A Simple Digital Semaphore Decoder Using DSP Processor

Syed Abd. Rahman Abu Bakar, Sheikh Hussain Sheikh Salleh, Ahmad Zuri Sha'ameri, Leow Cheah Wei
DSP Research Group
Fakulti Kejuruteraan Elektrik
Universiti Teknologi Malaysia
81310 Skudai, Johor
e-mail: syed@suria.fke.utm.my

Abstract: This paper presents an innovative semaphore decoder using a DSP processor. The goal is to automate the interpretation of a semaphore code captured by a camera using basic image processing functions. Some of these functions involve image thresholding, image thinning, and image recognition. The system also introduced two new operations in its implementation and these were person detection and shoulder detection operations. This system has been implemented using actual image and has successfully decoded the semaphore language.

Keywords
Digital image processing, region detection, thresholding, thinning.

I. INTRODUCTION

Semaphore is a sign language for sending signal manually using hands by holding two flags in certain predetermined combination of positions. These positions indicate letters A to Z. Figure 1 shows the hand position and its equivalent character for the semaphore code. Semaphore is mostly used by the Navy throughout the world to communicate between two vessels whether in dock or at sea. It is used to transmit messages between two nearby ships when the radio communication breaks down or when the use of radio communication is to be avoided. Therefore, this involves personnels that should be able to perform and understand the code precisely. Since humans are vulnerable to making mistakes serious consequences could result depending on the error occurred.

This paper presents a technique based on digital signal processing that will interpret this semaphore code and display its equivalent alphabetical form digitally using the computer screen. The work utilized a Texas Instrument TMS320C31 DSP chipset. The objective of this digital semaphore decoder is two folds. Firstly, by using DSP processor, the system can be integrated into a possibly hand-held digital camera thus avoiding bulky and heavy machines. Secondly, by having the DSP processor acts as the interpreter, any personnel will be able to get the message and not restricted to skilled officer only. Thus, even in the absence of skilled interpreter, the communication can still be established.

In implementing this system, several image processing techniques were utilized and these include image thresholding for binarizing the gray scale image, image thinning for reducing the pixel area, segmentation for locating region of interest, and finally image recognition for interpreting the result.

This paper is organized as follows. Section II discusses the general flow and concept on the implementation of the semaphore decoder. The role of the DSP processor is highlighted in this section. Next in section III, the main image processing techniques used in the implementation are discussed. The discussion on the new operations i.e. person detection and shoulder detection are elaborated in Section IV. Performance of the system is shown in Section V. Finally, conclusions and further works are given in Section VI.

II. GENERAL CONCEPT

The general block diagram of the semaphore decoder is shown in Figure 2. The gray shaded region indicates operations done using the DSP processor. Since this is the first ever prototype on the digital semaphore decoder, we started out with static image rather than acquiring video sequences. The system thus starts with reading raw image file and then converts the gray scale image into its binary image. Once the image has been thresholded, the binary image of the person is then thinned so that the person is represented by one pixel width lines. Prior to this thinning process, the system will detect the position of the person as well as the position of his shoulder. This step is very crucial, as these positions will be used as the reference points in establishing correct positions of the hands. The system will then perform the region detection process to determine the location of both hands. The result will then be compared to the codebook containing the associated code for each alphabet. Once the alphabet is found it will be displayed on the screen.

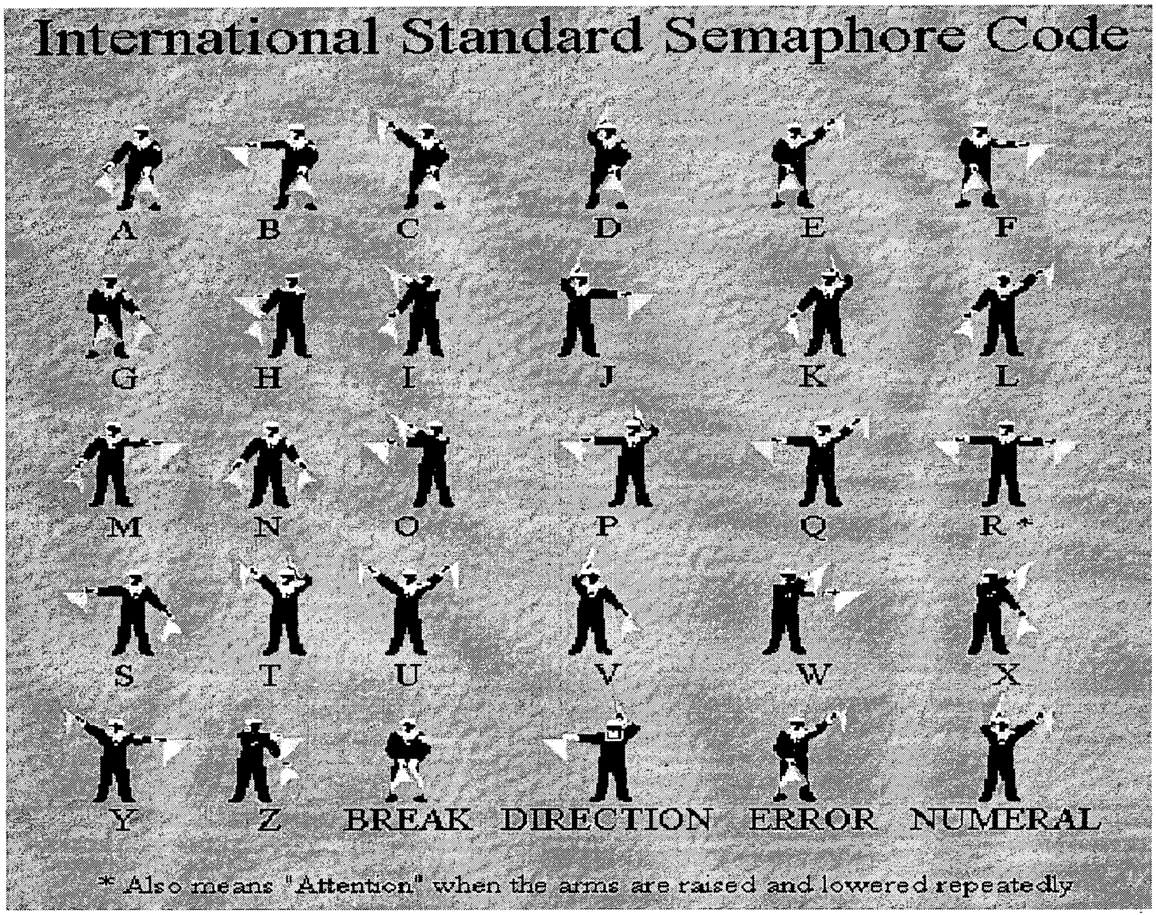


Figure 1: Semaphore code

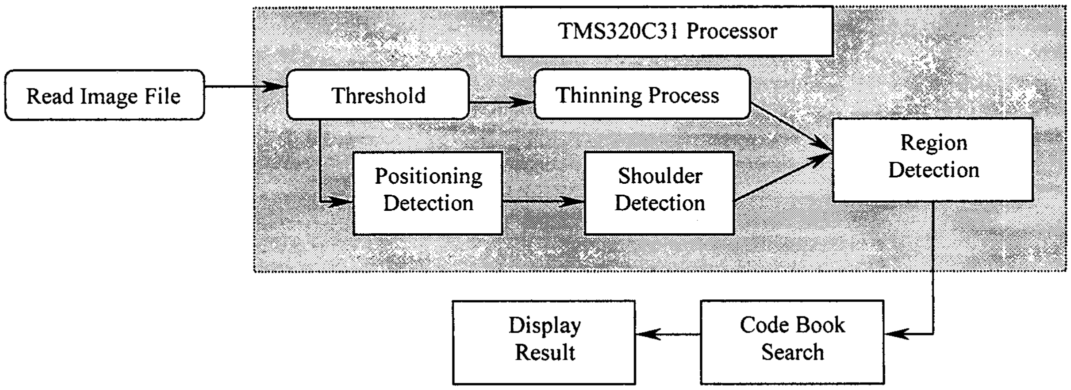


Figure 2: General block diagram for the Digital Semaphore Decoder.

III. IMAGE PROCESSING

The images used in this system were gray scale images of 8 bits resolution. Since the system does not need full 8 bits dynamic range, the images need to be thresholded into binary images. Faster computation could also be achieved by using binarized images. While using global threshold value is simple and less time consuming, it more often overlooks fine detail areas which results in loss of crucial information. Our approach is to vary this threshold value according to the nature of the surrounding pixels in a window of size 3x3 or 5x5.

We first compute the average intensity value in the window. If the average is less than 110 we set the threshold value to 110. If the average intensity in greater than 120 we then set the threshold value to 120. If the average value falls in between, we then set the threshold value to this average. The value 110 and 120 were chosen heuristically based on the images that we studied.

The next process involved is the thinning process. Here we used established technique to obtain thinned image [1]. The idea of thinning the image is to reduce the amount of pixels need to be processed, as this will also at the same time speed up the computation process.

Once the image has bee thinned, the next stage is to perform the region detection so that proper decoding can be done. Assuming that the person and shoulder positions have been detected (these two operations will be discussed in the next section), the image will be divided into 10 rectangular areas as shown in Figure 3 below.

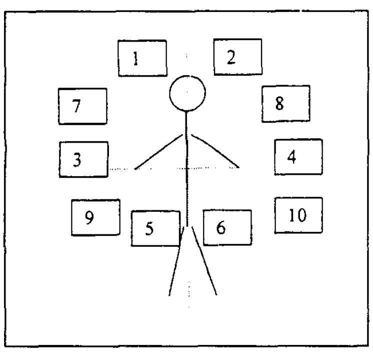


Figure 3: 10 regions surrounding the object.

These regions represent the probable locations of both hands of the personnel. The system will then count the number of pixels in the odd-numbered boxes as well as in the even-numbered boxes. The odd-numbered box that has the highest count will indicate the position of the right hand of the person while the even-numbered box with the highest count will indicate the position of the left hand. The combination of these two boxes are then compared to a predefined codebook where the semaphore codes have been stored. From the codebook, the result of the code is displayed using the ASCII character.

IV. POSITION DETECTION

One of the key criteria towards precise interpretation of the semaphore signal is to properly detect the location of the person. Since the person can be anywhere in the image area, it is important that the position of the person be identified. Once the location of the person has been determined, next operation is to locate the position of the shoulder. This shoulder detection is also essential so that proper alignment of the 10 regions of interest be positioned correctly. The detection of the position of the person and the position of the shoulder is done only once. Since this is our first experiment, we assume that the camera will be stationary and the image of the person needs to be registered first.

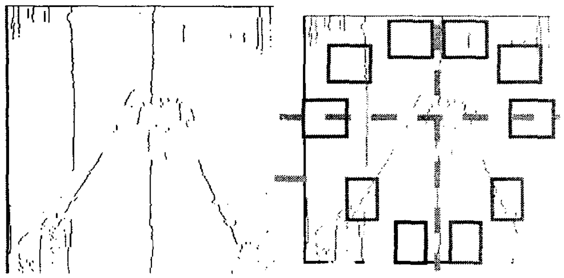
In the person position detection, we have used a simple technique where we have counted the number of pixels along each column. Assuming that the person has been assigned to a black pixel in a binary image and assuming that the height of the person occupies the image area, we decide that the column position of the person is the column that has the highest number of black pixels. This position is the center column for the person. We realize however, that this technique has its limitation.

Once the column position of the person has been located, the next stage is to locate the position of the shoulder. The strategy that we used is also similar that is counting pixels. Using the center column that we have found earlier, we count the number of connected black pixel towards the right and the left of this center position and add up the two values. We decide the location of the shoulder when we have reached the first highest constant number of black pixel hits.

V. EXPERIMENTAL RESULTS

We have tested our system to the real image of a person posing with a semaphore sign language. In our experiment we have used Texas Instrument TMS320C31 developer starter kit (DSK) with a 486 PC as its host. We have chosen this processor due to its simplicity and stability in programming [2]. Our test image and the result are shown in Figure 4 below.

(a) (b)



(c) (d)

Figure 4: Experimental result using actual figure. (a) Person in a position of a letter N. (b) Binaized image of the person. (c) Result of thinning operation. (d) Detecting the 10 regions of interest.

In this experiment, the person is showing the letter 'N'. In the output screen, our system has correctly identified this code as letter 'N'.

VI. CONCLUSIONS

We have presented a novel digital semaphore decoder in this paper. This decoder uses simple image processing techniques implemented in a C31 DSP processor. Among the techniques involved are thresholding, thinning, person and shoulder detection, and region detection. This decoder is at is initial stage and therefore still confined to many limitations.

It is our interest to upgrade the performance of this system. This will include blob analysis with center of mass calculation technique for robust positioning of the person and shoulder. Artificial neural network can also be a good candidate for the interpretation part.

REFERENCES

[1]. A. K. Jain, *Fundamentals of Digital Image Processing*, pp 382-383, Prentice-Hall, 1989.

[2]. R. Chassaing, *Digital Signal Processing with C and the TMS320 C30*, Wiley, New York, 1992.