be its usability and wide software support. Compression algorithm, used in BMP format, is one of the RLE run-length encoding variants (read more about RLE compression algorithm here). But most files are hold incompact.

#### 3.2.2.4 Tagged Image File Format. (TIFF)

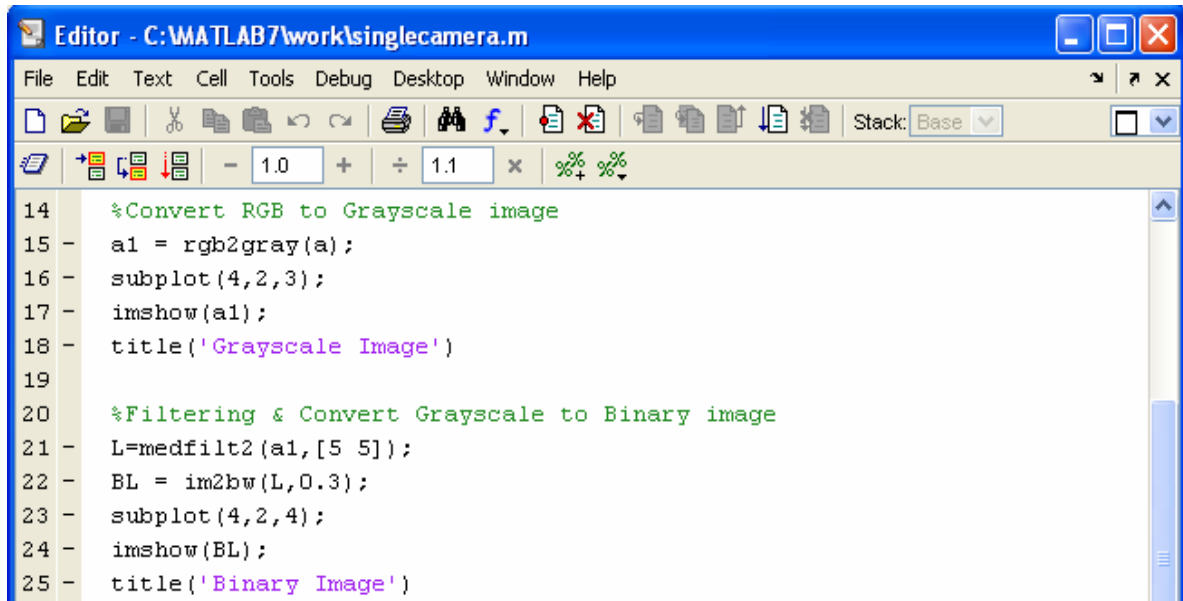
The format was released in 1986 and was generally used to hold scanned white and black images (Tiff 3.0). Released in 1987, Tiff 4.0 got prevalence for it supported incompact color RGB images. Tiff 5.0 formats, released in 1988, made possible to hold palette color images and support LZW compression algorithm (read more about LZW compression algorithm here). The current Tiff (Tiff 6.0) is a rich and flexible file format, supported by many programs. The format is capable to record halftone images with different pixel intensity. So, it is considered to be a perfect format for graphic data storage and processing. Tiff format is one of the most multipurpose and integrated raster formats. It is easily ported among platforms and used for different purposes (besides Internet).

#### 3.2.2.5 Portable Network Graphic (PNG)

.png file format is a comparatively new progressive format originally designed to replace dated Gif. .png format has got a set of new features that Gif lacks. .png also performs a lossless compression with the help of Deflate algorithm (read more about Deflate algorithm here) and supports interlaced mode. The format handles 256 transparency levels (that means that images might be partially transparent). The fact, that the format supports color depth up to 48 bits and performs a lossless compression, makes possible to hold photorealistic images. They won't lose its quality while compressing and decompressing. .png format is used specially for networking. Resolution of the captured images varies as per default camera settings. The resolution or size of the images can be varied depending on the image processing application.

From the entire format, the camera that used is in jpg format. The system will be processed in .jpg format for all the image acquisition. However, the images that used in acquiring database suppose to be perfect where no unequal lighting, additional shadow or noise that will spoiled the processing.

### 3.3 Conversion Image ('rgb2gray'; 'im2bw')

A screenshot of the MATLAB Editor window titled 'Editor - C:\MATLAB7\work\singlecamera.m'. The window has a menu bar (File, Edit, Text, Cell, Tools, Debug, Desktop, Window, Help) and a toolbar with various icons. The code editor shows the following MATLAB code:

```
14 %Convert RGB to Grayscale image
15 - a1 = rgb2gray(a);
16 - subplot(4,2,3);
17 - imshow(a1);
18 - title('Grayscale Image')
19
20 %Filtering & Convert Grayscale to Binary image
21 - L=medfilt2(a1,[5 5]);
22 - BL = im2bw(L,0.3);
23 - subplot(4,2,4);
24 - imshow(BL);
25 - title('Binary Image')
```

**Figure 3.4** Conversion Command

The conversion command is set to convert all the colored images into black and white images also known as grayscale image as shown in Figure 3.4.

#### 3.3.1 Indexed Image

This is an image whose pixel values are direct indices into an RGB colormap. In MATLAB, an indexed image is represented by an array of class uint8, uint16, or double. The colormap matrices is always an m by 3 array of class double and often use the variable name X to represent an indexed image in memory, and map to represent the colormap.



### 3.3.2: Image Intensity

This is an image consisting of intensity (grayscale) values. In MATLAB, intensity images are represented by an array of class uint8, uint16, or double. While intensity images are not stored with color maps, MATLAB uses a system color map to display them. Often use the variable name to represent an intensity image in memory. This term is synonymous with the term "*grayscale*".

RGB Images is an image in which each pixel is specified by three values – one each for the red, blue, and green components of the pixel's color. In MATLAB, often use the variable name RGB to represent an RGB image in memory. Image conversions involve converting from one to another.

The procedure here is starts from conversion of the RGB image to grayscale by the command `rgb2gray` follow by the representative unknown. Then the image is filtered using median filter (discussed on next topic) for the camera and then the clean image is then convert to binary using the command `im2bw` with the threshold value of 0.3. The image is shown on the result and discussion chapter and further information about filtering is discussed in the next chapter.

### 3.4 Filtering Binary Image

There are 3 types of filter's command; Adaptive filter, Gaussian filter, Linear filter and Median Filter.

### 3.4.1 Linear filtering

Filtering is a technique for modifying or enhancing an image to emphasize certain features or remove other features.

Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel.

Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

### 3.4.2 Filtering Using *imfilter*

Filtering of images, either by correlation or convolution can be performed using the toolbox function '*imfilter*'. Such a filter is often called an averaging filter. The output image has the same data type, or numeric class, as the input image. The *imfilter* function computes the value of each output pixel using double-precision, floating-point arithmetic. If the result exceeds the range of the data type, the *imfilter* function truncates the result to that data type's allowed range. If it is an integer data type, *imfilter* rounds fractional values.

### 3.4.3 Median Filters

Median filtering is similar to using an averaging filter, in that each output pixel is set to an "average" of the pixel values in the neighborhood of the corresponding input pixel.

However, with median filtering, the value of an output pixel is determined by the

median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called outliers). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image. The size of the neighborhood used for filtering is 3-by-3. Median filtering is a specific case of order-statistic filtering, also known as rank filtering.

#### 3.4.4 Gaussian Filters

`h = fspecial ('Gaussian')` creates a two-dimensional Gaussian low pass filter, `h`, of the specified type. `fspecial` returns `h` as a correlation kernel, which is the appropriate form to use with `imfilter`.

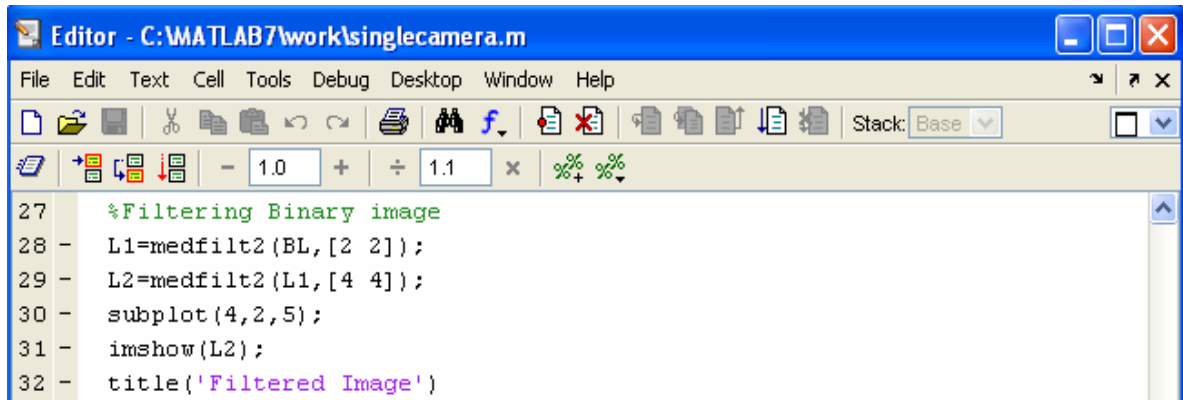
`h = fspecial ('gaussian', hsize, sigma)` returns a rotationally symmetric Gaussian low pass filter of size `hsize` with standard deviation `sigma` (positive). `hsize` can be a vector specifying the number of rows and columns in `h`, or it can be a scalar, in which case `h` is a square matrix. The default value for `hsize` is `[3 3]`; the default value for `sigma` is `0.5`.

#### 3.4.5 Adaptive Filtering

The `wiener2` function applies a Wiener filter (a type of linear filter) to an image adaptively, tailoring itself to the local image variance. Where the variance is large, `wiener2` performs little smoothing. Where the variance is small, `wiener2` performs more smoothing.

This approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high frequency parts of an image. In addition, there are no design tasks; the `wiener2` function handles all preliminary computations, and implements the filter for an input image. `wiener2`, however,

does require more computation time than linear filtering. Wiener2 works best when the noise is constant-power ("white") additive noise, such as Gaussian noise.



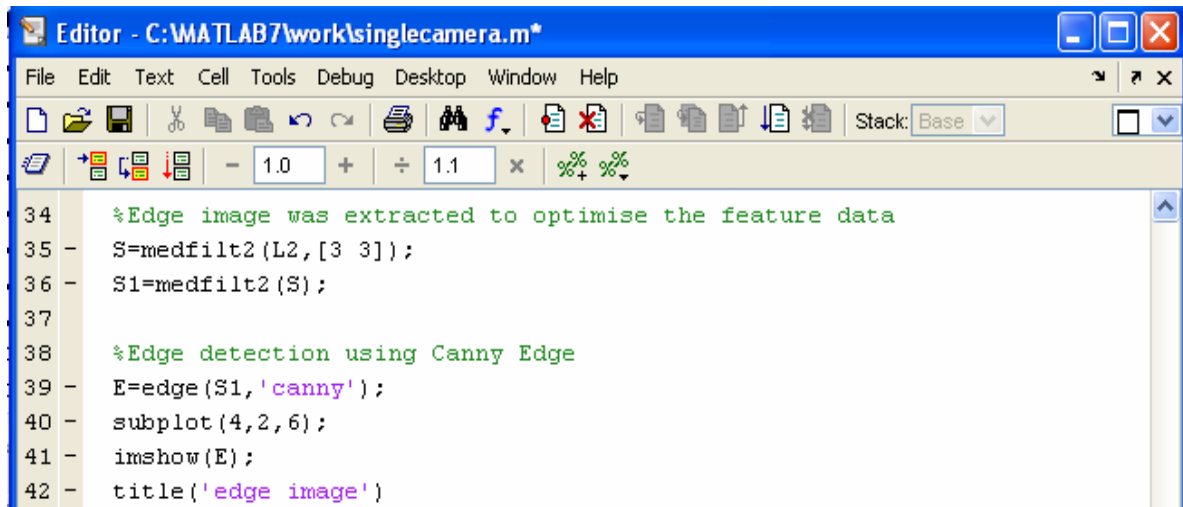
```
Editor - C:\MATLAB7\work\singlecamera.m
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
- 1.0 + ÷ 1.1 x %± %±
27 %Filtering Binary image
28 - L1=medfilt2(BL,[2 2]);
29 - L2=medfilt2(L1,[4 4]);
30 - subplot(4,2,5);
31 - imshow(L2);
32 - title('Filtered Image')
```

**Figure 3.5** Filtering Binary Images

The median filtering in Figure 3.5 is to remove type of noise consists of random pixels' being set to black or white (the extremes of the data range). In this case, the size of the neighborhood used for filtering is 2-by-2 and 4-by-4. The result in the next chapter that medfilt2 does a better job of removing noise, with less blurring of edges.

### 3.5 Edging

Figure 3.6 is where the image that already had been merged is being enhanced by using edge as detail information before acquiring the Singular Value Decomposition sum of numbers. In the program, been used median filtering first before applying the Canny edge detection.

The image shows a screenshot of a MATLAB editor window titled "Editor - C:\MATLAB7\work\singlecamera.m\*". The window has a standard menu bar (File, Edit, Text, Cell, Tools, Debug, Desktop, Window, Help) and a toolbar with various icons. Below the toolbar is a command window area with a stack set to "Base". The main area contains MATLAB code for edge processing. The code includes comments and function calls for median filtering and Canny edge detection. The code is as follows:

```
34 %Edge image was extracted to optimise the feature data
35 - S=medfilt2(L2,[3 3]);
36 - S1=medfilt2(S);
37
38 %Edge detection using Canny Edge
39 - E=edge(S1,'canny');
40 - subplot(4,2,6);
41 - imshow(E);
42 - title('edge image')
```

**Figure 3.6** Edge processing program

After calling out the added image, in this case the unknown is A, as the use of median filtering and set the pixel to 3-by-3 format and from here the image is filtered using median filter.

The image is then converts to edge image to get a line image where the image is still using black and white image. However here the image is in binary form, therefore, it can be called an one and zero images. The edge detector here is the Canny image instead of Sobel, Robert or Prewitt.

### 3.6 Feature Extraction

The feature here is focus on Singular Value Decomposition (SVD). This is a feature that can detect distance in form of a numbering for every pixel group. The programs are as shown in Figure 3.7:

```
43  
44 %Feature extraction through singular value decomposition  
45 - F=svd(double(E))  
46 - fout=fopen('input.dat','a');  
47  
48 - for j= 1:1:40  
49 -     fprintf(fout, '%f\t',F(j));  
50 - end  
51 -     fprintf(fout, '\n');  
52 - fclose(fout);  
53 - fprintf('Execution Over');
```

**Figure 3.7** Feature Extraction Program

The code to call out the feature extraction is SVD (double (unknown)). Put 'F' as the unknown for the previous data occupied from edge detection. The algorithms use a sum of eigenvalues to find the largest magnitude eigenvalues and corresponding eigenvectors of the pixels. This means that for the resolution 80x60, can get the biggest values about 6 to 10 with decimal numbers. The j there refers to SVD's (A, k, 40) which means A=1, k=1 and 40 uses eigs to find the 2(1) smallest magnitude eigenvalues and corresponding eigenvectors of  $j = [40 \ A; \ A' \ 40]$ , and then selects the k positive eigenvalues and their eigenvectors.

This feature extraction is good to extract the data that need to identify the image that been taken. This is because; this type of extraction can select all the number and put it in one file, which is the Excel file so that the file can execute the system thoroughly. This is where the execution for data acquisition and data processing ended. The neural network method is using to check the similarity of the input and the output of image file.