Simple Implimention of Sobel Image Edge Detection in Facial Regonition

Hendrik
Department of Computer Science
IPB University, Bogor, Indonesia
Hendrik_49@apps.ipb.ac.id

Peter Juma Ochieng
Department of Biochemistry& Biotechnology
Kenyatta University, Nairobi, Kenya
Peter26juma@gmail.com

***Abstract*— Image edge detection is a process of locating the edge of an image which is important in finding the approximate absolute gradient magnitude at each point *I* of an input grayscale image. The problem of getting an appropriate absolute gradient magnitude for edges lies in the method used. Therefore, this project implement and compare Sobel–Feldman operator, Laplacian and Canny filter design methods to detect edges of an images. The Sobel operator performs a *2-D* spatial gradient measurement on images. Transferring a *2-D* pixel array into statistically uncorrelated data set enhances the removal of redundant data, as a result, reduction of the amount of data is required to represent a digital image. The Sobel edge detector uses a pair of *3 x 3* convolution masks, one estimating gradient in the x-direction and the other estimating gradient in *y*–direction. The Sobel detector is incredibly sensitive to noise in pictures, it effectively highlight them as edges. Hence, Sobel operator is recommended in massive data communication found in data transfer.**

Keywords—Sobel-Feldman, Edge Detection, gradient

I. Introduction

Image processing is important in modern data storage and data transmission especially in progressive transmission of images, video coding (teleconferencing), digital libraries, and image database, remote sensing. It has to do with manipulation of images done by algorithm to produce desired images [1]. Digital Signal Processing (DSP) improve the quality of images taken under extremely unfavourable conditions in several ways: brightness and contrast adjustment, edge detection, noise reduction, focus adjustment, motion blur reduction etc [2]. The advantage is that image processing allows much wider range of algorithms to be applied to the input data in order to avoid problems such as the build-up of noise and signal distortion during processing [3]. Many of the techniques of digital image processing were developed in the 1960's at the Jet Propulsion Laboratory, Massachusetts Institute of Technology (MIT), Bell laboratory and few other places. But the cost of processing was fairly high with the computing equipments of that era. With the fast computers and signal processors available in the 2000's, digital image processing became the most common form of image processing and is general used because it is not only the most versatile method but also the cheapest. The process allows the use of much more complex algorithms for image processing and hence can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means [4]. Thus, images are stored on the computers as collection of bits representing pixel or points forming the picture elements [5]. Firstly, images are a measure of parameter over space, while most signals are measures of parameter over time. Secondly, they contain a great deal of information [6]; image processing is any form of information processing for which the input is an image, such as frames of video; the output is not necessarily an image, but can be for instance be a set of features of the image[7]. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. The process involves the enhancement or manipulation of an image which resulted in another image, removal of redundant data and the transformation of a 2-D pixel array into a statically uncorrelated data set [8]. Since images contain lots of redundant data, scholars have discovered that the most important information lies in it edges (Canny, 1986). Edges being the local property of a pixel and its immediate neighbourhood, characterizes boundary [9]. They correspond to object boundaries, changes in surface orientation and describe defeat by a small margin. Edges typically correspond to points in the image where the gray value changes significantly from one pixel to the next. Edges represents region in the image with strong intensity contrast; representing an image by its edges has the fundamental advantage that the amount of data is reduced significantly while retaining most of image's vital information with high frequencies [10]. Thus, detecting Edges help in extracting useful information characteristics of the image where there are abrupt changes [11]. Edge detection is a process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. Edges consist of meaningful features and contained significant information. It's reduce significantly the amount of the image size and filters out information that may be regarded as less relevant, preserving the important structural properties of an image [12]. Most images contain some amount of redundancies that can sometimes be removed when edges are detected and replaced, when it is reconstructed [13]. Eliminating the redundancy could be done through edge detection. When image edges are detected, every kind of redundancy present in the image is removed [14]. The purpose of detecting sharp changes in image brightness is to capture important events.

Applying an edge detector to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. The image quality reflects significant information in the output edge and the size of the image is reduced. This in turn explains further that edge detection is one of the ways of solving the problem of high volume of space images occupy in the computer memory. The problems of storage, transmission over the Internet and bandwidth could be solved when edges are detected[15]. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when images are divided into areas corresponding to different objects.

Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. The process involves the enhancement or manipulation of an image which resulted in another image, removal of redundant data and the transformation of a 2-D pixel array into a statically uncorrelated data set [16]. Since images contain lots of redundant data, scholars have discovered that the most important information lies in it edges [17]. Edges being the local property of a pixel and its immediate neighbourhood, characterizes boundary[18].

They correspond to object boundaries, changes in surface orientation and describe defeat by a small margin. Edges typically correspond to points in the image where the gray value changes significantly from one pixel to the next. Edges represents region in the image with strong intensity contrast; representing an image by its edges has the fundamental advantage that the amount of data is reduced significantly while retaining most of image's vital information with high frequencies [19].

Thus, detecting Edges help in extracting useful information characteristics of the image where there are abrupt changes [20]. Edge detection is a process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. Edges consist of meaningful features and contained significant information. It's reduce significantly the amount of the image size and filters out information that may be regarded as less relevant, preserving the important structural properties of an image [21]. Most images contain some amount of redundancies that can sometimes be removed when edges are detected and replaced, when it is reconstructed [22] . Eliminating the redundancy could be done through edge detection. When image edges are detected, every kind of redundancy present in the image is removed [23].

The purpose of detecting sharp changes in image brightness is to capture important events. Applying an edge detector to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. The image quality reflects significant information in the output edge and the size of the image is reduced. This in turn explains further that edge detection is one of the ways of solving the problem of high volume of space images occupy in the computer memory. The problems of storage, transmission over the Internet and bandwidth could be solved when edges are detected [24]. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when images are divided into areas corresponding to different objects.

II. RELATED METHODS

Different methods are used to detect edges in image processing among these is Roberts Cross Algorithms. Robert process a photograph into a line drawing, transform the line drawing into a three-dimensional representation and finally display the three-dimensional structure with all the hidden lines removed, from any point of view (Robert, 1965). The Roberts cross algorithm (Mario& Maltoni, 1997) performs a 2-D spatial gradient convolution on the image. The main idea is to bring out the horizontal and vertical edges individually and then to put them together for the resulting edge detection. The two filters highlight areas of high special frequency, which tend to define the edge of an object in an image. The two filters are designed with the intention of bringing out the diagonal edges within the image. The G_x image will enunciate diagonals that run from the top-left to the bottom-right where as the G_y image will bring out edges that run top right to bottom-left. The two individual images G_x and G_y are combined using the approximation equation.

$$|G| = |G_x| + |G_y|$$

The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multistage algorithm to detect a wide range of edges in images. In addition, canny edge detector is a complex optimal edge detector which takes significantly longer time in result computations. The image is firstly run through a Gaussian blur to get rid of the noise. When the algorithm is applied, the angle and magnitude is obtained which is used to determine portions of the edges to retain. There are two threshold cut-off points where any value in the image below the first threshold is dropped to zero and values above the second threshold is raised to one. Canny (1986) considered the mathematical problem of deriving an optimal smoothing filter given the criteria of detection, localization and minimizing multiple responses to a single edge. He showed that the optimal filter given these assumptions is a sum of four exponential terms. He also showed that this filter can be well approximated by first-order derivatives of Gaussians. Canny also introduced the notion of non-maximum suppression, which means that given the presmoothing filters, edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. Another algorithm used is the Susan edge detector. This edge detection algorithm follows the usual method of taking an image and using a predetermined window centered on each pixel in the image applying a locally acting set of rules to give an edge response (Vincent, 2006). The response is then processed to give the output as a set of edges. The Susan edge filter has been implemented using circular masks (kernel) to give isotopic responses with approximations used either with constant weighting within it or with Gaussian weighting. The usual radius is 3.4 pixels, giving a mask of 37 pixels, and the smallest mask considered is the traditional *3□3* mask. The 37 pixels circular mask used in all feature detection experiments is placed at each point in the image and for each point the brightness of each pixel within the mask is compared with that of nucleus. The comparison equation is

$$C(\vec{r}, r_o) = \begin{cases} 1 & if \quad |I(\vec{r}) \quad - \quad I(\vec{r_o})| \leq t \\ 0 & if \quad |I(\vec{r}) \quad - \quad I(\vec{r_o})| > t \end{cases}$$

where r r is the position of the nucleus in the dimensional image, r r_o is the position of any other point within the mask, ⊃(*r* r *)* is the brightness of any pixel, ***t*** is the brightness in difference threshold and *C* is the output of the comparison. This comparison is done for each pixel within the mask where total *n* of the outputs *(c)* is given as

$$n(\vec{r}_{0^o}) = \sum_{\vec{F}} C(\vec{r}, \vec{r}_o)$$

III. METHODOLOY

A. *Sobel Filter Design*

Most edge detection methods work on the assumption that the edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if one take the derivative of the intensity value across the image and find points where the derivative is maximum, then the edge could be located. The gradient is a vector, whose components measure how rapid pixel value are changing with distance in the *x* and *y* direction. Thus, the components of the gradient may be found using the following approximation:

$$\frac{\partial f(x,y)}{\partial x} = \Delta x = \frac{f(x+dx,y) - f(x,y)}{dx}$$

$$\frac{\partial f(x,y)}{\partial y} = \Delta y = \frac{f(x+dx,y) - f(x,y)}{dy}$$

where *dx* and *dy* measure distance along the *x* and *y* directions respectively. In discrete images, one can consider *dx* and *dy* in terms of numbers of pixel between two points. $dx = dy = 1$ *(*pixel spacing) is the point at which pixel coordinates are (*i*, *j*) thus,

$$\Delta x = f(i+1,j) - f(i,j)$$
$$\Delta y = f(i,j+1) - f(i,j)$$

In order to detect the presence of a gradient discontinuity, one could calculate the change in the gradient at (*i, j*) .This can be done by finding the following magnitude measure

$$M = \sqrt{\Delta x^2 + \Delta y^2}$$

and the gradient direction θ is given by

$$\theta = \tan^{-1}\left[\frac{\Delta y}{\Delta x}\right]$$

The Sobel operator is an example of the gradient method. The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function (Sobel & Feldman, 1968). The different operators in eq. (5) and (6) correspond to convolving the image with the following marks.

$$\Delta x = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}, \Delta y = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$$

The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & +2 & -1 \end{bmatrix} * A$$

where $*$ here denotes the 2-dimensional signal processing convolution operation. Since the Sobel kernels can be decomposed as the products of an averaging and a differentiation kernel, we compute the gradient with smoothing given by

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

The *x*-coordinate is defined here as increasing in the "right"-direction, and the *y*-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2}$$

Using this information, we can also calculate the gradient's direction:

$$\Theta = \tan^{-1}\left(\frac{G_x}{G_y}\right)$$

where for example, Θ is 0 for a vertical edge which is lighter on the right side.

IV. RESULTS AND DISCUSSION

The Sobel operator performs a *2-D* spatial gradient measurement on an image. Typically, it is used to find the approximate absolute gradient magnitude at each point *I* of an input grayscale image. The Sobel edge detector uses a pair of *3 x 3* convolution masks, one estimating gradient in the *x*-direction and the other estimating gradient in *y*- direction. A convolution is usually much smaller than the actual image. As a result, the mask is slide over the image manipulating a square of pixels at a time. The mask is slides over an area where the input image changes with that pixel's value and then shifts one pixel to the right and continues to the right until it reaches the end of the row which automatically starts again at the beginning of the next row. It is important to note that pixels in the first row and last row, as well as the first and last column cannot be manipulated by a *3 x 3* mask. This is because when placing the centre of the mask over a pixel in the first row for example, the mask will be outside the image boundaries. The *Gx* mask highlights the edges in the horizontal direction while the *Gy* mask highlights the edges in vertical direction. After taking the magnitude of both, the resulting output detects edges in both directions. This is done by:

A. Sobel operator edge dection

Technically, it is a discrete differential operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel–Feldman operator is either the corresponding gradient vector or the norm of this vector. The Sobel–Feldman operator is based on convolving the image with a small, separable, and integer-valued filter in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high-frequency variations in the image. Since the intensity function of a digital image is only known at discrete points, derivatives of this function cannot be defined unless we assume that there is an underlying continuous intensity function which has been sampled at the image points. With some additional assumptions, the derivative of the continuous intensity function can be computed as a function on the sampled intensity function, i.e. the digital image. It turns out that the derivatives at any particular point are functions of the intensity values at virtually all image points. However, approximations of these derivative functions can be defined at lesser or larger degrees of accuracy. The Sobel-Feldman operator represents a rather inaccurate approximation of the image gradient, but is still of sufficient quality to be of practical use in many applications. More precisely, it uses intensity values only in a 3×3 region around each image point to approximate the corresponding image gradient, and it uses only integer values for the coefficients which weight the image intensities to produce the gradient approximation.

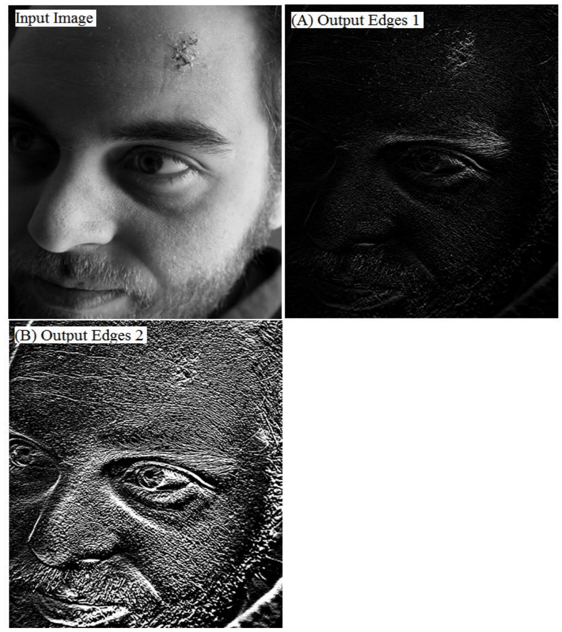

Fig. 1. Sobel operater (A):output edge apature size 3; (B) output edges appeture size 7.

B. Laplacian Method

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 (see zero crossing edge detectors). 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, and hence the two variants will be described together here. The operator normally takes a single graylevel image as input and produces another graylevel image as output. 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: (i) since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. (ii) The LoG (`Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image Note that as the Gaussian is made increasingly narrow, the LoG kernel becomes the same as the simple Laplacian kernels shown in Figure 1. This is because smoothing with a very narrow Gaussian ($\sigma < 0.5$ 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.

The LoG operator calculates the second spatial derivative of an image. This means that in areas where the image has a constant intensity (*i.e.* where the intensity gradient is zero), the LoG response will be zero. In the vicinity of a change in intensity, however, the LoG response will be positive on the darker side, and negative on the lighter side. This means that at a reasonably sharp edge between two regions of uniform but different intensities, the LoG response will be: (a) zero at a long distance from the edge; (b) positive just to one side of the edge; (c) negative just to the other side of the edge; (d) zero at some point in between, on the edge itself.

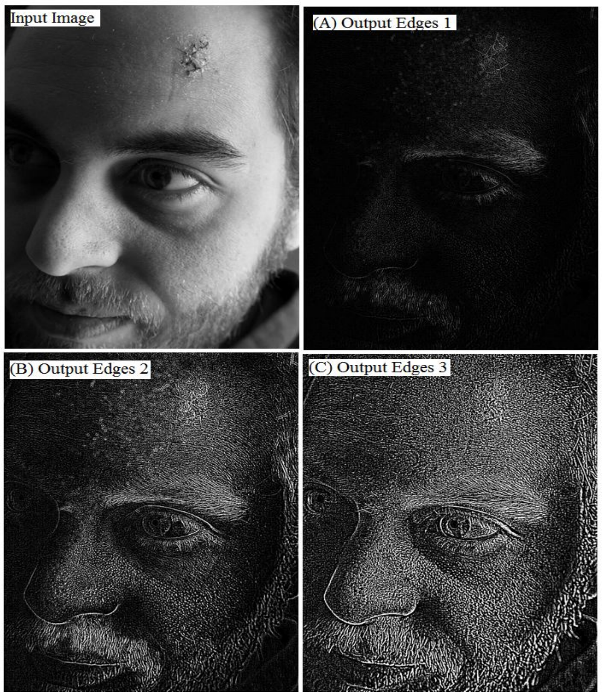

Fig. 2. Laplacian method (A):output edge apature size 3; (B) output edges appeture size 5; (C): output edges appeture size 7

C. Canny method

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. A more refined approach to obtain edges with sub-pixel accuracy is by using the approach of differential edge detection, where the requirement of non-maximum suppression is formulated in terms of second- and third-order derivatives computed from a scale space representation (Lindeberg 1998) – see the article on edge detection for a detailed description. A variational explanation for the main ingredient of the Canny edge detector, that is, finding the zero crossings of the 2nd derivative along the gradient direction, was shown to be the result of minimizing a Kronrod–Minkowski functional while maximizing the integral over the alignment of the edge with the gradient field (Kimmel and Bruckstein 2003). See article on regularized Laplacian zero crossings and other optimal edge integrators for a detailed description. The Canny algorithm is adaptable to various environments. Its parameters allow it to be tailored to recognition of edges of differing characteristics depending on the particular requirements of a given implementation. In Canny's original paper, the derivation of the optimal filter led to a Finite Impuse Response filter, which can be slow to compute in the spatial domain if the amount of smoothing required is important (the filter will have a large spatial support in that case). For this reason, it is often suggested to use Rachid Deriche's infinite impulse response form of Canny's filter,

Which is recursive, and which can be computed in a short, fixed amount of time for any desired amount of smoothing. The second form is suitable for real time implementations in FPGAs or DSPs, or very fast embedded PCs. In this context, however, the regular recursive implementation of the Canny operator does not give a good approximation of rotational symmetry and therefore gives a bias towards horizontal and vertical edges.

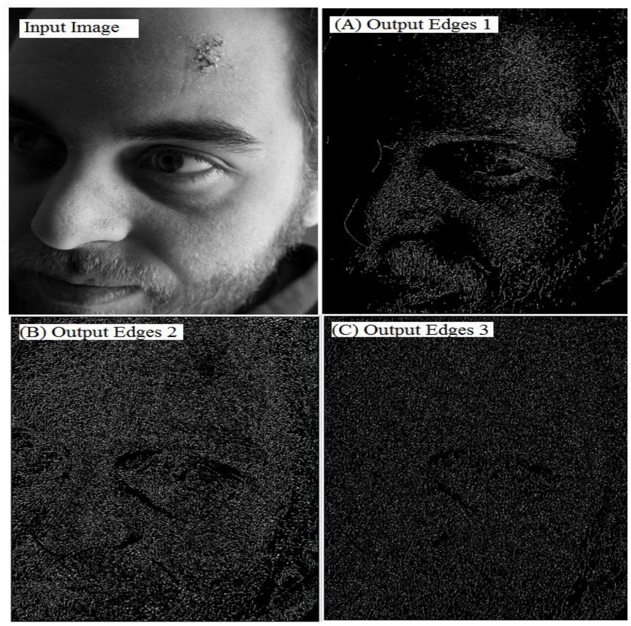

Fig. 3. Canny method (A):output edge apature size 3; (B) output edges appeture size 5; (C): output edges appeture size 7

V. CONCLUSSION

The Sobel operator performs a *2-D* spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point *I* of an input grayscale image. The Sobel edge detector uses a pair of *3 x 3* convolution masks, one estimating gradient in the *x*-direction and the other estimating gradient in *y*–direction. It is easy to implement than the other operators. Transferring a *2-D* pixel array into statistically uncorrelated data set enhances the removal of redundant data, as a result, reduction of the amount of data required to represent a digital image. Considering data communication especially the internet, massive data transfer causes serious problems for interactive network users. Edge detection helps in optimizing network bandwidth and it is needed to keep track of data flowing in and out of the network. It helps to extract useful features for pattern recognition. Although the Sobel operator is slower to compute, it's larger convolution kernel smoothness the input image to a greater extent and so makes the operator less sensitive to noise. The larger the width of the mask, the lower its sensitivity to noise and the operator also produces considerably higher output values for similar edges. Sobel operator effectively highlights noise found in real world pictures as edges though, the detected edges could be thick. The Canny edge detector and similar algorithm solved these problems by first blurring the image slightly then applying an algorithm that effectively thins the edges to one-pixel. This may constitute a much slower process, hence, Sobel operator is highly recommended in massive data communication found in image data transfer. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image.

ACKNOWLEDGMENT

The authors express their gratitude to Ministry of Research and Higher Education, Republic of Indonesia, for funding part of this research project through research grant and also to Department of Computer Science, IPB University and Department of Biochemistry and Biotechnology, Kenyatta University, for their contribution toward this project.

REFERENCES

[1] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989. Agaian, S. S., Baran, T. A., & Panetta, K. A. Transform-based image compression by noise reduction and spatial modification using Boolean minimization. *IEEE Workshop on Statistical Signal Processing*, 2003, pp. 226 – 229.

[2] Baker, S., & Nayar, S. K. Pattern rejection. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, 1996, 544-549.

[3] Canny, J. F. A computational approach to edge detection. *IEEE Trans Pattern Analysis and Machine Intelligence, 1986, 8*(6), 679-698.

[4] Chang-Huang, C. (2002). *Edge detection based on class ratio*. 152, sec.3, Peishen Rd., Shenkeng, Taipei, 2002, 22202, Taiwan, R.O.C.

[5] Folorunso O., Vincent, O. R., & Dansu, B. M. Image edge detection: A knowledge management technique for visual scene analysis. *Information Management and Computer Security,2007, 15*(1), 23-32.

[6] Gonzalez, R., & Woods, R. *Digital image processing* (2nd ed.). Prentice-Hall Inc. 2002, 567-612.

[7] Guthe, S., & Strasser, W. Advanced techniques for high-quality multi-resolution volume rendering. *Graphics and Visualization,2004, 28*, 25-78.

[8] Keren, D., Osadchy, M., & Gotsman. C. Antifaces: A novel, fast method for image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23*(7), 747-761.

[9] Mario, D., & Maltoni. Direct gray-scale minutiae detection in fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997. 19*(1), 27-40.

[10] Michael, U. Mathematics properties of the JPEG 2000 wavelet filters. *IEEE Transactions on Image Processing, 2003, 12*(9), 080-1090.

[11] Milan, S., Vaclav, H., & Roger, B. Image processing analysis and machine vision. London: Chapman and Hall, 2002, 255-280.

[12] Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*.

[13] Priotr, P.Practical signal decomposition design based on Haar – Wavelet transformation. *Journal of Applied Computer Science, 2004, 2*(1), 61-82.

[14] Roberts, L. G. Machine perception of three-dimensional solids. In J. T. Tippett (Ed.), Optical and electro-optical information processing 1965, pp. 159-197).

[15] Cambridge, Massachusetts: MIT Press. Scharr, H. (1996). *Digitale Bildverarbeitung und Papier: Texturanalyse mittels Pyramiden und Grauwertstatistiken am Beispiel der Papierformation*. Master's thesis, FakultÄat fÄur Physik und Astronomie, Ruprecht-Karls-UniversitÄat Heidelberg, Germany.

[16] Sobel, I., & Feldman, G. (1968). *A* 3 × 3 isotropic gradient operator for image processing. Presented at a talk at the Stanford Artificial Project.

[17] Sparr, G. Image processing and pattern classification for character recognition. Center for Mathematical Sciences, Lund University, 2002, 2, 25-78.

[18] Vincent, O. R., Folorunso, O., & Agboola, A. A. A. Modified algorithm for transform-based image compression. *Journal of Computer Science and its Application, 2006, 13*(1), 43-48.

[19] Vincent, O. R.Optimization of network bandwidth using image compression. *Proceedings of the National Conference on advanced Data Computing Communications and Security*, Kadi, India. 2007.

[20] Yuval, F. *Fractal image compression (theory and application).* Institute for non-linear Science, University of California, San Diego, USA. 1996.