CHAPTER 2 : LITERATURE REVIEW

2.0 INTRODUCTION OF OBJECT DETECTION AN IMAGE

The object is always refer to the entity that we saw in image that sometimes are not recognize, this is because of the several factor that influence to the image, which is cause by their image itself, hardware or resolution are not supportable. Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings or cars) in digital images and videos. [20] Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.[15]

The challenges that can be face in these techniques will be the amount of variation in visual appearance. An object detector must cope with both the variation within the object category and with the diversity of visual imagery that exist in the world large.[20]

Techniques to identify the object detection could be the step in Sobel and Canny operator:

To recognize the object, we start from detect the edge-based segmentation, which is provide in Sobel and Canny detector.

- Sobel approach(basic understanding):[5]

  In theory, the operator consists of a pair of 3x3 convolution kernels as shown in figure 1. One kernel is simply the other rotated by 90 degree.
These Kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

\[ |G| = \sqrt{G_x^2 + G_y^2} \]

An approximate gradient magnitude is computed using:

\[ |G| = |G_x| + |G_y| \]

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

\[ \theta = \arctan(G_y/G_x) \]

In this case, orientation 0 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anti-clockwise from this.
We also can compute this gradient magnitude as faster as it. The two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 2.

![Figure 2](image)

**Figure 2** Pseudo-convolution kernels used to quickly compute approximate gradient magnitude

Using this kernel the approximate magnitude is given by:

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)|$$

- Canny approach (basic understanding): [4]

  First of all, we can make some adjustment to an image by using this operator to be an optimal edge detector. It takes as input a grayscale image, and produces as output an image showing the position tracked intensity discontinuities. The first step in Canny operator is start on the image which is the image itself will be smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high spatial derivatives. Edges give rise to the ridges in the gradient magnitude image.
Gaussian convolution is the Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump.

The Gaussian distribution in 1-D has the form:

\[
G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}
\]

where \(\sigma\) is the standard deviation(sigma) of the distribution. We have also assumed that the distribution has a mean of zero (i.e. it is centered on the line \(x=0\)).

In terms how to apply the canny detector, usually it started with the first step:

- First step:

  The first step in Canny detector algorithm is to filter out any noise in the original image before trying to locate and detect any edges. Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of a pixels at a time. The larger the width of the Gaussian mask, the lower is the detector’s sensitivity to noise. The localization error in the detected edges also increase slightly as the Gaussian width is increased.
• Second step

After smoothing the image and eliminate the noise, next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image for approximately the edge strength at each point that can be found. The Sobel operator uses a pair of 3x3 convolution mask, one estimating the gradient in the x-direction(columns) and the other estimating the gradient in the y-direction.

• Third step

Find the edge direction, which is to ensure the rotation in the x-direction and y-direction. If it is equal to zero, there will be some error.

• Fourth step

Once the direction obtain, the next step is to relate the edge direction to a direction that can be traced in an image. If a pixel in a 3x3 or 5x5 will be shown. Then, the possibility of the direction can be seen, there only have four possibility direction to the surrounding pixels. 0 degrees will show in the horizontal direction, 45 degrees along the positive diagonal, 90 degrees in the vertical direction or 135 degrees along the negative diagonal. Now the edge orientation must be resolve into one of these four directions depending on which direction it is closest to.

• Fifth step

After the edge direction are known, nonmaximum suppression now has to be applied. Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value.

• Sixth step

Hysterisis is used as means of eliminating streaking. Streaking is the breaking up an edge contour caused by the operator output fluctuating above and below the threshold.
2.1 **EXISTING DIGITAL IMAGE PROCESSING**

Currently nowadays, the image processing is often to be discussing in general, either in higher institution or in research implementation about image processing. Image processing become popular after people realize how their impact in daily life, especially when there are existing problems that lead to criminal abuse to occur. Their advantage through this technology increasing the power of tracing a particular image could be interpreted.

Digital image processing is use of computer algorithm to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analog image processing, it allows a much wider range of algorithm to be applied to the input data, and can avoid problems such as the buildup of noise and signal distortion during processing.[12]

Nowadays, we are in the middle of a second revolution by the rapid progress in video and computer technology. Personal computers and workstations have become powerful enough to process image data.

As a result, multimedia software and hardware is becoming standard for handling images, image sequences, and even 3-D visualization. The technology is now available to any scientist or engineer. In consequence, image processing has expanded and is further rapidly expanding from a few specialized applications into a standard scientific tool. Image processing techniques are now applied to virtually all the natural sciences and technical disciplines.[11]
2.2 TECHNIQUES IN COMPUTING IMAGE SEGMENTATION

In computer vision, segmentation seemly refer to the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically use to locate objects and boundaries (lines, curves, etc).[15] The result from the image segmentation will divided each set of the segments that are collectively cover the entire image, or a set of contours extracted from the image will be discuss as edge detection.[14]

Edge detection is often related to image processing, which is can be categorize into many meanings, such as region, boundary. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected.[14] Therefore, there might be techniques that can overcome these problems. Discontinuities are bridged if the distance between the two edges is within some predetermined threshold.

Thresholding is a non-linear operation that converts a grayscale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value. [13] The method will start from calculate value using an iterative method. Image histogram is a simple bimodal distribution. Adaptive thresholding will evaluates threshold based on the last 8 pixels in each row, using alternating rows. There are too many techniques that will be applied here depending on how we want to generate the output of the image itself.
2.3 CURRENT TECHNOLOGY ON OBJECT DETECTION

2.3.1 Edge Detection Using Neural Networks [7]

There is existence of several techniques to obtain an edge of object in an image by using edge based segmentation techniques. Many researches tend to use the technique to detect an edge commonly such as Sobel and Canny detector, these two methods will enhance the edge of an object, but there are certain factor that effect the algorithm between the both of these technique that make both are these technique are differences. There are various methodologies in term to identify this problem, the existence application is being made to resolve the problem in detect the edges, the following below can be as a guidelines for compute the algorithm:

The application of edge detection in neural network can also be as an application that is being made by other researcher in obtaining the edge algorithm. In neural network application it is explain about how this application can be a useful tool for edge detection. Neural network edge detector is a non linear filter, which is can have a built-in thresholding capability. Thus, the filtering, thresholding operation of edge detection is a natural application for neural network processing. Neural network type processing has the advantage of fault tolerance and speed when processed in highly parallel fashion, such as in VLSI or multi-processor digital hardware. It is an architecture that are particularly suited to edge detection because of the thresholding nonlinearity introduced by the bias weight and sigmoid function at the output node. The threshold operation required to create binary images at the output of a standard linear edge detection filter.
Neural network offer two approaches to resolve to the problem, which is from the neural network it can be learn or to be trained to recognize edges using a method such as backpropagation. The alternative way, it may skip the training process altogether and “engineer” the network by hand, incorporating some predetermined linear edge detection filter. The advantage from this method, it can create edge-detection filters based on the actual data one is expected to encounter in processing. So, this application could be implemented in an image processing program much as sliding kernel filter.

The approach start with considering a method for training a neural network to recognize an image object that is simple and work well for pattern recognition. As far as a neural network is concerned an edge is just an image object. This method can be used to train a network to recognize edges. The example of the implementation can be assume when we want to recognize specific object, we can say that the network will recognize this object if it returns +1 when given the object as an input, and -1 otherwise. To accomplish this training, the preparation of a dataset must be arranged in the following manner. In system pixel values range from -1 to +1, the input window is initially filled with a uniform value, usually -1 representing black on display at every pixel location. The output pattern for learning will be either +1 or -1, depend on the location of the object relative to the input window. If they are concentric the output pattern should be +1, otherwise it should be -1.
Figure 3. Slide Object Across Window

The Figure 3 show a dataset that being prepared for recognition training of a circular pattern. The first input pattern to the network will be a vector of length 9, representing a 3x3 all white image except for the upper left corner, which will contain a gray level due to presence of part of a circle at that pixel. To create the next training pattern the circle will be slid to the right one pixel. Now, the first two pixels of the second input training pattern will have some gray level due to the presence of the circle, while other pixels are white. This procedure of the circle slid over the entire detection window will continue. The network will has only one output that tells whether a pattern has been recognized(output=+1) or not (output=-1). When the circle is off-center as in figure 3 a, the output training pattern is -1. When the circle is on-center as in figure 3 b, it is +1.

Another approach, is engineering the network, which is directly code the weights for the desired function. Consider a simple network with the structure in figure 4. The input layer consists of five inputs. In the figure they are identified by which pixel from 3x3 input images is selected. In this configuration corner pixels are not used, and the central pixel is subtracted from surrounding pixels. This difference is then biased to the appropriate level with a bias weight that
is fed a constant input of +1. The thresholding is done at the output layer using a sigmoid function such as hyperbolic tangent.

Figure 4. Simple Neural Net Edge Detector

From the current application above, we can see there are too many techniques to identify the edges of a particular object. We can also have an algorithm to compute the pixels of an image, which is proven that in certain coding we can implement an algorithm to show that the pixels of the edges that we want to compute are achieve. It is can give benefit to image processing techniques in a way to create back a complete object and also to find the algorithm that can detect the edge of object that are not very clear to be seen.

2.4 CURRENT PROBLEM

Image processing could be categorized as a form of signal processing for which the input is an image, such as photographs or frames of video, the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image processing
techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.[11]

Image processing (picture processing) is the processing of the information contained in a digital image. Digital image is an image that consist of data which is specifically a set of elements) defined on an n-dimensional image regular grid that has the potential for display. [12] These elements are referred to as pixels. The pixels in different images may represent a variety of types of information, such as temperature, pressure, velocity, terrain height or tissue density.

The problems in image processing sometimes happen when the image itself is not clearly enough to be recognized. There is various type of image, such as gray scale image, color image, binary image and etc. A grayscale in digital image is an image in which the value of each pixel is a single sample, that is, carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are distinct from one-bit black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white also called binary images. Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum, for example infrared, visible light or ultraviolet, and in such cases they are monochromatic proper when only a given frequency is captured.[18]

For that purpose, in recognizing the object for their exact shape, are difficult to implement, because of the intensity of the pixel are not same, especially in color image, the presence of the light will cause to make the image of their pixel are not in balance. Therefore, the edges of the object will partially exist, and others are discontinuity.
2.4.1 Existing Application

Below is the example of the application of GUI interface using MATLAB software in developing the user interface:

![GUI Interface Example](image)

**Figure 5: The Design of the GUI Figure**

The figure above show the initial framework of the GUI interface, at first, we need to know what we want to have in the GUI figure (the final user interface), such as how many input/output images/plots, how many other kinds of control objects, and how to arrange it. Second, we need to come up the names for each control objects (including the axes to show images/plots). Usually, the names (tags) of the objects have two parts: the function of the object (such as “input image” and “close figure”), and the type of the object (such as “Edit box”, “Push Button”, and “Slider”). The red words that appear in that figure indicate the name of the objects we would like to have in the program.
Figure 6. After designed the GUI figure.

This figure show the designed GUI figure, MATLAB will automatically save it as whatever we want to name it such as GUIdemo.fig. MATLAB also will automatically generate GUIdemo.m which is associates with this GUI figure. In this figure it is showing that this is just an empty GUI program. Therefore, we need to add some code to make this program will be functional. The function for the coding part will be generated on m-file, which consist some predefined functions, such as:

```matlab
function varargout = GUIdemo(varargin)

This function is to set up the whole figure.

function GUIdemo_OpeningFcn(hObject, eventdata, handles, varargin)

This function is to initialize the GUI figure when it is opened at the first time.

image_file = get(handles.inEdit,'String');
```
This function will show default image in the axes “orglm” with the name in the Edit box “inEdit”.

The above function is an example of the coding that would be implemented in m-file, which is the function that will works synchronized with the button that provided in GUI figure.

Below is the example of coding that will implement in m-file:

```matlab
function appPush_Callback(hObject, eventdata, handles)
% hObject handle to appPush (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

flag = get(hObject,'Value');
image_file = get(hObject,'String');

if flag == 0
    % Displaying color image
    scale = get(handles.inEdit,'Value');
    image_file = get(handles.imageFile,'String');
    crm = str2double(crm);
    if Flag == 0
        disp('Display color image');
        im = crm*scale;
        im = im./255 * (im + (1-max)*255);
        set(hObject.newIm,'HandleVisibility','off');
        set(hObject.orgIm,'HandleVisibility','off');
        set(hObject.orglm,'HandleVisibility','off');
        set(hObject.newlm,'HandleVisibility','off');
        max(handles.newlm);  
        n = max(max(max(crm)));
        if n>255
            t = n;
            else t = 255;
        end
        image(im/n/t);
        max(handles.eqlm);
        max(handles.newlm);
```
Figure 7. After add some code for the opening function

Figure 7 above, show the image, after we add the function in MATLAB command line, we input "GUIdemo" or whatever name you give to this GUI program, the GUI program will show an image.

2.5 CONCLUSION

This chapter explain the current algorithm application that are use in detect the edge of the object in an image, there are variety technique to enhance the edge of the object nowadays, but seems that too difficult to trace the discontinuity edge of the object. Based on the techniques that are discussed before, each of them has their own advantage and disadvantage. The techniques that being used will be as guidelines to create the algorithm that can approximate the discontinuity of the edges of an object. The type of image also being discussed to make an understanding of image type that leads the problem to occur. This chapter will briefly described
the actual step in MATLAB environment, how to create a function and implemented the coding in m-file editor and also techniques on develop the interface in GUI.