Hardware Emulation of HOG and AMDF Based Scale and Rotation Invariant Robust Shape Detection

Murat Peker
Electrical and Electronics Engineering
Niğde University
Nigde,Turkey

Halis Altun
Department of Computer Engineering
Mevlana University
Konya,Turkey

Fuat Karakaya
Electrical and Electronics Engineering
Niğde University
Nigde,Turkey

***Abstract*—In this study, a hardware emulation of HOG and AMDF based scale and rotation invariant robust shape detection for Field Programmable Gate Arrays (FPGA) is described. For this purpose, a robust algorithm with light-computational load has been developed based on features extracted from histogram of oriented gradients (HOG). A normalization scheme is proposed to obtain scale-invariant robust features using HOG algorithm. Also a novel method for shape detection is proposed using Average Magnitude Difference Function (AMDF) which leads to a rotation-invariant and computationally light shape detection method. It is shown that the proposed method is robust against noise as well. Also it is indicated that the performance of the method is highly satisfactory for implementation on real-life industrial problems.**

Keywords-Shape detection; Histogram of Oriented Gradients (HOG); Average Magnitude Difference Function (AMDF); FPGA

I. INTRODUCTION

Although it is intensively investigated problem, studies related to image processing and pattern recognition are still demanding especially for real-time industrial application and hence new approaches and methods are currently being proposed by the researchers to improve the existing algorithms. One of a typical application area which employs image processing and pattern recognition is to recognize and classify an object based on its shape. Finding an object automatically in an image using features related to its shape is mostly a fundamental step in the automatic detection and classification of the objects. Therefore shape detection plays an important role and has an intensive usage in various applications.

In literature there are plenty of related studies on the object detection based on its shape. A few of them is listed as hand shape and gesture recognition [1], traffic sign detection [2], object classification [3] etc. In these studies researchers are proposed to use variety of methods to achieve a satisfactory performance such as the model based region grouping [4], Haar wavelet features [2], shape symmetry [3], Hough transform [5], edge based shape detection [6].In this study a hardware simulation for FPGA implementation is designed for scale-and-rotation invariant method and it is tested in Matlab environment. The proposed methods use a set of features extracted from the histogram of oriented gradients (HOG). The proposed shape detection system is, first of all, confirmed on software using MATLAB with a dataset of 780 samples and 96.26% success rate is obtained under ideal conditions (i.e. no contamination due to noise). The main motivation of developing such a robust method on FPGA is that real time signal processing algorithms inherently introduce very heavy computational load to be handled by the processor. As FPGA's have a capability of parallel processing, it is possible to implement real time application if a parallelization scheme of algorithms is possible. Furthermore FPGA's are preferable because of power requirements and cost advantages. Yet implementation of an algorithm on FPGA is not a straightforward and it is probably very complex task. This requires verification of the proposed methods in simulation environment and we have selected Matlab for this purpose.

The proposed algorithm makes use of the HOG features which is suitable for parallelization but requires non-linear operations that are hard to implement directly on hardware. Within this study, we propose to alleviate the problem related to implementation of HOG based shape detection. HOG features exhibit superior performance in many applications under different and severe conditions. The HOG was initially proposed by Shashua et al. [7] as identifiers for pedestrian recognition systems and then successfully adapted for human recognition in a complex environment by Dalal et al. [8]. HOG features consist of the orientation gradients of image pixels so the angle distribution can easily be extracted. In our approach HOG features are used to classify the object using the average magnitude difference function (AMDF). We propose to use the AMDF as a measure of dissimilarity between an input image and reference images due to its easy-to-implement nature. AMDF associates the reference image with the input image, if it exhibits the lowest dissimilarity. Based on this result, the input image will be recognized and labeled.

The performance of proposed method achieves a superior success with proper parameter selection. The crucial parameter in HOG features is the size of angle bins. Also it has been unearthed that normalization process of HOG features has shown a positive effect on the performance. The effect of increasing the bin size of HOG algorithm will result in more detailed angle distribution information about the object in the image, hence in our study we select the bin size which gives the

978-1-4673-4810-2/12/$31.00 ©2012 IEEE

maximum accuracy in term of classification as indicated in [9]. As the AMDF measures the discrepancy between two sources of HOG, the discrepancy due to scale variability should be addressed. The normalization of the parameters deals with this problem and it leads to eliminate the effects of the scale discrepancy between reference images and the input image. Also the proposed approach has an advantage that the selection of AMDF as a measure of dissimilarity gives an opportunity of effective hardware implementation of the proposed algorithm due to its simplicity for embedded system.

The paper is organized as follows. Section II explains the algorithms for shape detection used in this study, namely HOG and AMDF. Section III explains the hardware simulation which is tested in MATLAB environment. Section IV reports the result, by comparing software and hardware simulation of the proposed method. Section V concludes and indicates future directions.

II. PROPOSED ALGORITHMS

In the proposed approach HOG features are fed to AMDF in order to measure the difference between the reference HOG features and that of the current input image, which contains an object to be classified. The current input image is associated with a reference in terms of their similarity and then, based on the result, the current input image is labeled accordingly. As the proposed approach is based on HOG features and AMDF, these are introduced in details below.

A. HOG Features

HOG features reflect the characteristics of pixel orientation (θ) and magnitude values. The purpose of the HOG features is to define an image as a group of local histograms. These histograms contain the sum of the gradient magnitudes at the orientation of gradients in a local region of the image. The implementation steps of the HOG could be given as follows: First of all, edges I_x and I_y are obtained (1) and (2) by applying horizontal and vertical Sobel filters, S_h and S_v, on the input image I.

$$I_x = I * S_h \quad (1)$$
$$I_y = I * S_v \quad (2)$$
$$|\, G \,| = \sqrt{I_x^2 + I_y^2} \quad (3)$$
$$\theta = arctan(\frac{I_x}{I_y}) \quad (4)$$

where S_h, S_v are horizontal, vertical Sobel operators and G and θ are the amplitude and angle of the gradient, respectively. Then magnitude and orientation angle (G and θ) of the gradients are calculated from I_x and I_y using (3) and (4). Finally a histogram is obtained using the gradients at specific angle which is determined by the number of bins. In our implementation, the effect of the bin number on the performance is investigated by varying the number of bins between 0 ° to 360 °. Also in our HOG implementation we do not divide the image into sub-regions and hence no calculation is performed to obtain local histograms. Instead whole image is considered as one single region and calculations are performed accordingly. A further consideration is to normalize the HOG values in order to obtain a scale-invariant algorithm. Normalization is performed by dividing all bin values with the maximum HOG value which is formalized in (5). This operation makes it sure that the maximum value of the angle bins will be set to 1.

$$hog_i = \frac{hog_i}{max\,(hog)} \qquad i = 1,2 \dots N \quad (5)$$

B. AMDF as a measure of dissimilarity

AMDF is an algorithm which is mostly used in speech analysis to determine the periodicity of the waveform under consideration. The most common application of the AMDF is to determine the pitch [10] of speech in a frame-wise fashion. The idea is to find the dissimilarity between the speech waveform in a frame and its lagged version. If a degree of periodicity presents in the waveform, at a certain lag which corresponds to the period of the waveform, the difference between two waveforms will be minimum. Therefore we may interpret that the minimum value of AMDF indicates the similarity between two waveforms. Another mostly used method to measure similarity/dissimilarity between waveforms is Autocorrelation function (ACF). In contrast to AMDF, ACF produces a maximum value at the lag at which two waveforms show a high similarity. Compared to AMDF, on the other hand, ACF will introduce a heavy computational load as it requires huge number of multiplication. Because of this heavy load ACF might be not suitable algorithm to implement on hardware compared to AMDF. This is the main reason why AMDF has been chosen in this study, as a measure of the dissimilarity/similarity between HOG vectors. AMDF between the i^{th} reference HOG and that of j^{th} input image is defined as a function of the lag value of τ in (6) as follows

$$f_{AMDF_{ij}}(\tau) = \frac{1}{N}\sum_{i=1}^{N}\left|S_i - S_{j-\tau}\right| \quad \tau = 0,1,\dots N-1 \quad (6)$$

where N is the number of bins in the histogram, S_i is the histogram of reference image, $S_{j-\tau}$ the lagged version of histogram of the input image. The value of τ ranges from 0 to N-1. In this study the reference HOG vectors, which associate with particular class label, are extracted for each class of the objects. Using these reference HOG templates it is possible to measure the dissimilarity between the reference object and the current input object, and to determine the class label of the current input object based on the minimum value of AMDF. Thus, the input object is classified as an object from the i^{th} class if the function $f_{AMDF_{ij}}(\tau)$produces the lowest value for the i^{th} HOG reference features

III. HARDWARE SIMULATION OF THE METHOD

This section explains the basics of the implementation issues of the proposed method on FPGA along with the simulation related issues on MATLAB. In addition, the details of the approximation relevant to the number formats and bit lengths are also discussed in this section.

HOG algorithm consists of nonlinear computations such as square root and arctangent operations. These functions should be implemented on hardware using the integer number format in order to avoid floating point operations. Therefore, in the simulation of the method on Matlab, input image format should be converted to 8 bit unsigned gray-scale values and the pixel values, in this case, should be set between 0 and 255. Another issue related number format should be dealt after Sobel operator. The values from Sobel operator are normalized so that the pixel values might vary between -255 and 255. These values are represented by 9 digits of signed format.

The first operation on the image is to smooth the image using a Gaussian filter with a kernel of 5x5 pixel size. After application of Sobel operators on the input image, two major modules are employed to calculate the HOG values. The first module is used to calculate the angle of the gradient, θ, using (3) and the second module is used to calculate the magnitude of the gradient using (4). As it is clear, the calculation of the angle requires arctangent operation and there are different approaches in practice to implement this non-linear operation. In literature "look up tables (LUT)", "piece-wise", and "CORDIC" based arctangent approximations proposed by the researchers. In this study we employed the LUT and piece-wise approximations in combination to obtain high accuracy with less demand on the resources. As the precise calculation of the HOG values depend on the angle of the gradient, θ, the calculation of angle is highly crucial on the performance of the method and therefore as an accurate approximation as possible is desirable at the expense of resource usage. At the final stage, a histogram is obtained from the magnitudes by accumulating the gradients to the corresponding bins.

As the value of pixel after Sobel filter will be in the range of [-255, 255],the input range of the arctangent approximation must cover this whole range. Under this requirement, it is observed that if the piece-wise approach is solely utilized in approximation to the arctangent function, the error dramatically increases for the θ values greater than 45°. On the other hand if only LUT approach is used as an approximation, the memory requirement of the module will dramatically increase. So we propose to use a hybrid method. If the angle θ is above 45°, the LUT approach will be used and if it is less than 45 ° the piece-wise approximation will be utilized. Using this hybrid method the angle can be acquired with a maximum error of 2.23° below 45° and with an error of 0.5° if the angle is greater than 45°. In the LUT method the arctangent values are calculated according to (7) after obtaining the horizontal and vertical gradients, namely I_x and I_y.

$$\arctan(I_x, I_y) = LUT(I_x + I_y + 255) \tag{7}$$

In the explained hybrid method only the angle values within 0° to 90° are calculated. In order to cover all the range of the angle, the region of the angle should be determined using the sign of the I_x and I_y and the angle is accordingly mapped to its corresponding region. The effectiveness of the proposed method is illustrated in Fig. 1. It is clearly seen from Fig. 1 that the proposed hybrid approximation produces highly accurate values with very small error.

Based on the calculation of the θ angle of the pixel, the corresponding bin is determined and the magnitude of the gradient of the pixel is added this bin recursively in order to obtain the histogram of gradients. Accumulating the gradient is governed by a simple threshold which is set as a mean of HOG values. If the gradient of the pixel is less than the mean value, this gradient is simply ignored.

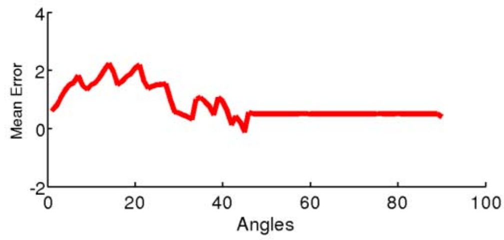

Figure 1. Error between 0° and 90°

Another consideration for a light computational method is to eliminate the square-root operation in the calculation of gradients. If we omit the square-root in (3), the magnitude of gradient can be given as follows

$$|G| = I_x^2 + I_y^2 \tag{8}$$

As we obtain HOG features simply by accumulating the gradient values into corresponding bins, omitting the square-root in (3) does not change the structure of the HOG features. The final step to obtain HOG feature is the normalization using (9).

$$HOG_i = (HOG_i * 255)/\max(HOG) \tag{9}$$

This normalization operation set the values between 0 and 255. As it can be easily seen the order of operation in the normalization procedure is crucial. If division is carried out before multiplication the operation yields 0 all the time due to integer arithmetic.

Fig. 2 shows the HOG values for a rectangular shape. Fig. 2(a) illustrates the shape, Fig. 2(b) gives the calculated HOG values using floating point operations and finally in Fig. 2(b) HOG values are shown which are calculated using the proposed methods. As it can be seen clearly the proposed method is able to capture the structure of HOG despite of simplified arithmetic operations. This behavior of the proposed method is expected to have a high success rate in detection stage.

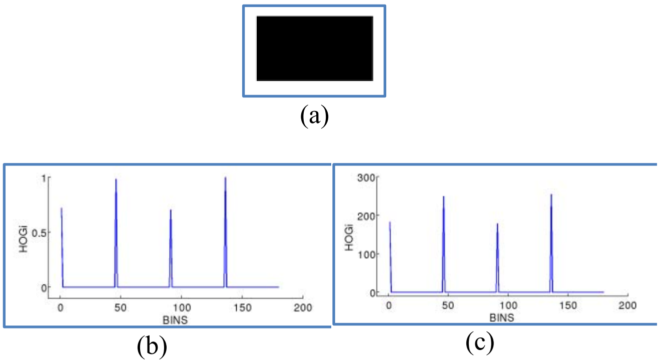

Figure 2. a) input image b)HOG values obtained using floating point number format, c) HOG values obtained using proposed method

As AMDF requires only addition/subtraction and shift operation it is straightforward to implement on hardware.

IV. SIMULATION RESULTS

In the hardware implementation of the proposed method on FPGA, the number format will be the bottlenecks which should be addressed carefully. Hence, in this paper, a pure integer number format for the calculations of Sobel filter and the HOG values are proposed. The proposed method is tested with a dataset that includes 8 different reference geometric shapes as object classes. In order to obtain objects with different scale and rotation, all reference shapes have firstly been rotated between 0° to 360° with a step size of 10° (expect the circle shape as rotation is not meaningful for this shape)and the size of each resultant images is scaled with 0.75 and 1.5. So the final test dataset size consists of 780 images. In Fig. 3 reference shapes and their classes are shown.

Class Label	**1**	**2**	**3**	**4**
Shape				
Class Label	**5**	**6**	**7**	**8**
Shape				

Figure 3. The basic shapes in dataset and their classes

The floating point implementation of the method on Matlab reaches up to of 100% success rate with a bin size of 180. With the proposed method, on the other hand, the success rate decrease down to 96.26% with the same bin size. The degradation of performance is caused, as expected, by the loss of precision due to the number format. However, the performance is still satisfactory. Investigation of results revels that the error is mostly occurred between certain classes such as the 6^{th} and the 7^{th} classes, 3^{th} and the 8^{th} classes.

Table 1: Confusion matrix for the proposed method

Classes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	**3**	0	0	0	0	0	0	0
2	0	**111**	0	0	0	0	0	0
3	0	0	**111**	0	0	0	0	0
4	0	0	0	**111**	0	0	0	0
5	0	0	10	0	**101**	0	0	0
6	0	0	0	0	0	**94**	17	0
7	0	0	4	0	0	0	**107**	0
8	0	0	11	0	0	1	0	**99**

In order to test the robustness of the proposed methods the input images are deteriorated by adding a noise with different level. 5%, 10% and 25% of the image pixels are contaminated by the pepper&salt noise (See Fig. 4). The success rates for the contaminated images are measured as 95.95%, 95.89% and 95.76%, respectively. The results indicate that the proposed method is robust against noise and the performance of the proposed method does not degrade significantly.

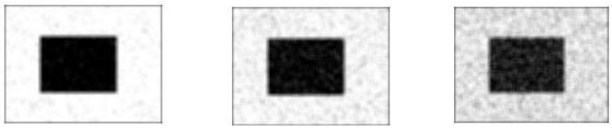

Figure 4. The contaminated input image by the pepper&salt noise (5%, 15% and 25% respectively)

V. CONCLUSION

A scale and rotation invariant method for shape detection which is suitable for implementation on FPGA is devised. It is shown that the performance of the method is highly acceptable and does not suffer from noise. Under the ideal condition 96.26% recognition rate is obtained and this figure does not reduce dramatically under the pepper&salt noise. Experiments show that the performance degradation due to noise is measured as 0.3220% 0.3844% 0.5194%, respectively. As a future work, the implementation of the proposed method on FPGA will be dealt.

ACKNOWLEDGEMENT

This work partially funded by Ministry of Science, Industry and Technology under the contract of 01063.STZ.2011-2

REFERENCES

[1] Ong, E. and Bowden, R., “A Boosted Classifier Tree for Hand Shape Detection”, Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FG'04), 2004, Seoul, Korea, 889 – 894

[2] Bahlmann, C. Zhu, Y. Visvanathan R. Pellkofer, M. and Koehler, T., “A system for traffic sign detection, tracking, and recognition using color, shape, and motion information”, Intelligent Vehicles Symposium, 2005, Siemens Corp. Res. Inc., Princeton, NJ, USA, 255-260

[3] Loy, G. and Barnes, N., “Fast shape-based road sign detection for a driver assistance system”, Intelligent Robots and Systems, Comput. Vision & Active Perception Lab., 2005, Stockholm, Sweden 70-75

[4] Liu, L. and Sclaroff, S., “Deformable shape detection and description via model-based region grouping”, Computer Vision and Pattern Recognition, 1999. IEEE Computer Society

Conference, 2 vol. XXIII-637-663

[5] Maji, S. and Malik, J., “Object detection using a max-margin Hough transform”, Computer Vision and Pattern Recognition, 2009. CVPR 2009, 1038-1045

[6] Moon, H. Chellappa, R. and Rosenfeld, A., “Optimal edge-based shape detection”, Image Processing, IEEE Transactions on, 2002, vol.11, no.11, 1209-1227

[7] Shashua A. Gdalyahu Y. and Hayon G., “Pedestrian detection for driving assistance systems: Single-frame classification and system level performance”, In Proceedings of IEEE Intelligent Vehicles Symposium, 2004, 1-6

[8] Dalal, N. and Triggs B., “Histograms of oriented gradients for human detection”, In C. Schmid, S. Soatto, and C. Tomasi, editors, International Conference on Computer Vision and Pattern Recognition, volume 2, 2005, 886–893

[9] M. Peker, H. Altun, F. Karakaya, “Amdf Based Shape Detection Using Hog Features”, International Conference on Informatics, 2011

[10] G. Muhammad (2011), “Extended Average Magnitude Difference Function Based Pitch Detection”, The International Arab Journal of Information Technology, Vol. 8, No. 2,197-208