International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 1 Issue 5, July– 2017, Pages: 62-85

Computer Vision Based Human Detection

Md. Ashikur Rahman
Dept. of Computer Science and Engineering
Shaikh Burhanuddin Post Graduate College
Under
National University, Dhaka, Bangladesh
ashikur@bracu.ac.bd

Abstract: *From still images human detection is challenging and important task for computer vision-based researchers. By detecting Human intelligence vehicles can control itself or can inform the driver using some alarming techniques. Human detection is one of the most important parts in image processing. A computer system is trained by various images and after making comparison with the input image and the database previously stored a machine can identify the human to be tested. This paper describes an approach to detect different shape of human using image processing. This thesis mainly based on shape based detection. Shape of the input image is extracted using a operator namely cany operator. Different images are used to train up the system. Then after training the system with the input image when a test image is provided to detect, test image is then compared with the database. If a certain threshold value is found then the test image is considered as the specific human. The average accuracy and precision rate achieved by the system is above 93%.*

Keyword: Computer Vision, Human Detection, Edge detection

1. INTRODUCTION

Analysis of visual scenes involving humans is one of the very popular, yet demanding applications of Computer Vision. Some of the tasks that fall under this domain are face recognition, gesture recognition and tracking the whole body. The motivation stems from the desire to improve human computer interaction which has been one of the general goals of artificial intelligence. Detecting humans in still images is a relatively new field. This domain is rich and challenging because of the need to segment rapidly changing scenes in natural environments. Additional momentum has been provided by the technological advances in the real time capture, transfer and processing of images. Today, the basic capability of a smart surveillance system would be to detect if humans are indeed present in the captured frames/images. This paper deals with detecting humans with fairly upright position in images. Human detection in images is a challenging task owing to variable appearance and wide range of poses that they can adopt. Hence, a robust feature set is needed that allows the human form to be discriminated clearly even in cluttered backgrounds and difficult illumination. Below we present some applications which will justify the need of a robust human detector using Canny Operator. In this thesis, an approach has been used to detect different human. To detect human, firstly the system is trained with various database. Shape based detection approach is used in this thesis.

1.2 What is Computer vision?

Computer vision techniques are used in such systems and human detection. Human detection and identification is an essential task for many applications such as Human-Robot-Interaction (HRI), video surveillance, human motion tracking, gesture recognition and human behavior analysis. Among many applications, we are interested in the field of HRI. As because intelligent robots should coexist with humans in a human-friendly environment, they must be aware of humans in their proximity, and identify them. Often, a single static camera is used for human detection due to its low cost and easy handling. However, a single camera is not practical for human detection by a mobile robot because the robot (camera) and the human are moving relatively each other, and the illumination conditions and backgrounds are changeable. Hence, the depth cues from a stereo camera can help to detect and identify humans effectively in mobile robot applications. Thus, stereo-based vision is used to detect and identify the humans in this paper.

1.3 Background of the Research

Human detection in real world scenes is a challenging problem. Intelligent vehicles refer to cars, trucks, buses etc. on which sensors and control systems have been proposed and assisted driving task. In recent years a variety of approaches have been proposed and impressive results have been reported on a variety of data bases.

1.4 Objective of the project

An intelligent machine must have the capability to detect various types of human. This thesis will be able to detect different types of object after training with various databases. The main human of this thesis is to detect specific object from still images. The major parts are:

1. Gray scale conversion.
2. Shape detection
3. Image comparisons
4. Human detection

1.5 Scope of the project

This paper presents a novel system for the real time detection and tracking from a moving vehicle. The scope of the work is versatile. Detecting human beings in images has gained prominence in the field of Computer Vision, with applications in the fields.

1.6 Proposed System's overview

This thesis presents a computer vision based human detection system using boundary based approach. By using camera this project firstly capture the real time images. After detecting moving object by segmentation method and eliminating noises this project detects the object. By comparing the boundary with predefined templates it can detect human. This approach used in this system has advantages over other human detection systems in its speed, simplicity, learning, capability and robustness to small changes in the images. This system can not only detect humans but also other object we want. This thesis is very efficient to train with different moving objects.

1.7 Project Organization

This thesis is organized into five chapters. This second chapter presents related works done previously with their success and limitation. Chapter three explains the details of the detection methodology used by us to tackle the problem of human detection, including experimental setup, feature vector generation algorithm, datasets used, evaluation methodology used and configuration settings for the feature descriptors tested. Chapter four explains the steps adopted towards choosing the best feature descriptor for compressed images. Finally in the last chapter explains the results obtained and concludes the paper by discussing its key contributions, limitations as well as future works.

1.8 Conclusion

In this paper, we use a combination of feature extraction and learning frame work to classify whether an image contains/does not contain human(s). We also propose a feature descriptor that is resistant to compression.

2. RELATED WORKS

2.1 Introduction

Human detection is closely related to general object recognition techniques. It involves two steps

- feature extraction and training a classifier as shown in Figure 2.

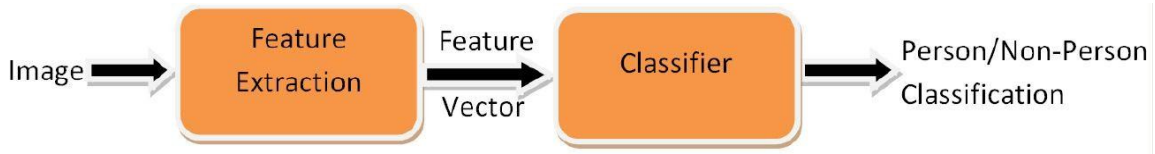


Figure 2.1 Components of Human Detection System

The image feature set that needs to be extracted should be the most relevant ones for object detection or classification, while providing invariance to changes in illumination, changes in viewpoint and shifts in object contours. Such features can be based on points [1] and [2], blobs (Laplacian of Gaussian [3] or Difference of Gaussian [4]), intensities [5], gradients [6] and [7], colour, texture, or combinations of several or all of these [8]. The final descriptors need to characterize the image sufficiently well for the detection and classification task at hand. We will divide the various approaches to descriptor selection into two broad categories: **Sparse representations** are based on local descriptors of relevant local image regions. The regions can be selected using either

key point detectors, image fragments or parts detectors. On the other hand, **dense representations** are based on image intensities, gradients or higher order differential operators. Image features are often extracted densely (often pixel-wise) over an entire image or detection window and collected into a high-dimensional descriptor vector that can be used for discriminative image classification or labeling the window as object or non-object.

2.2 Local Shape-Based Human Detection

Mori et al . [Mori, 2002] model human body configurations where body part templates are represented by local Shape Context. In the later work, they apply normalized cuts segmentation and use shape, shading and focus cues for retrieving the body parts. M. Oren et al. [Oren, 1997] use Haar wavelet coefficients to build o global human model. Edgar Seeman, Bastian Leibe et al [Leibe,2003] studied different shape based human detection algorithms. There are mainly two kinds of shape based detection techniques: Global approach and Local approach.

2.2.1 Global Approach

Global approach is known as Global chamber matching technique [Gavrila,2000] and Local approach is known as Implicit Shape Model (ISM) [Harris, 1998].Object shape silhouettes to image structure are match on the Global chamfer matching approach. For that purpose, a silhouette is shifted over the image and a distance Dchamfer (T,L) between a silhouette T and the Edge image at each image location L is calculated.

2.2.2 Local Approach- Implicit Shape Model (ISM)

The local approach is subdivided into three subsections. Such as: Model training, Hypothesis generation, segmentation and verification. The ISM is trained by extracting local features from training images [Leibe, 2000].Then modeling their spatial occurrence distributed on the object. An Interest Point Detector is applied for each training images. After training with the images Hypothesis is generated and finally segmentation and verification is performed to detect the human.

By using the Interest Poit Detector local features are calculated. There are Interest Point Detector techniques such as Harris detector [Harris ,1988], Deference of Gaussian detector [Lowe, 2004] Harris-Laplace detector [Mikolajczyk, 2001], Hessian-Laplace [Mikolajczyk, 2004]

2.3 Dense Descriptors of Image Regions

One of the primary works using simple image intensities is the Eigen faces approach of [9]. Approaches using image gradient descriptors are [7], where histograms of gradients have been used. The Census algorithm [10] transforms the intensity space to an order space, where a bit pattern is formed by looking at the orders of a given pixel with its neighbors. [11] uses an improved version of this algorithm where they somewhat alleviate the problem of counting salt- and-pepper noise in a pixel multiple times. [12] proposed a method in which the penalty for an order flip is proportional to the intensity difference between the two flipped pixels, thereby improving the noise immunity. Finally, [13] presents a statistical approach whose match measure can be tuned to the underlying error process. All of these methods assume that the pixel locations do not vary across the two patches and are thus inappropriate for a feature matching problem where the pixel locations might undergo some shift. [14] makes feature descriptors with point pairs that are invariant to Gaussian noise. A penalty is awarded if there is an order change for a point pair between the two patches and such penalties for different pairs are summed in order to determine the difference between the two features. The Local Binary Patterns (LBP) descriptor [15], which is a variant of the Census approach [10] has also shown promise in texture description [16]. As the LBP operator produces a rather high dimensional histogram and is therefore difficult to use in the context of a region descriptor, a Center-Symmetric LBP (CS- LBP) which only compares center-symmetric pairs of pixels was considered for feature description in [17]. More recently, [18] came up with the idea of Center Symmetric-Local Ternary Patterns (CS-LTP) to make CS-LBP resistant to noise together with proposing a global order based descriptor (Histogram of Relative Intensities (HRI)) which handles saturation and illumination changes better.

2.4 Work in human Detection

[19] describes a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) -based variant in [20]. However we find that linear SVMs (weighted sums of rectified wavelet outputs) give similar results and are much faster to calculate. [19] shows results for pedestrian, face, and car. [21] takes a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [22]. [23] built an efficient moving person detector, using AdaBoost [24] to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. [6] built an articulated

body detector by incorporating SVM based limb classifiers over 1st and 2nd order Gaussian filters in a dynamic programming framework similar to those of [25] and [26]. [27] uses combinations of orientation position histograms with binary-threshold gradient magnitudes to build a parts based method containing detectors for faces, heads, and front and side profiles of upper and lower body parts. [28] uses a combination of [7] and [15] to build a more efficient descriptor and also try to overcome occlusion.

2.5 Different Types Edge Detector

2.5.1 Introduction

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. There are an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include Edge orientation, Noise environment and Edge structure. The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges. Edge detection is difficult in noisy images, since both the noise and the edges contain high frequency content. Attempts to reduce the noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges. Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity [1].

The operator needs to be chosen to be responsive to such a gradual change in those cases. So, there are problems of false edge detection, missing true edges, edge localization, high computational time and problems due to noise etc. Therefore, the objective is to do the comparison of various edge detection techniques and analyze the performance of the various techniques in different conditions. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

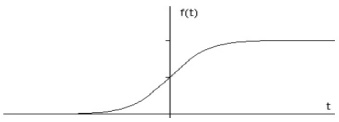
Gradient based Edge Detection:

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

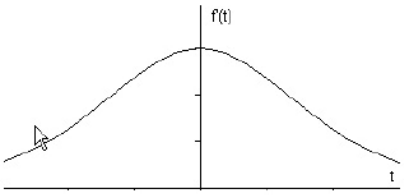
Laplacian based Edge Detection:

The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

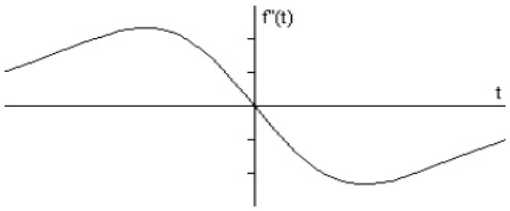
Suppose we have the following signal, with an edge shown by the jump in intensity below: Suppose we have the following signal, with an edge shown by the jump in intensity below:



If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to t) we get the following:



Clearly, the derivative shows a maximum located at the center of the edge in the original signal. This method of locating an edge is characteristic of the "gradient filter" family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, you can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is shown below:



2.5.2 Edge Detection Techniques

2.5.2.1 Sobel Operator

The operator consists of a pair of 3×3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 2.2: Masks used by Sobel Operator

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 component in each orientation (call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

Typically, an approximate magnitude is computed using:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

$$|G| = |Gx| + |Gy|$$

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

$$\theta = \arctan(Gy / Gx)$$

2.5.2.2 Robert's cross operator:

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The

operator consists of a pair of 2×2 convolution kernels as shown in Figure 2. One kernel is simply the other rotated by 90°[4]. This is very similar to the Sobel operator.

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 2.3: Masks used for Robert operator

These kernels are designed to respond maximally to edges running at 45° 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 component in each orientation (call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

although typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much faster to compute.

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

$$\theta = \arctan(Gy / Gx) - 3\pi / 4$$

2.5.2.3 Prewitt's operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

-1	0	+1
-1	0	+1
-1	0	+1

Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

Figure 2.4: Masks for the Prewitt gradient edge detector

2.5.2.4 Laplacian of Gaussian:

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output.

The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by:

$$L(x,y)= \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernelthat can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in Figure 4.

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Figure2.5: Three commonly used discrete approximations to the Laplacian filter.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages: Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.

The LoG (`Laplacian of Gaussian')[6] kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image.

The 2-D LoG function centered on zero and with Gaussian standard deviation has the form:

$$LoG(x,y)= -1/\pi\sigma^4\left[1- \left(\frac{x^2+y^2}{2\sigma^2}\right)\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

and is shown

0	1	0
1	-4	1
0	1	0

in Figure 5.

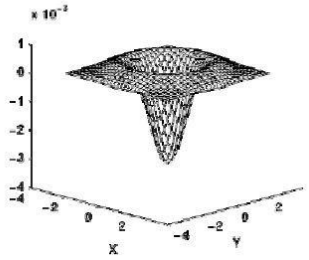


Figure 2.6: The 2-D Laplacian of Gaussian (LoG) function. The *x* and *y* axes are marked in standard deviations

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Figure2.7: Discrete approximation to LoG function with Gaussian S = 1.4

Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels shown in figure 4. This is because smoothing with a very narrow Gaussian (< 0.5 pixels) on a discrete grid has no effect. Hence on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussians.

2.5.2.5 Canny Edge Detection Algorithm

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

Step 1:-

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the 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 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 increases slightly as the Gaussian width is increased.

Step 2:-

After smoothing the image and eliminating the noise, the 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. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

The magnitude, or edge strength, of the gradient is then approximated using the formula:

$|G| = |Gx| + |Gy|$

Step 3:-

The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sumX is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

Theta = invtan (Gy / Gx)

Step 4:-

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5:-

After the edge directions are known, non-maximum suppression now has to be applied. Non- maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6:-

Finally, hysteresis [12] is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

2.6 Detection and Tracking Using Combination of Thermal and Visible Imaging

The previous method was useful in indoors but in outdoor environment it require a high resolution camera. In this case a fusion of Infrared Camera and Visible imaging is used. This significantly reduces the processing time and power required for detecting human. The operational diagram for such kind of processing system is shown below.

The main process of the technique are Segmentation and Classification. These are used in many detection techniques.

2.6.1 Segmentation

The most important part of any real time human detection technique based on images is firstly detecting the still objects from the background and ROI(Region of Interests).This process is known as segmentation. Most methods use intensity, texture and contrast properties in the image over a period of time to construct a back ground model. The back ground model is updated by averaging frames over a period of time to a account for slow changes in illumination. The back ground model is the subtracted on a pixel basis from the current image. There proposed system segment the object using the information of the temperature. Humans are segment by examining the hot object . Again not all hot object are humans for this reason this stage were also used for eliminating small clusters of hot objects that are unlikely to be people.

2.6.2 Classification

After segmentation there is a group of hot objects after that it is require classifying the hot objects which are human and which are not Amitage et al. used straightforward shape analysis. They used vertical histogram projection to get the shape of the object. They find that human have a shape, similar to a normal Gaussian curve which is differ from other hot objects likes cars, busses etc. by examining the shapes they classify the humans.

2.7 Summary

In this chapter various methods of human detection is described briefly. The main purpose of these researches is to detect human more still images by computer. The works has some limitations for this reason researches are going on to improve the techniques. The next chapter presents sour proposed human detection methodology.

3. PROPOSED HUMAN DETECTION METHODOLOGY

3.1 Introduction

In the previous chapter we have discussed various human detection systems, their successes and limitations. In this chapter we will discuss on proposed human detection system. The discussion is broken into several modules: Image acquisition, edge detection, and finally human detection by analyzing the shapes. Section 3.2 describes the proposed system architecture. Section 3.3 describes the detail of human detection technique.

3.2 Proposed System Architecture

Figure 3.1 shows a basic architecture of proposed human detection. In this propose system, images are captured using a digital camera. These images are passed through the human detection module. In the human detection module, input RGB images to convert into Gray-scale images; then normalized boundary is compared with predefined templates and if sufficient match is found then human is bounded by a rectangular box. After detecting human from the real-time image , the system can take several actions. Such as it can aware about the presence of the human by making alarm or displaying some light signal instructions.

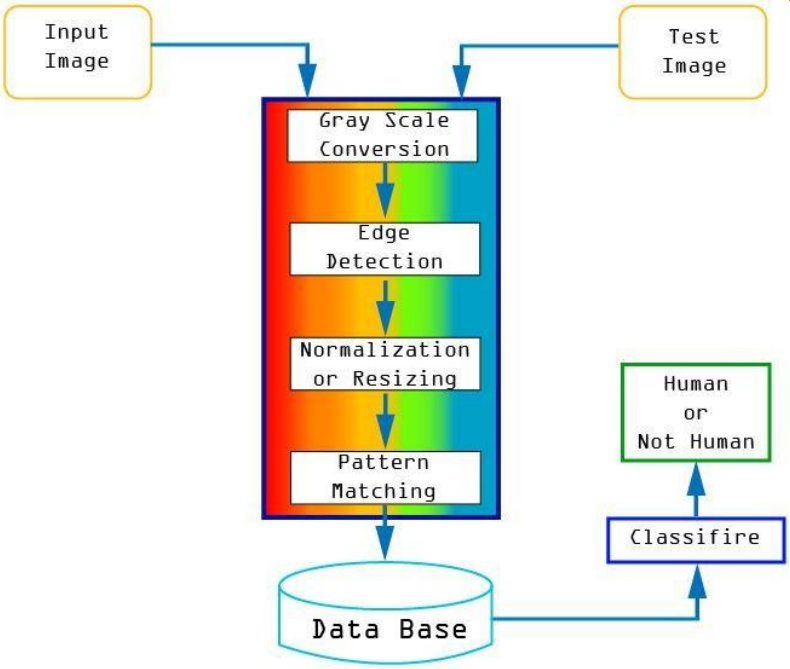


Figure 3.1 Proposed System Architecture

3.3 Details of Human Detection

Although motion is a very important cue for recognizing actions, when we look at such images, we can more or less understand human actions in the picture. This is mostly true in news or sports photographs, where the people are in stylized poses that reflect an action.. We used a digital camera to capture the image. From the segmented images boundary is detected and detected is compared with predefined templates for matching.

Human detection is closely related to general object recognition techniques. It involves two steps-training or learning phases and classifier for human detection phases Figure 3.2 presents the flow chart of the proposed human detection method. Each part of the work will be described in the following subsections. Subsection 3.3.1 focuses on the images Acquisition. Subsection 3.3.2 focuses on the Gray-scale conversion from RGB images. Subsection 3.3.3 describes on shape/ boundary detection method. Subsection 3.3.4 describes Normalization technique and finally subsection 3.3.5 focuses on pattern matching approaches.

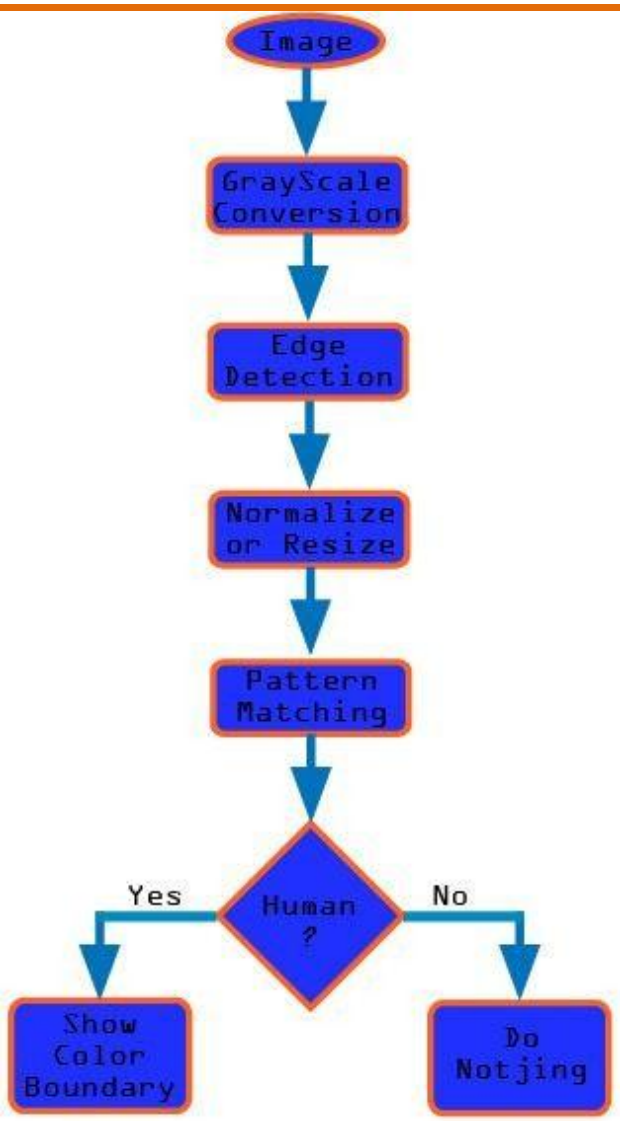


Figure 3.2 Flowchart of The Proposed System

3.3.1 Images Acquisition

We can get images various way such as digital camera, scanner and store image on computer hard disk.

3.3.2 Gray Scale Conversion

There are several methods for image conversion. In this paper, we use the gray scale conversion for further process to detected human area by traditional approach.

$$New_i = (R_i + G_i + B_i) / 3 \quad \text{.....(3.1)}$$



Figure 3.3 Gray Scale Conversion

3.3.3 Edge Detection

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

Step 1:-

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the 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 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 increases slightly as the Gaussian width is increased.

Step 2:-

After smoothing the image and eliminating the noise, the 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. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2000-000X
Vol. 1 Issue 5, July– 2017, Pages: 62-85

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

The magnitude, or edge strength, of the gradient is then approximated using the formula:

$|G| = |Gx| + |Gy|$

Step 3:-

The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

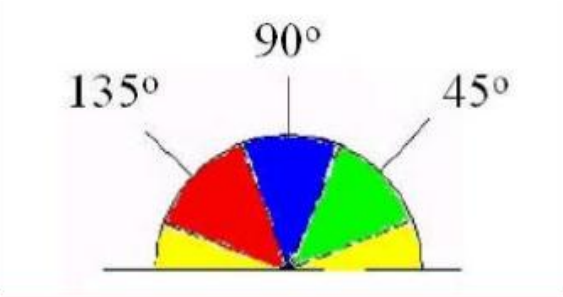
Theta = invtan (Gy / Gx)

Step 4:-

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows:

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.



Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

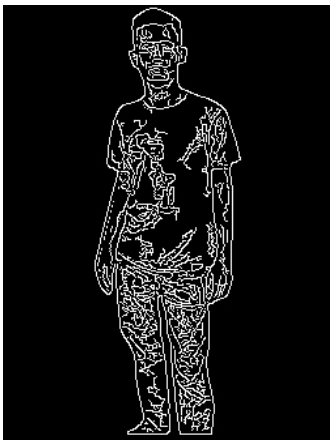
Step 5:-

After the edge directions are known, non-maximum suppression now has to be applied. Non- maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6:-

Finally, hysteresis [12] is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1





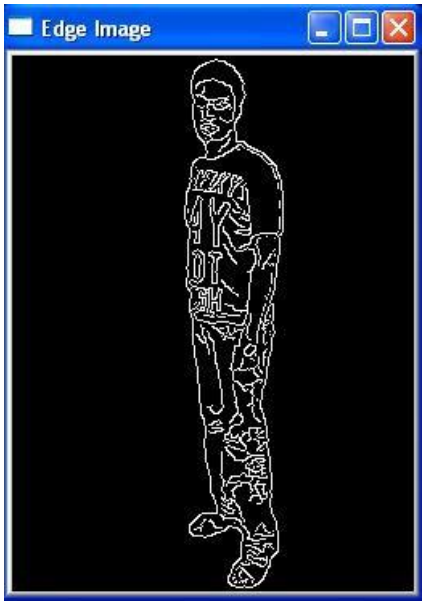


Figure 3.4 Original Image and Edge Detected Image of Canny Operator

3.5 Normalization or Resize

Normalization phase scale up all edge images into equal size. Each edge image is scaled to a rectangular image of M`xN` resolution. In our project, normalized images are of 180x200 resolutions. The edge image B[(0,0), (xm,ym)] is scale to image N[(0,0), (64x64)] using the equation 3.3.

$$N(x_i, y_i) = B(x_i \times S_x, y_i \times S_y) \quad \text{.....(3.3) Where,} \quad S_x = 64/x_m \text{ and } S_y = 64/y_m$$

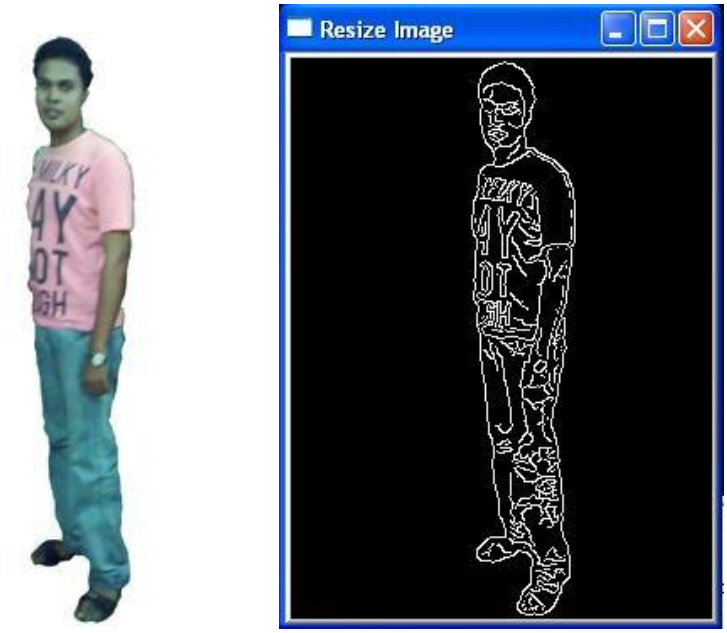


Figure 3.5 Example Scenario of Normalizing Method (A) Source Image (B) Before Normalizing (C) After Normalizing

3.3.6 Pattern Matching

After normalizing, the contour image is compared with the predefined template images. In section 33 we will describe template image formation technique. The detail of our matching algorithm is as follows:

Step 1:

Read the contour image, and save the co-ordinates that hold 1

Step 2:

Compare with known templates and measure hit and miss score.

a) Hit Score:

If Tst (x, y) = Tmp (x, y) = 1 then increment Hit Score.

b) Miss Score:
If Tst (x, y) ≠ Tmp (x, y) then increment Miss Score.

Step 3:
Calculate the Hit ratio

Hit Ratio:

$$Hr = \frac{H}{H+M}$$ where, H = Total Hit Score and

M = Total Miss Score

Step 4:
If Hr is greater than predefined threshold then human is localized and bounded using a rectangle color box.

The threshold is selected through experiment. We considered the threshold value is as 0.75. This box is drawn by calculating the minimum and maximum values of X and Y co-ordinates.

3.4 **Template Image Formation Technique**

To train the human detection system we collect several template images to detect the shape/contour of human. This system prepares the template by detecting the shape area, eliminating noises, detecting edge, and finally scales to 180×200. This boundaries of the human are used for matching with every captured image. We have taken 400 images of different human in the different poses and in different scenarios. All these template images are kept in size

180×200 resolutions. Figure shows example of several template images and figure shows the extracted boundaries of those images.

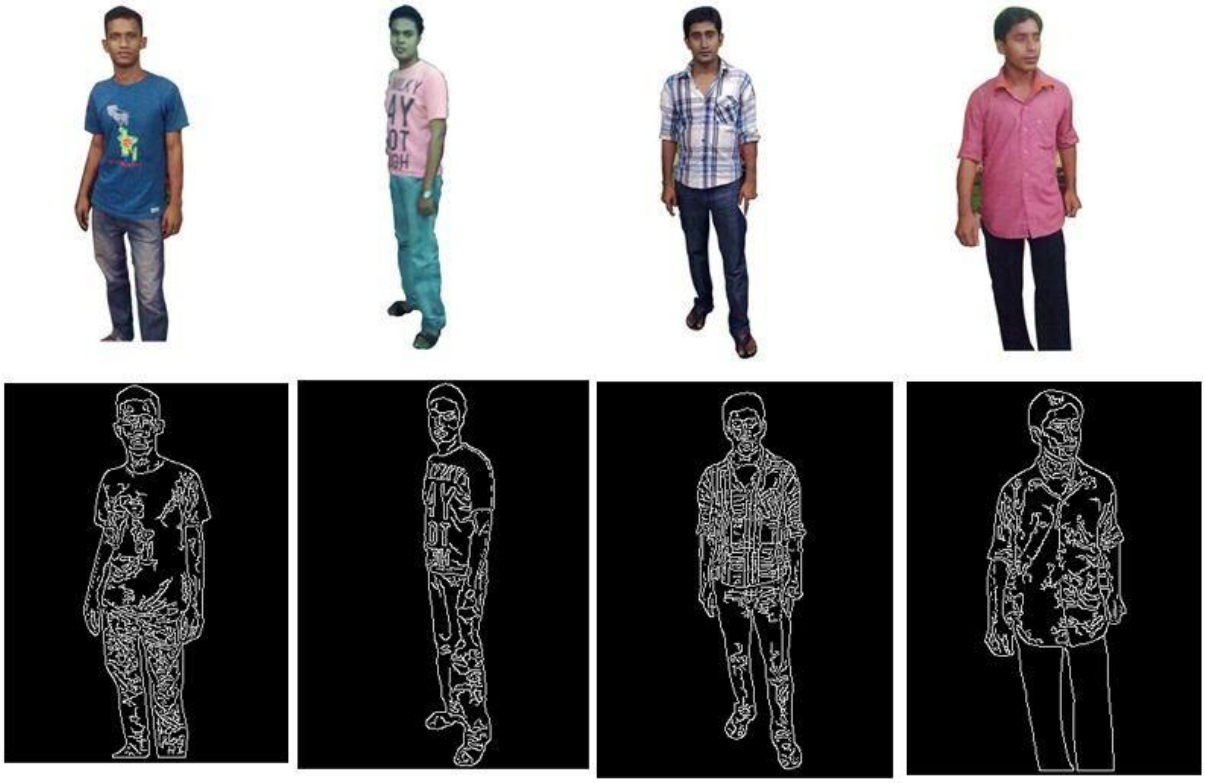


Figure 3.6 Example Template images and shape templates

3.5 Summary

In this chapter we have discussed different modules of proposed human detection system. This technique is mainly focused on detecting objects, noise elimination, shape detection and linking, and finally matching module. Next chapter presents experiments, results with related discussions.

4 EXPERIMENT, RESULT AND DISCUSSIONS

4.1 Introduction

This chapter presents experimental results together with valuable discussion. The experimental result focuses on following areas: results of conversion method, results of filtering method, results of boundary detection method and finally results of human detection method. Section 4.2 describes the experiment setup. Section 4.3 present the results of human detection.

4.2 Experiment Setup

In this thesis we uses core2 duo 2.8GHz PC with 1 GB RAM. We also use application software for subtract the background of an image. This image will be input to the system for further processing.

The proposed method is implemented using Microsoft Visual C++(6.0) , OpenCV library function, OpenGL programming language and Adobe Photoshop CS2.



Figure 4.1: Computer System

4.3. Experimental Results of Proposed System

This proposed system can detect human from white background still image. This section will present the result of the system. Subsection 4.3.1 show the result of color conversion method, Subsection 4.3.2 present the result of boundary detection method, Subsection 4.3.3 present the result of resizing method and finally Subsection 4.3.4 show the result of human detection method.

4.3.1 Result of Color Conversion

Color can be converted by use various method but we proposed the traditional approach that can convert the image from RGB to GRAY.





Figure: 4.2 A) Source Image B) Gray Image

4.3.2 Results of Boundary or Edge Detection

Boundary can be detected by scanning the image using masking but we proposed an algorithm, which is called (canny edge algorithm) that scans the image for detecting edges. Figure 4.5 shows the detected boundaries by using canny edge algorithm.

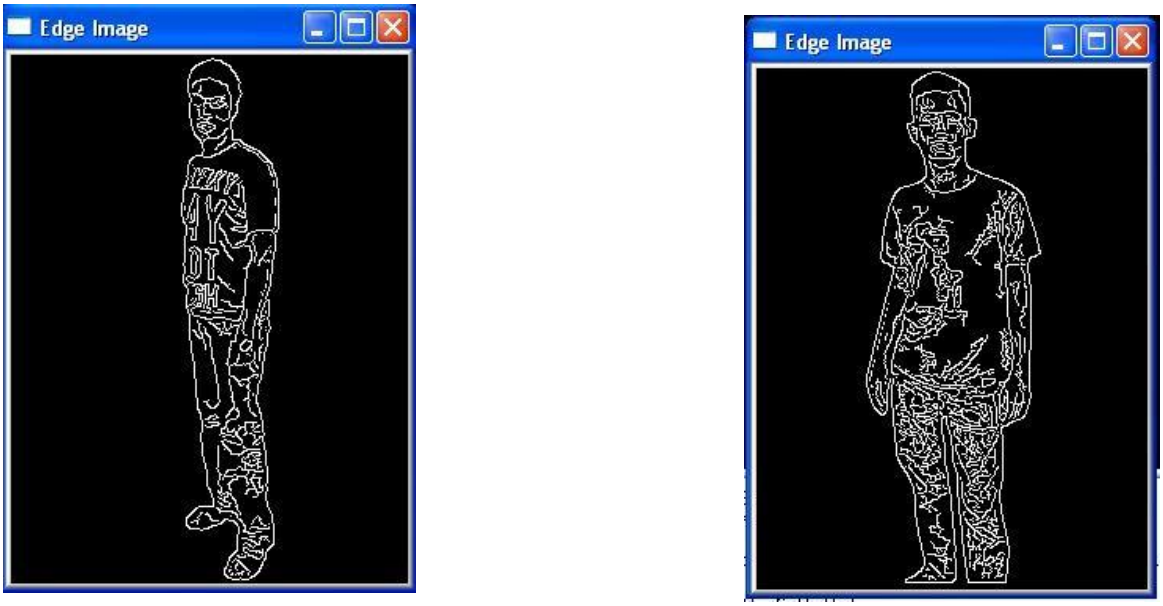


Figure 4.3: Result of edge detect

Performance of human detection varies, based on boundary detection techniques. Table 4.2 shows the comparison of human detection using canny edge operator boundary detection methods.

Table 4.1: Experiment Result of human Detection where Boundary is Detected using canny edge Scanning.

	Edge Detection Method
Experiment	Accuracy (%) (Scan Edge)
1	92.4
2	95.5
3	93.33
4	96.97
5	90.87
6	94.2
	Average =93.87

4.3.3 Result of Resizing Method

For resize the we used a formula, which given below:

$$N(x_i, y_i) = B(x_i \times S_x, y_i \times S_y) \quad \text{.....(3.3)}$$ Where,

$S_x = 64/x_m$ and $S_y = 64/y_m$



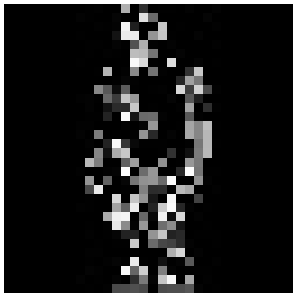


Figure 4.4: Result of Resizing Image

4.3.4 Results of human Detection

This thesis finally gives a visual output by surrounding a rectangular box of the detected human. Figure 4.7 shows a sample visual output. Left side of the window is the original image that is inputted by subtracting the background and right side of the window shows the detected human by making a rectangular box around the human.

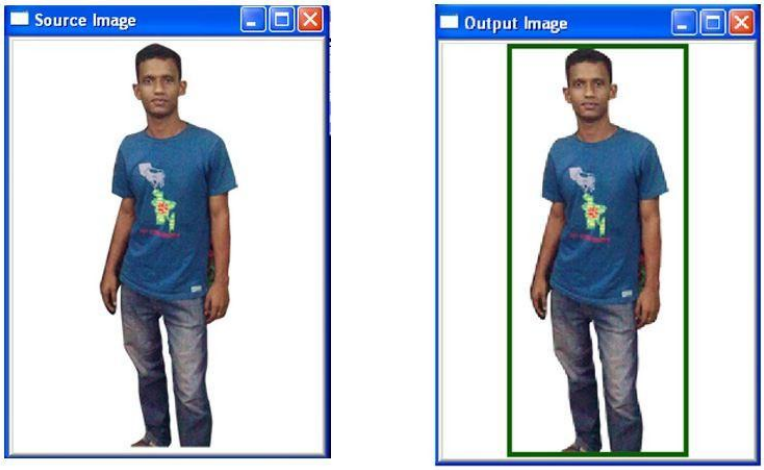


Figure 4.5: Sample Visual Output of Human Detection



Figure: 4.6 shows example results of human detection. Only true pedestrian are detected and bounded by a rectangular green box.

Performance of human detection method

Table 4.3 represents the performance evaluation of out proposed human detection method. Equation 4.1 and 4.2 defines the accuracy and precision of human detection method

$Pp = C / C + F * 100$

Here, Pp = Precision (%) of human detection

P = Number of total human

C = Number of Correctly detected human

F = Number of False detection

Table 4.2 : Performance of Human Detection Method

Experiment	Human Detected Correctly ©	False Detection (F)	Precision (%) (Pp)
#1	95	5	95
#2	38	0	100
#3	280	9	96.88
#4	378	25	93.79
#5	488	10	97.99
#6	370	12	96.85
#7	520	19	96.47

Performance of human detection method in term of accuracy is presented in figure 4.9 . In experiment number #7, we can see from the figure that the accuracy is highest. The duration of this experiment is 60 seconds and total correctly detected human is 120 among 150 human. In this case lighting condition was satisfactory. In some cases slight degradation of accuracy is observed. This is due to variation of noises and cluttered background. Figure 4.10 presents the precision of human detection method for various experiments. In some cases the precision is very low. This is also due to variation background and noise. Overall performance of the detection method is satisfactory. The average precision is 92.72% and the average accuracy is

90.05%.

From the above experiment we can say that this very efficient and successful in detecting human in different poses. The outcome can be used in the area where human or pedestrians are related. Mainly the idea of this thesis was come from the idea of autonomous driving. When in future there will be no driver on the vehicle and the vehicle will able to drive itself then the idea of this thesis can be used.

4.4 Summary

In this chapter we have presented experimental results of the proposed human detection system.

In the next chapter we conclude our work by mentioning the major contributions, limitations of the system and our future works.

5. CONCLUSION AND FUTURE WORKS

This thesis work describes the human detection from still images using the contour/boundary based matching. This system is tested in different position for detecting human. From the experimental result we conclude that performance of the system is satisfactory.

5.1 Contribution

This thesis has great contributions in the field of image analysis and object recognition. In this thesis we have presented an algorithm for detecting human. The proposed system runs with satisfactory success rates. The contribution of our work can be summarized as follows.

- A novel method for detecting human from still images.
- A novel method for shape/contour detection from still images.
- A satisfactory performance in detecting human using contour/shape matching.
- Average accuracy and precision of human detection method in the system is 93.05%

  And 95.72%.

5.2 Limitation and Future Works

The major limitation of the thesis is that the system is not dynamic. It only can detect human from still images.

Another limitation of the thesis is that the system cannot detect if the multiple human is present

There are situation such as when a human is just backside of another human, this thesis cannot detect the two human separately.

Another limitation of this thesis is that the system cannot subtract the background automatically. It is to do by manually for the system input images to detect the human.

In future we will make this thesis more robust against the various limitations and detect multiple human.

5.3 Concluding Remarks

The ultimate goal of this thesis was to established an efficient, robust and user friendly system to detect human. Our achievement from this research is satisfactory. We aspire to do more research in the same field.

REFERENCES

[1] Harris, C., Stephens., M.: A combined corner and edge detector. Alvey Vision Conference (1998) 147–151

[2] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. 7th European Conference on Computer Vision 1 (2002) 128–142

[3] Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30 (1998) 79–116

[4] Lowe, D.G.: Local feature view clustering for 3D object recognition. Conference on Computer Vision and Pattern Recognition (2001) 682–688

[5] Kadir, T., Brady, M.: Scale, saliency and image description. International Journal of Computer Vision 45 (2001) 83–105

[6] Cordelia, R.R., Schmid, C., Triggs, B.: Learning to parse pictures of people. European Conference on Computer Vision (2002) 700–714

[7] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1 (2005) 886–893

[8] Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 530 –549

[9] Sirovitch, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America 2 (1987) 586–591

[10] Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision (1994) 151–158 on Pattern Analysis and Machine Intelligence 20 (1998) 415–423

[11] Mittal, A., Visvanathan, R.: An intensity-augmented ordinal measure for visual correspondence. CVPR '06: Proceedings of the 2006 IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition (2006) 849–856

[12] Singh, M., Parameswaran, V., Ramesh, V.: Order consistent change detection via fast statistical significance testing. IEEE Conference on Computer Vision and Pattern Recognition, 2008 (2008)

[13] Gupta, R., Mittal, A.: SMD: A locally stable monotonic change invariant feature descriptor. 10th European Conference on Computer Vision (2008) 265–277

[14] Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29 (1996) 51–59

[15] Ojala, T., Pietik¨ainen, M., M¨aenp¨a¨a, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 971–987

[16] Heikkil¨a, M., Pietik¨ainen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recognition 42 (2009) 425–436

[17] Gupta, R., Patil, H., Mittal, A.: Robust order-based methods for feature description. In: IEEE Conference on Computer Vision and Pattern Recognition. (2010)

[18] Papageorgiou, C., Poggio, T.: A trainable system for object detection. International Journal of Computer Vision 38 (2000) 15–33

[19] 20. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 349–361

[20] 21. Gavrila, D., Philomin, V.: Real-time object detection for smart vehicles. Proceedings of the 7th International Conference on Computer Vision 87{93 (1999)

[21] 22. Gavrila, D.M., Giebel, J., Munder, S.: Vision-based pedestrian detection: The

[22] PROTECTOR system. IEEE Intelligent Vehicles Symposium (2004) 13–18

[23] Jones, M., Viola, P., Viola, P., Jones, M.J., Snow, D., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Conference on Computer Vision (2003) 734–741

[24] Schapire, R.E.: The boosting approach to machine learning, an overview. MSRI Workshop on Nonlinear Estimation and Classification (2002) 25International Journal of Computer Vision 43 (2001) 45–68

[25] Krystian Mikolajczyk, C.S., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. European Conference on Computer Vision (2004) 69–82

[26] Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. International Conference on Computer Vision (2009)