EDGE DETECTION METHODOLOGIES FOR COLOR IMAGES

A dissertation submitted in partial fulfilment of the requirements for the M. Tech. (Computer Science) degree of the Indian Statistical Institute

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE

2011

BY

AVISEK CHAKRABORTY

Under the guidance of

Prof. C. A. Murthy
Machine Intelligence Unit
Indian Statistical Institute

Acknowledgments

It is my great pleasure to express my heartfelt gratitude to my supervisor Professor (Dr.) C. A. Murthy, Machine Intelligence Unit , Indian Statistical Institute (ISI Kolkata) for guidance, motivation and freedom to work during the present study and in preparation of the script. His active interest, care, concern and unequivocal help throughout the course of investigation and in the preparation of the manuscript made this a success.

I am grateful to the authorities of ISI (Director and Dean of studies) for providing me the financial assistance and all the necessary facilities during master's programme.

The images used for the experiments and results were obtained from the "Berkeley Segmentation Dataset and Benchmark" (BSDB) and some other images which were earlier used by Dr. Sarif Kumar Naik during his stay at ISI. My heartfelt thanks to them.

I would also want to thank all my teachers who have taught me in the past, in school, in college and at ISI.

I would like to thank my friends for their constant support and encouragement.

And last but not the least, I consider myself blessed to have such caring parents and my brother. I would like to thank them for their constant support, motivation, encouragement, everlasting love and prayers which helped me sail through every phase of life.

Avisek Chakraborty
July 15, 2011

Edge Detection Methodologies for Color Images

Table of Contents

List of Figures and Images

CHAPTER 1

INTRODUCTION

1.1 Edge Detection

AN EDGE is characterized by an abrupt change in intensity indicating the boundary between two regions in an image. However, there is no universally accepted mathematical definition of an edge. It is a local property of an individual pixel and is calculated from the image function in a neighborhood of the pixel. Edge detection is a fundamental operation in computer vision and image processing. It concerns the detection of significant variations of a gray level image. The output of this operation is mainly used in higher-level visual processing like three-dimensional (3-D) reconstruction, stereo motion analysis, recognition, scene segmentation, image compression, etc. Hence, it is important for a detector to be efficient and reliable. In edge-detection, the input is an image (gray-scale or color) and the output is a binary image with the edge pixels and the non edge pixels.

For edge detection, different approaches have been followed, such as mathematical morphology, Markov random fields, surface models, or PDE. Surface fitting approach for edge detection has been adopted by several authors. Bergholm’s edge detector applies a concept of edge focusing to find significant edges. Detectors based on some optimality criteria are also developed. Statistical procedures are also adopted. Other approaches on edge detection include the use of genetic algorithms neural networks, the Bayesian approach, and residual analysis-based techniques. The most common method is still the derivative approach with linear filtering. Many derivative filters have been studied and used to compute the intensity gradient of gray-level images : Roberts, Sobel, or Prewitt operators, finite impulse response filters with a large kernel, such as Canny’s filters, first derivative of the Gaussian function. Some well-known edge detectors for gray-scale images are the Marr and Hildreth edge detector, Canny edge detector, Demigny edge detection filter and Paplinski edge detection filter.

Color images provide more information than gray scale images. Thus more edge information is expected from a color edge detector than a gray scale edge detector. Sometimes it is difficult to detect a low intensity edge between two regions in gray scale, but in color image, the clarity is more because, without being much different in intensity there can be a substantial difference in hue.However, it becomes more challenging when color images are considered because of its multidimensional nature. Also the gray values are partially ordered in gray scale images but in a color image this freedom is not there. One of the earliest color edge detectors is proposed by Navatia. Other color image edge detectors are due to Huckel, DiZenzo, Cumani and others.

1.2 Outline of the Report

The document primarily consists of 5 chapters. In Chapter 1 we discuss the definition of an edge, and hence define edge detection. Then we go on to discuss the basic principles of edge detection. We conclude the chapter by mentioning the names of several edge detectors for gray scale and color images. In Chapter 2 we discuss some well-known algorithms for edge detection in gray scale images.
We discuss the Canny framework followed and the Demigny 1-D filter and Paplinski's n-directional filtering. The omnidirectional edge detector for gray level images is introduced next. We build the foundation for describing our work and the motivation to extend omnidirectional edge detectors to color images. In Chapter 3 we state the problem under consideration. We discuss methods to find the gradient magnitude and gradient direction in a color image. We discuss the intuition for our proposed method followed by its description and the proposed algorithm. In Chapter 4 we show the results of our experiments on different types of images using different types of filters. We use natural as well as artificially generated images for our purpose. The report ends with Chapter 5 citing the references.

CHAPTER 2

EDGE DETECTION ALGORITHMS AND FILTERS

Some well-known edge detection algorithms and filters are discussed here :

2.1 Canny Edge Detector for Gray Scale Images

The Canny edge detector as proposed by John Canny, describes a computational approach to edge detection. The success of the Canny approach depends on the definition of a comprehensive set of goals for the computation of edge points. The Canny edge detector overcame the following drawbacks that were present in the edge detectors proposed prior to the Canny : prior to Canny it was thought difficult to detect edges with only the first derivative maxima test. Also, deciding the value of the threshold was considered difficult. Canny was successful in obtaining single pixel width connected edges. Canny's three criteria for edge detection quality were: good detection, good localization, and low multiplicity of false detections.

The Canny edge detector is widely considered to be the standard edge detection algorithm in the industry. It was first created by John Canny for his Masters thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. Canny saw the edge detection problem as a signal processing optimization problem, so he developed an objective function to be optimized. The solution to this problem was a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. The steps in the Canny edge detector are as follows:

1. Smoothen the image with a two dimensional Gaussian.
   A bivariate Gaussian function such as the following is used :

$$e^{[(\frac{x-i}{h})^2+(\frac{y-i}{h})^2]} \tag{1}$$

where h^2 is the variance. We also see that at $x = i, y = j,$ the value of the Gaussian is 1.

2. The gradient of the image is then computed. This shows changes in intensity, which indicates the presence of edges. This actually gives two results, the gradient in the x direction and the gradient in the y direction. Gradient operators are used to approximate derivatives. Gradient operators such as PREWITT and SOBEL are generally used.

PREWITT operator : Let the gray value of the pixel at location *(i, j)* in the given image be $f(i,j)$. Then the approximation of partial derivatives in the x-direction or the direction of columns denoted by $\frac{\partial f(i,j)}{\partial x}$, is defined as

$$\frac{\partial f(i,j)}{\partial x} = f(i+1, j) - f(i, j) \qquad (2)$$

Similarly, the approximation of partial derivative at pixel location *(i, j)* in the y-direction or direction of rows is denoted by $\frac{\partial f(i,j)}{\partial y}$, and is defined as,

$$\frac{\partial f(i,j)}{\partial y} = f(i, j+1) - f(i, j) \qquad (3)$$

Thus the filter masks are given by :

-1	+1

and

-1
+1

SOBEL Operator masks :

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

and

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

The gradient magnitude is then calculated using either of the following expressions :

$$Mag(i,j) \;=\; \sqrt{\frac{\partial f(i,j)}{\partial x^2}+\frac{\partial f(i,j)}{\partial y^2}} \qquad (4)$$

or

$$Mag(i,j) = \left|\frac{\partial f(i,j)}{\partial x}\right|+\left|\frac{\partial f(i,j)}{\partial y}\right| \qquad (5)$$

Either of the above two expressions for the magnitude could be used but whichever is used must be used throughout.

The expression for the gradient direction is given by :

$$\theta(i,j)=\arctan\left(\frac{\frac{\partial f(i,j)}{\partial y}}{\frac{\partial f(i,j)}{\partial x}}\right) \qquad (6)$$

3. <u>Non-maximal suppression</u> : Edges will occur at points the where the gradient is at a maximum. Therefore, all points not at a maximum should be suppressed. In order to do this, the magnitude and direction of the gradient is computed at each pixel. Then for each pixel it is checked if the magnitude of the gradient is greater at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress the pixel intensity.

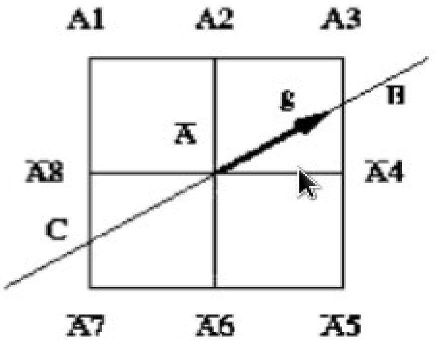

Fig. 1.

Gradient magnitude at pixel A is checked for being maxima in a $3x3$ *neighborhood during nonmaxima suppression phase. The arrow mark denotes the direction perpendicular to the edge direction.*

The steps are as follows :
i) Interpolate the magnitude of gradients at hypothetical pixels that lie along the direction perpendicular to the edge direction at pixel *(x, y)* in a $3x3$ neighborhood around it. The gradient vector at *(x, y)* is obtained using $f_x(x, y)$ and $f_y(x, y)$ as in the equations (1) and (2).
ii) If the magnitude of the gradient at *(x, y)* is not maximum among the interpolated magnitudes, then it is not an edge point.

4. Edge Thresholding : The method of thresholding used by the Canny Edge Detector is referred to as "hysteresis". It makes use of both a high threshold and a low threshold. If a pixel has a value above the high threshold, it is set as an edge pixel. If a pixel has a value above the low threshold and is the neighbor of an edge pixel, it is set as an edge pixel as well. If a pixel has a value above the low threshold but is not the neighbor of an edge pixel, it is not set as an edge pixel. If a pixel has a value below the low threshold, it is never set as an edge pixel.

5. Edge Tracking : The subprogram Track-edge() declares a pixel with gradient magnitude lying between upper and lower thresholds as edge pixel by checking its connectivity with already declared edge pixels. This forms a subpart of thresholding with hysteresis module and result in a smooth and connected edge map.

2.2 Demigny Filter

The main hypothesis of the Demigny approach is to work directly with signals in the sampled domain and to consider as input a pulse of a given width . The criteria are then defined and optimal filters for the width (i.e., for a given resolution) are computed. Demigny revisited the analytical expressions of the three Canny’s criteria for edge detection quality: good detection, good localization, and low multiplicity of false detections. However, Demigny gives the criteria for discrete sampled signals, i.e., for the real, implemented filters. Instead of a single-step edge as input signal, Demigny uses pulses of various width. The proximity of other edges affects the quality of the detection process. This is taken into account in the new expressions of these criteria. Then an optimal filter for each of the criteria and for any combination of them is derived. In particular, an original filter which maximizes detection and localization and a simple approximation of the optimal filter for the simultaneous maximization of the three criteria is defined. The upper bounds of the criteria are computed which will allow users to measure the absolute and relative performance of any filter (In [4], exponential, Deriche, and first derivative of Gaussian filters is evaluated). The criteria can also be used to compute the optimal value of the scale parameter of a given filter when the resolution of the detection is fixed.

The optimal filter designed by Demigny for combination of all the three criteria can be mathematically expressed as :

$$h_k = Aq^k \cos(k\theta) + Bq^{-k} \cos(k\theta) + Cq^k \sin(k\theta) + Dq^{-k} \sin(k\theta) + F \qquad (7)$$

where A, B, C, D, F are evaluated from some boundary conditions.

2.3 Paplinski's n-directional filtering

In [5], Paplinski proposed complex-valued filtering for simultaneous edge magnitude and direction detection. Since a complex number, $z = re^{j\theta}$, embeds both a magnitude r and a direction θ in itself, complex valued filtering is a suitable choice for directional edge detection.

We assume that we want to detect edges in some direction φ , $0 \leq \varphi \leq 2\pi$, and our optimal 1-D filter is s(r) .Then the two-dimensional complex-valued filter is specified as :

$$h_{\boldsymbol{\varphi}} \,\mathbf{(}z = re^{j\theta}) = s(r\ )\ p(\theta - \varphi) \qquad (8)$$

where p(θ) is a real valued function which attenuates as θ deviates from direction zero.
Specifically, $p(\theta\ ) = \exp\ (-\ c\theta^2\ )$, where c is specified by the user. To detect edges in n prespecified directions φ1 ,φ2 ,...,φn simultaneously, the following filter has been suggested:

$$h(z = re^{j\theta}) = \sum_{k=1}^{n} h_{\varphi_k}(r\,e^{j\theta})\,e^{j\varphi_k} \qquad (9)$$

Paplinski [5] suggested n-directional filtering as a generalization of gradient-based methods in which gradients are computed in n, rather than just two, directions. Specifically, he suggested the use of tri-directional filtering. One major drawback of n-directional filtering is that it is biased towards the pre-specified directions. In [8], this drawback is remedied by introducing omnidirectional filtering. The omnidirectional filter is not biased to any direction and clearly is a generalization of Paplinski's idea with $n = \infty$.

2.4 Omnidirectional Edge Detection for Gray Level Images

The authors in [8] proposes a new method for extending 1-D step edge detection filters to two dimensions via complex-valued filtering. Complex-valued filtering allows to obtain edge magnitude and direction simultaneously. The method can be viewed either as an extension of n-directional complex filtering of Paplinski to infinite directions or as a variant of Canny's gradient-based approach. In the second view, the real part of our filter computes the gradient in the x-direction and the imaginary part computes the gradient in the y direction. Paplinski claimed that n-directional filtering is an improvement over the gradient-based method, which computes gradient only in two directions. However the authors in [8] show that the omnidirectional and Canny's gradient-based extensions of the 1-D DoG coincide. In contrast to Paplinski's claim, this coincidence shows that both approaches suffer from being confined to the subspace of two 2-D filters, even though n-directional filtering hides these filters in a single complex-valued filter. Aside from these theoretical results, the omnidirectional method has practical advantages over both n-directional and gradient-based approaches. The experiments performed on synthetic and real world images show the superiority of omnidirectional and gradient-based methods over n-directional approach. In comparison with the gradient-based method, the advantage of omnidirectional method lies mostly in freeing the user from specifying the smoothing window and its parameter. Since the omnidirectional and Canny's gradient-based extensions of the 1-D DoG coincide, the experiments on extending the 1-D Demigny filter. This filter has been proposed by Demigny as the optimal edge detection filter in sampled images.

The 2-D filter is denoted by *f (r,* θ *).*

As the filter should not be biased towards any direction, it satisfies the following condition :

$$f(r,\theta)=|f(r,\theta)|\mathrm{e}^{jarg(f(r,\theta))} \qquad (10)$$

The authors goes on to prove that as $|f(\theta)|$ *= 1,* so *f (* θ *) =* $\mathrm{e}^{j\theta}$ and the omnidirectional filter is given by :

$$f(r,\theta) = f(r)\ \mathrm{e}^{j\theta} \qquad (11)$$

The authors then prove that the only omnidirectional complex filter satisfying all the conditions is

$$f(r,\theta)=f(r)\mathrm{e}^{j\theta} \qquad (12)$$

where *f(r)* is the given 1-D filter .

2.5 Edge Detection in Color Images

The expressions for gradient magnitude and direction in a gray-scale image cannot be applied straightforward to each component in a color image because the intensity of a pixel in a color image is a vector. The expressions used in [6] for finding the gradient magnitude and direction are as follows :

$$\lambda = \frac{(E+G)+\sqrt{(E-G)^2+4\mathrm{F}^2}}{2} \qquad (13)$$

and

$$\theta=\frac{1}{2}\arctan(\frac{2\mathrm{F}}{E-G}) \qquad (14)$$

where λ is the gradient magnitude and θ is the gradient direction and,

$$E = f_x^R f_x^R + f_x^G f_x^G + f_x^B f_x^B \ , \tag{15}$$

$$F = f_x^R f_y^R + f_x^G f_y^G + f_x^B f_y^B \ , \tag{16}$$

$$G = f_y^R f_y^R + f_y^G f_y^G + f_y^B f_y^B \ . \tag{17}$$

where f_x^C and f_y^C are the partial derivative along x and y directions respectively for each of the color components red(R), green(G) and blue(B) in the RGB color model. *(Details discussed in Section 3.3)*

The authors in [6] also use the statistical variability of partial derivatives at each pixel is used to obtain standardized edge magnitude and is thresholded using two threshold values. The advantage of standardization is evident from the results obtained. The efficiency of many image processing and computer vision tasks depends on the perfection of detecting meaningful edges. Many algorithms reported in the literature prior to this adopted ad hoc schemes for this purpose. These algorithms require the threshold values to be supplied and tuned by the user. There are many high-level tasks in computer vision which are to be performed without human intervention. In such cases automatic computation of the thresholds is very useful. Thus there is a need to develop a scheme where a single set of threshold values would give acceptable results for many color images.

However no omnidirectional filter for color images has been developed so far.

CHAPTER 3

THE PROBLEM

3.1 Motivation

It follows from the discussions before that no omnidirectional edge detector for color images has been developed so far. The problem in hand is to develop an omnidirectional edge detector for color images and to compare the results with already existing edge detectors for color images.

However we have an omnidirectional edge-detector for gray-scale images [8]. Also we have several methods for evaluating the gradient magnitude and direction in color imgaes for example in [6] and [7]. So our intuition was based on the basic steps of the omnidirectional edge detector in [8] and the evaluation of the gradient magnitude and direction in color images as in [6] and [7]. The task is therefore to extend them to color images.

Also an omnidirectional edge detector promises to provide better edge maps as it detects edges not being biased in any directions unlike edge detectors based on calculating gradients in two directions only. Moreover as claimed by the authors in [8], the omnidirectional operator developed for gray scale images frees the user from specifying the smoothing window and its parameters. The chief motivation for working towards an omnidirectional edge detector for color images is carrying forward the advantages obtained in using omnidirectional filters for gray scale images to color images.

3.2 The Color Image Model

We have used color images described in the RGB Color Model. In the RGB model, each color appears in its primary spectral components of red, green and blue. So intensity at a pixel is now a vector composed of the red, green and blue components as shown in the following figure, instead of a single value as in gray-scale images.

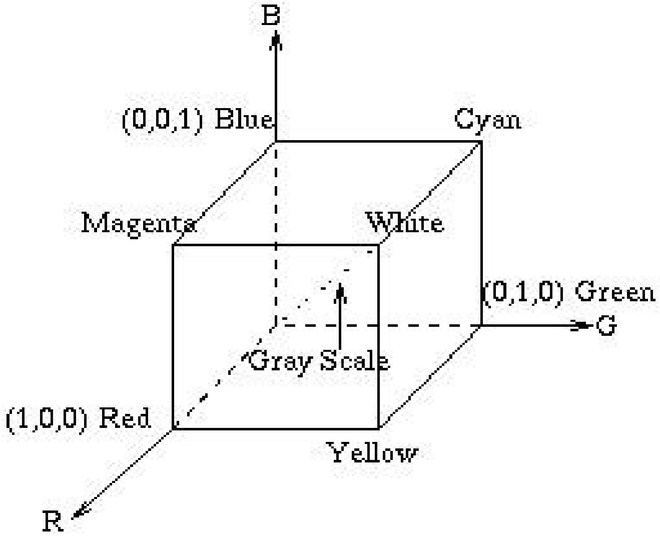

Fig. 2
Gradient magnitude at pixel A is checked for being maxima in a $3x3$ neighborhood during nonmaxima suppression phase. The arrow mark denotes the direction perpendicular to the edge direction.

3.3 Gradient Magnitude and Gradient Direction in Color Images

We have already seen that finding the gradient magnitude and direction is a very important step in edge detection using derivative operators. However unlike in gray-scale images, for color images we have to compute the gradient direction and magnitude in the color vector space. The gradient computed for gray-scale images cannot be directly extended to color images as it is not defined for vector quantities. Thus computing the gradient on individual images and then using the results to form a color image will lead to erroneous results. In a gray scale image, edges are detected by detecting the discontinuities in the image surface i.e. the discontinuities in the intensity of a sequence of pixels in a particular direction called gradient direction. The discontinuities in gray scale is easy to determine because gray values are partially ordered, but in a color image this freedom is not there. The simple difference between color vectors does not give the true distance between them. The edge magnitude λ of f at a pixel (x, y) is given by the largest eigenvalue of the 2 × 2 matrix,

$$A = \begin{bmatrix} E & F \\ F & G \end{bmatrix}$$

and the direction of maximum contrast is given by the corresponding eigen vector.

In the above expression,

$$E = f_x^R f_x^R + f_x^G f_x^G + f_x^B f_x^B ,$$

$$F = f_x^R f_y^R + f_x^G f_y^G + f_x^B f_y^B ,$$

$$G = f_y^R f_y^R + f_y^G f_y^G + f_y^B f_y^B .$$

The characteristic equation of the matrix A above is given by,

$$A – \lambda I = 0$$

which when solved gives,

$$(E - \lambda)(G - \lambda) - F^2 = 0$$

$$\text{or, } \lambda^2 - \lambda (E + G) + EG – F^2 = 0$$

The gradient magnitude is given by the highest eigen vector is given by,

$$\lambda = \frac{(E+G)+\sqrt{(E-G)^2+4F^2}}{2}$$

and the direction is given by the direction of the highest eigen vector,

$$\theta=\frac{1}{2}\arctan\left(\frac{2F}{E-G}\right)$$

3.4 Proposed Method

We extend the omnidirectional filter to color images :

3.4.1 Details of the Methodology

The omnidirectional methodology in [8] can be fit into the standard framework of the Canny edge detector. The extension to color image edge detection can be designed on similar lines. The input to the edge detector is the color image I_{inp} of size m x n, threshold values T_1 and T_2 and window size W. The omnidirectional filter of the required window size is then generated. The omnidirectional filter is applied to each component of the color image i.e the red, green and blue components. As discussed in [8] complexed-valued filtering is employed here. The output of the filter hence has a real part and an imaginary part. The real part of the output gives the gradient of the component image in the x-direction whereas the imaginary part gives the gradient of the component image in the y-direction. From the partial derivatives of the component images the gradient magnitude and the gradient direction is calculated using equations (13) and (14) in Section 2.5 and later discussed in details in Section 3.3. From here on, as in [8] for gray scale images, the algorithm takes the course of the standard Canny framework. The gradient magnitude is used for non-maxima suppression as discussed next. Next using the thresholds, hysteresis thresholding is performed followed by Edge-tracking.

3.4.2 Algorithm for the Proposed Methodology

Input : Color image I_{inp} of size m x n, threshold values T_1 and T_2 and window size W.
Output : Binary edge map with edge pixels and non-edge pixels.

1) Input the color image I_{inp} of size m x n. Also input the two threshold values T_1 and T_2 and the window size W.

2) Apply the omnidirectional filter as discussed in Chapter 2 on each component of the input image $\boldsymbol{I}_{inp}^{C}$. The omnidirectional filter is a complex-valued filter. The real component of the output gives the gradient of the image in the x-direction whereas the imaginary component of the output gives the gradient in the y-direction. Hence find the partial derivatives $f_x^C(i, j)$ and $f_y^C(i,j)$ of each component image, where C stands for each of the component R (Red), G (Green) and B (Blue).

```
3) for $i = 1, 2, \ldots, m$
        for $j = 1, 2, \ldots , n$
                calculate the gradient magnitude $\lambda$ and gradient direction $\theta$ as
                discussed in equations (13) and (14).
```

4) Define two matrices I_{temp} and I_{out} both of size $m x n$, with all the elements equal to zero.

```
5) for $i = 1, 2, \dots, m$
       for $j = 1, 2, \dots, n$
           apply nonmaxima suppression algorithm using $\lambda$ and $\theta$. If the pixel $(i, j)$ is
           not suppressed using non-maxima suppression then, apply Hysteresis
           Thresholding as follows :

    If $\lambda > T_2$,
        then $I_{temp}(i, j) \leftarrow 1$
    else if $\lambda > T_1$,
        then $I_{temp}(i, j) \leftarrow 2$
    else
        $I_{temp}(i, j) \leftarrow 0$
```

```
6) for $i = 1, 2, \dots, m$
       for $j = 1, 2, \dots, n$
           if $I_{temp}(i, j) = 2$
               then Track-edge$(i, j)$.
```

The subprogram Track-edge() declares a pixel with gradient magnitude lying between upper and lower thresholds as edge pixel by checking its connectivity with already declared edge pixels. This forms a subpart of thresholding with hysteresis module and result in a smooth and connected edge map. The main steps of the algorithm for this is given as follows.

```
Track-edge$(i, j)$
If $I_{out}(i, j) \neq 1$, then
    1) $I_{out}(i, j) \leftarrow 1$.
    2) for $a = i-1, i, i+1$
           for $b = j-1, j, j+1$
               if $(a, b) \neq (i, j)$ and (a, b) doesn't belong to the set of considered
               pixels for $(i, j)$ and $I_{temp}(i, j) > 0$, then Track-edge$(a, b)$.
```

Nonmaxima Suppression:

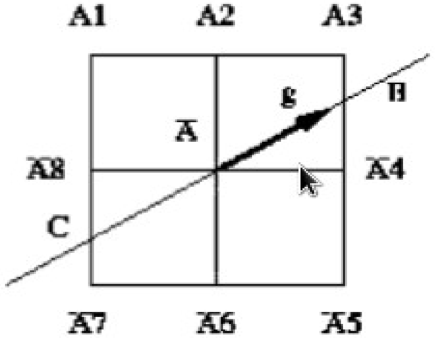

Fig. 3.
Gradient magnitude at pixel A is checked for being maxima in a $3x3$ *neighborhood during nonmaxima suppression phase. The arrow mark denotes the direction perpendicular to the edge direction.*

1) Interpolate the magnitude of gradients at hypothetical pixels that lie along the direction perpendicular to the edge direction at pixel *(x, y)* in a $3x3$ neighborhood around it. The gradient vector at *(x, y)* is λ and the gradient direction is θ obtained using equations (13) and (14) in Section 2.5 and also discussed in Section 3.3

2) If the magnitude of the gradient at *(x, y)* is not maximum among the interpolated magnitudes, then it is definitely not an edge pixel.

CHAPTER 4

RESULTS AND COMPARISONS

A method has been proposed in Chapter 3 to extend the omnidirectional edge detection filter to color images. This has been implemented on natural images as well as on artificially created images. The concentric circles image is an artificially created image We also show here the result of applying the standard Canny edge detector, omnidirectional extension of the Demigny and DOG filters on gray scale images as well. We then show the result of applying our extension of omnidirectional filters that extend the DOG and Demigny 1D filters to color images and also the results of implementations in [6]. All the images are thresholded with low threshold value 25th percentile of the gradient magnitude and high threshold 75th percentile of the gradient magnitude. In all the results below W indicates the window size and σ is the variance for Gaussian.

The list of images used and the corresponding results are mentioned in a separate table at the beginning of the report.

4.1 Results on gray-scale images

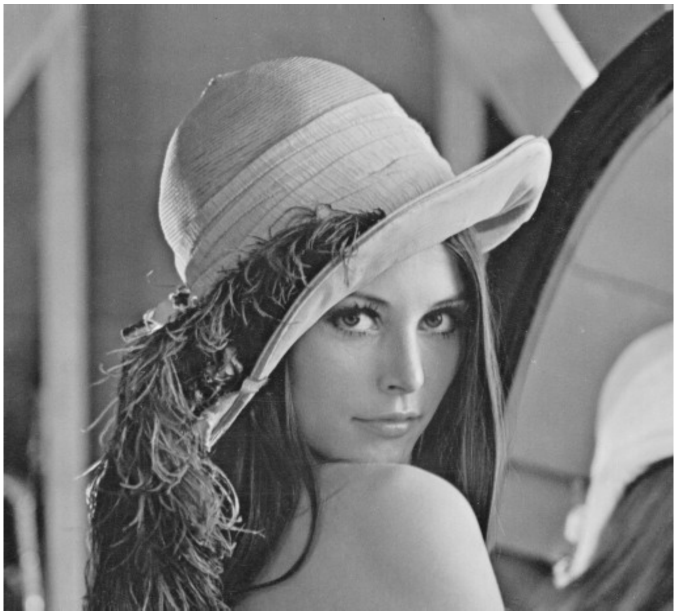

Fig. 4 (a)
Original LENA Image (gray-scale) (original size 512 x 512, re-sized here in the report document)

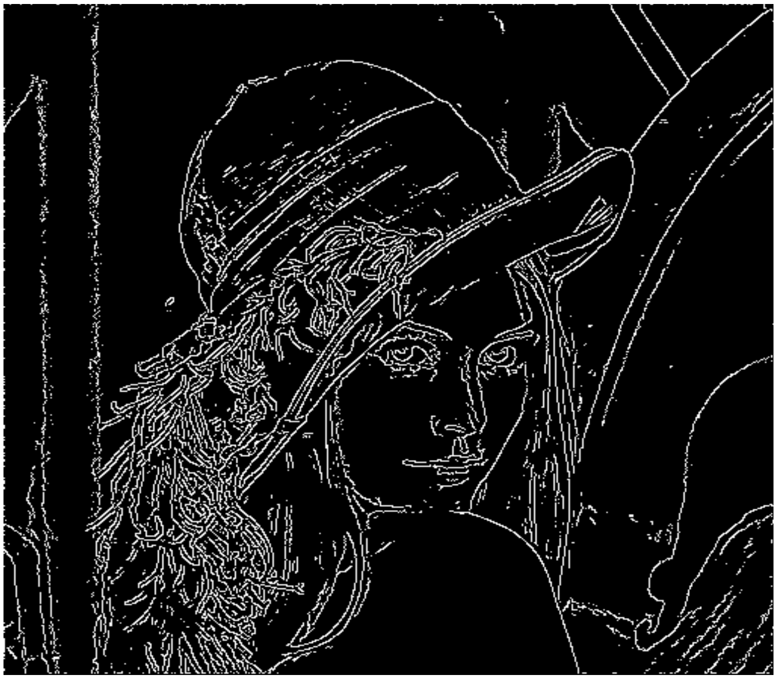

Fig. 4 (b)

Result of applying Canny edge detector on the LENA image (gray scale) with Gaussian smoothing window size=5 and $\sigma = 1$

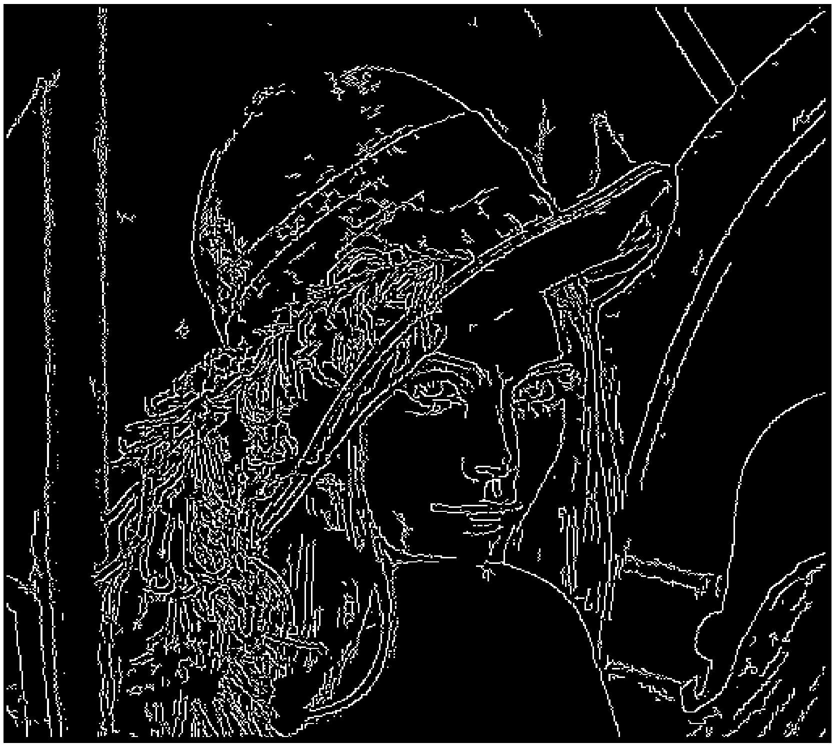

Fig. 4 (c)

Result of applying the omnidirectional extension of the DOG filter with W=5 and $\sigma = 1$ *on the LENA image (gray scale)*

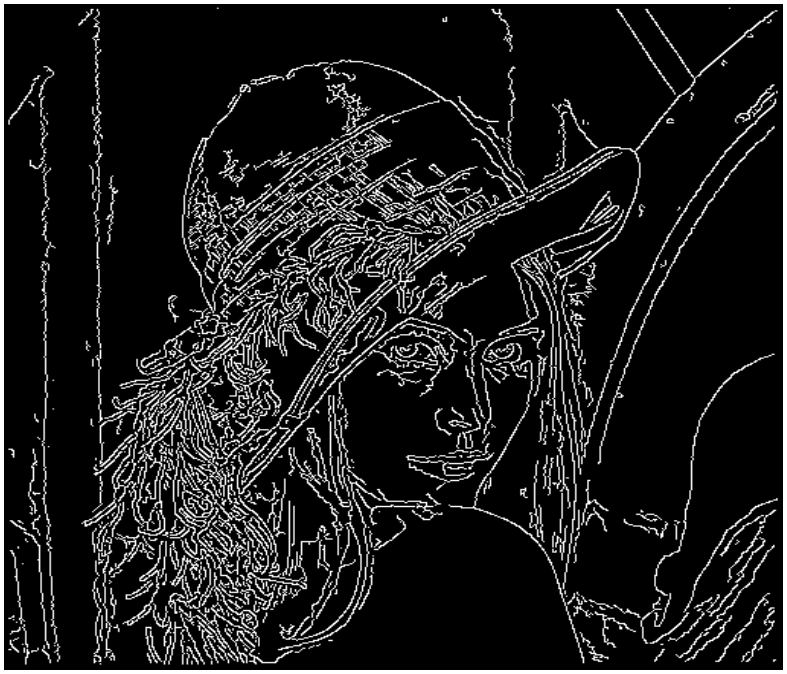

Fig. 4 (d)
Result of applying the omnidirectional extension of the Demigny filter with W=5 on the LENA image (gray scale)

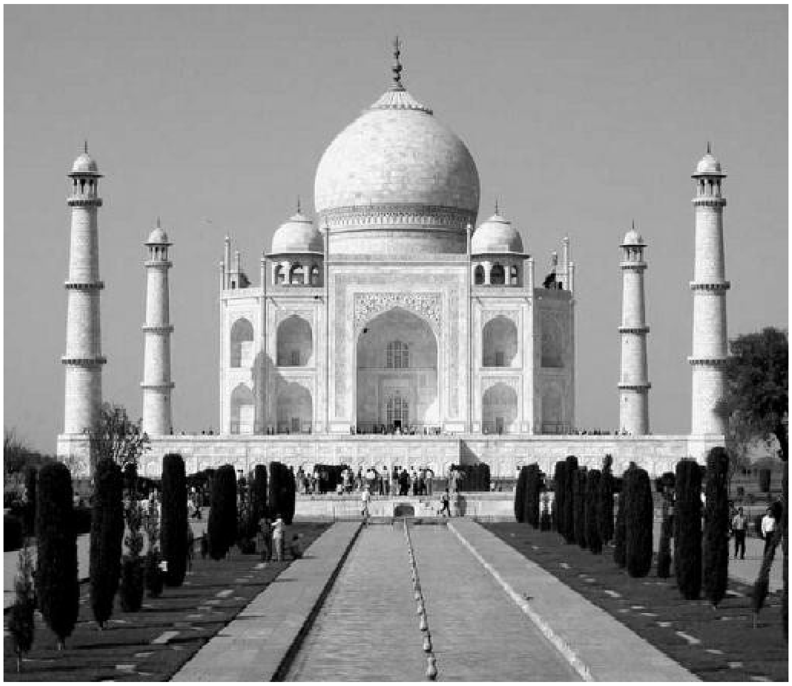

Fig. 5 (a)
Original TAJ MAHAL Image (gray-scale) (original size 512 x 512, re-sized here in the report document)

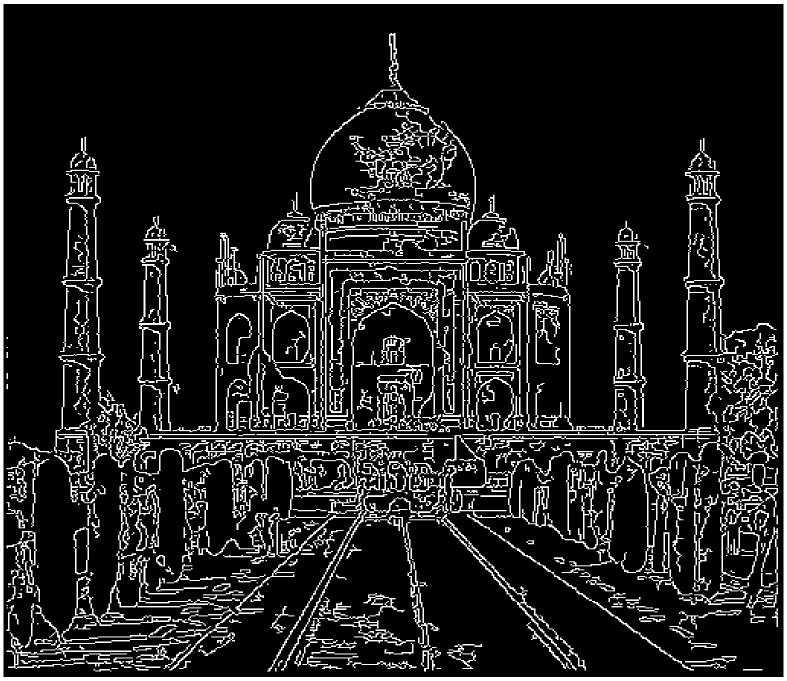

Fig. 5 (b)

Result of applying Canny edge detector on the TAJ MAHAL image (gray scale) with Gaussian smoothing window size=5 and σ *= 1*

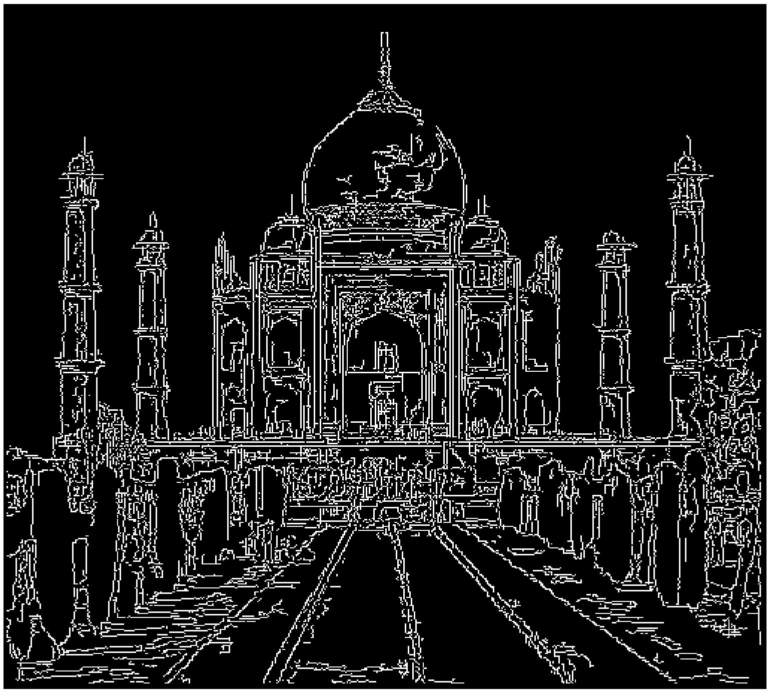

Fig. 5 (c)

Result of applying the omnidirectional extension of the DOG filter with W=5 and σ *= 1 on the TAJ MAHAL image (gray scale)*

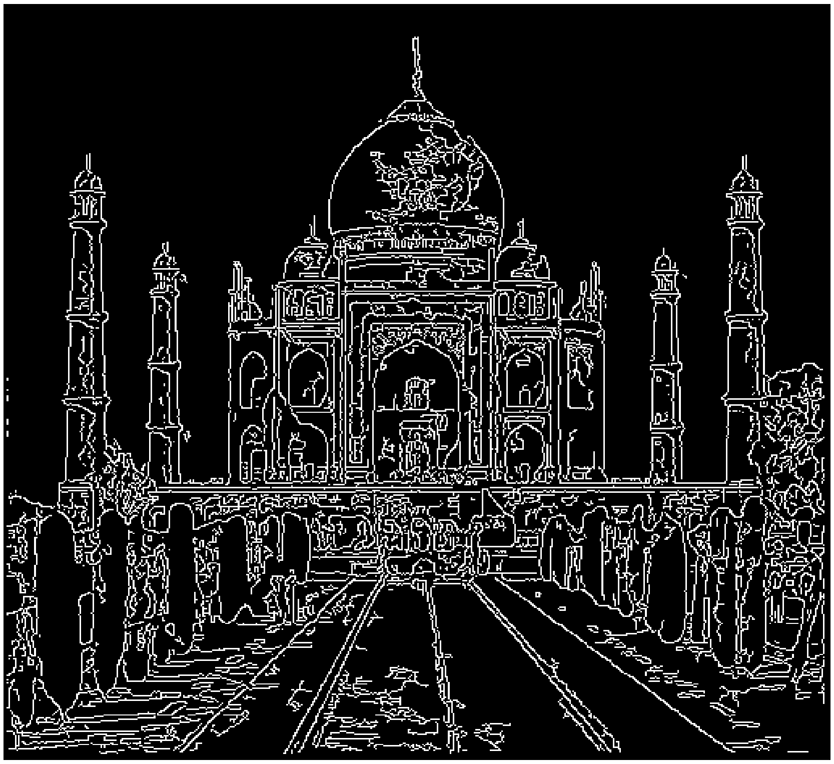

Fig. 5 (d)

Result of applying the omnidirectional extension of the Demigny filter with W=5 on the TAJMAHAL image (gray scale)

4.2 Results on color images

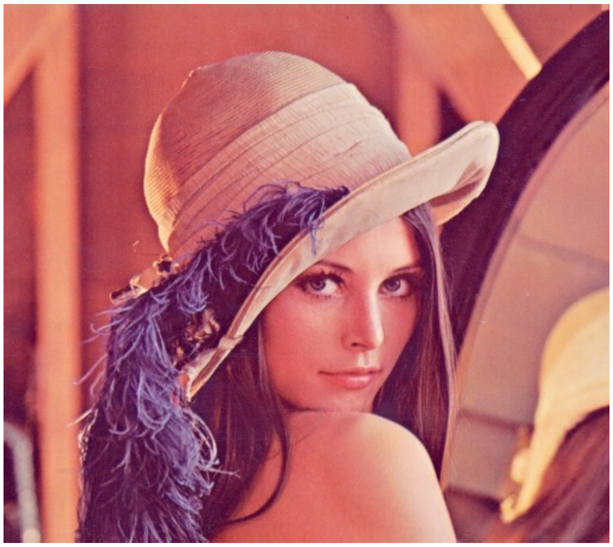

Fig. 6 (a)

Original LENA Image (color) (original size 512 x 512, re-sized here in the report document)

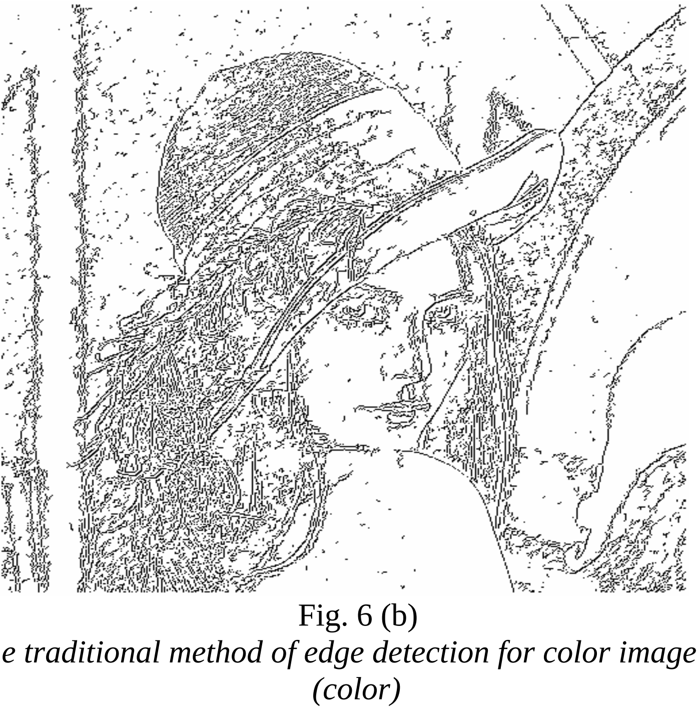

Fig. 6 (b)

Result of applying the traditional method of edge detection for color images on the LENA image (color)

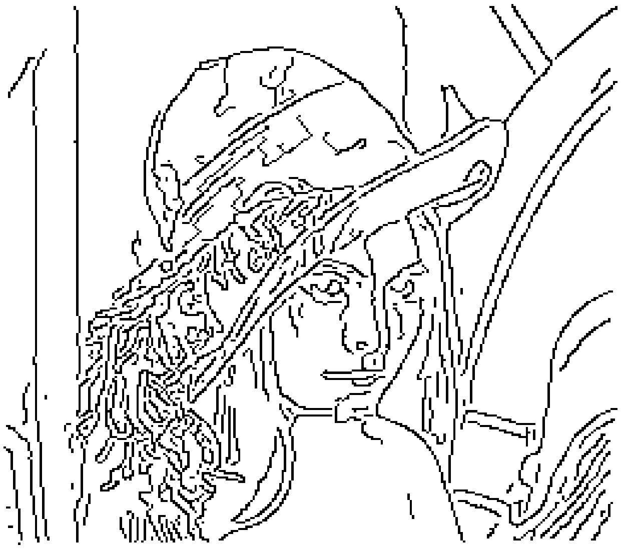

Fig. 6 (c)
Result of applying Sarif et al.'s method with h = 1.25, Tl = 5, Tu = 30 on the LENA image (color)

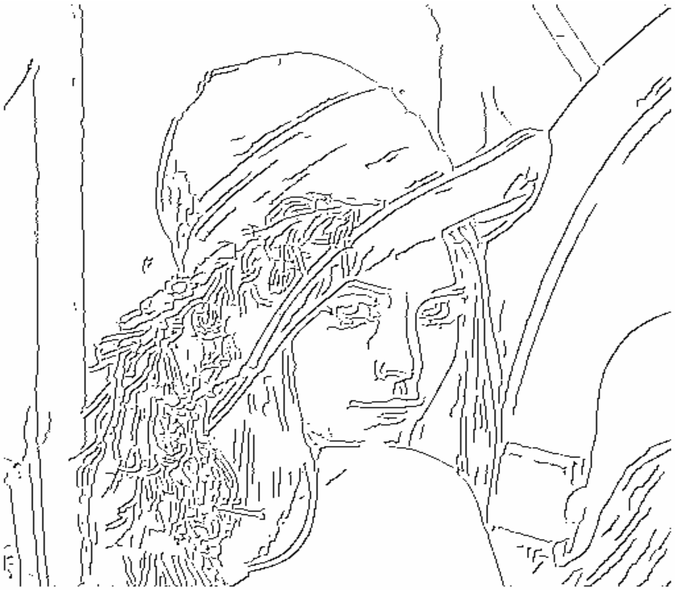

Fig. 6 (d)
Result of applying the proposed method (with the omnidirectional extension of the DOG filter to color images) with W=5 and σ = 1 on the LENA image (color)

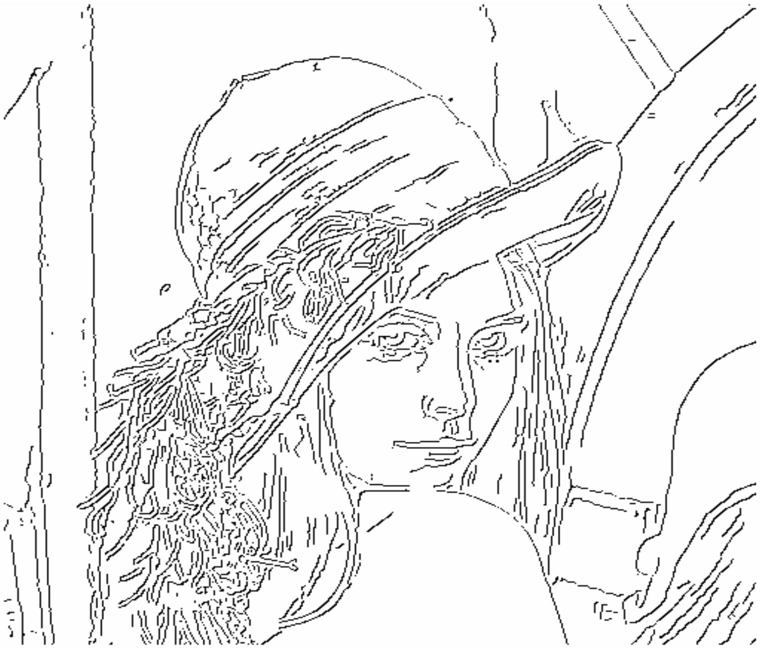

Fig. 6 (e)
Result of applying the proposed method (with the omnidirectional extension of the Demigny filter to color images) with W=5 on the LENA image (color)

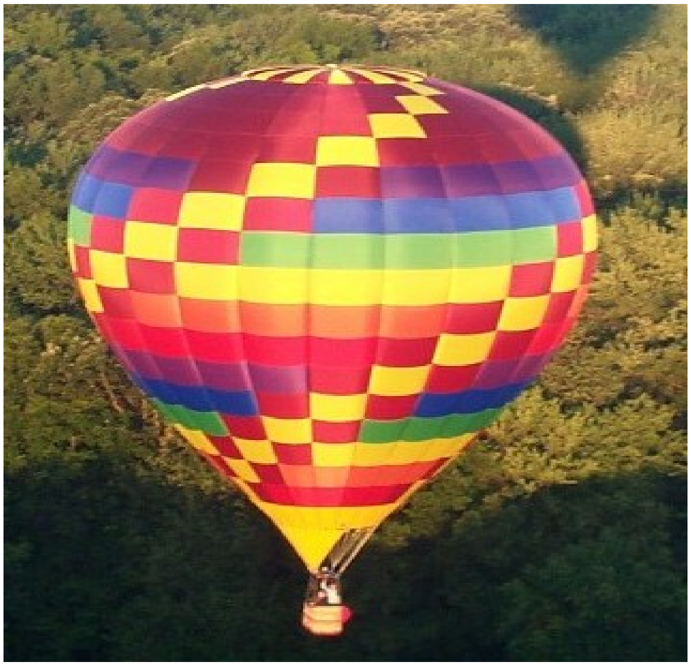

Fig. 7 (a)
BALLOON Image (color) (original size 320 x 373, re-sized here in the report document)

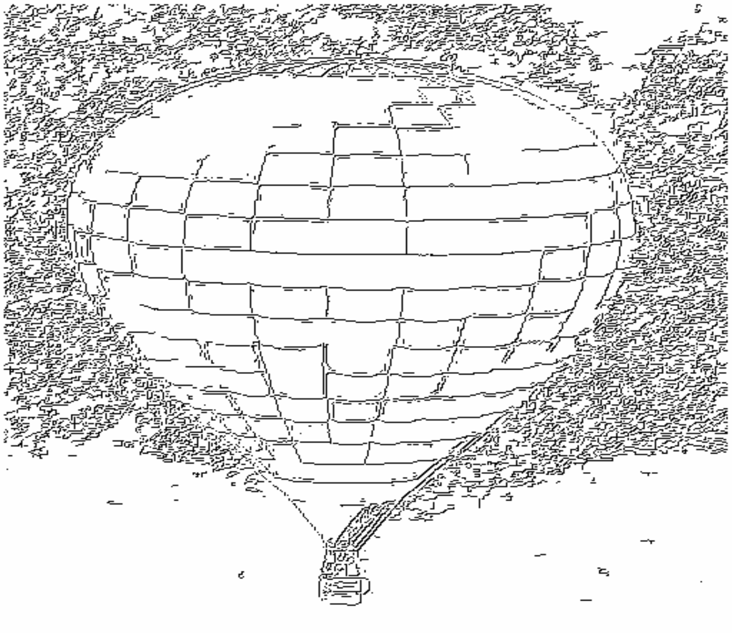

Fig. 7 (b)

Result of applying the traditional method of edge detection for color images on the BALLOON image (color)

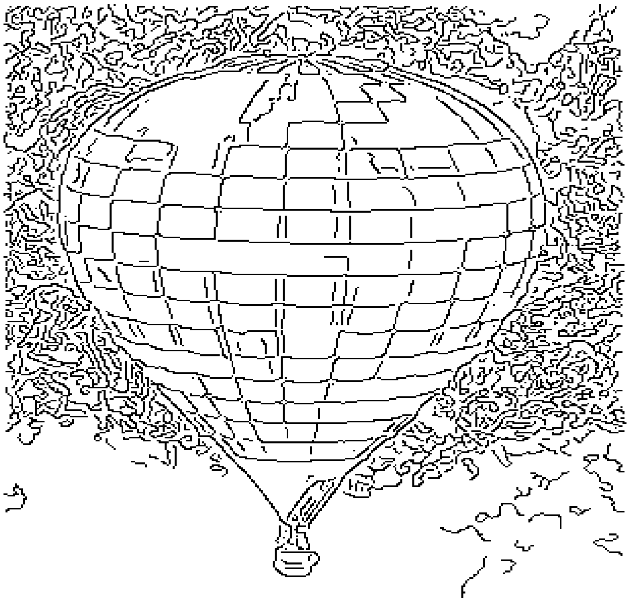

Fig. 7 (c)
Result of applying Sarif et al.'s method with h = 1.25, Tl = 5, Tu = 30 on the BALLOON image (color)

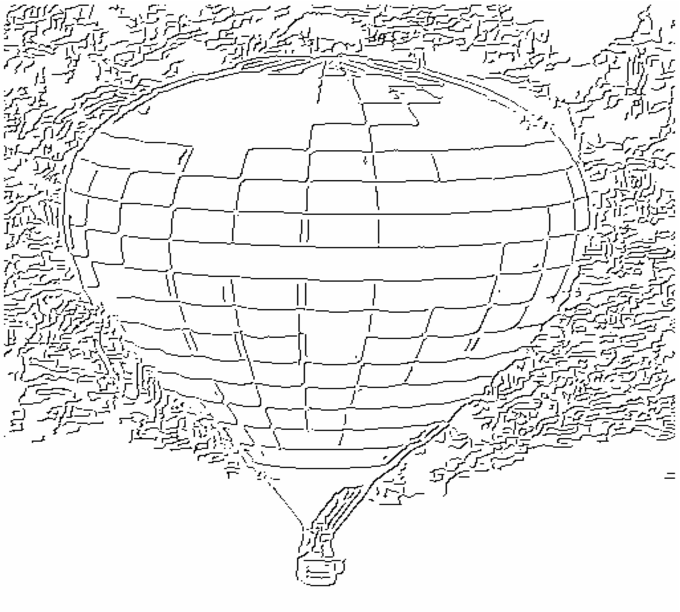

Fig. 7 (d)
Result of applying the proposed method (with the omnidirectional extension of the DOG filter to color images) with W=5 and σ = 1 on the BALLOON image (color)

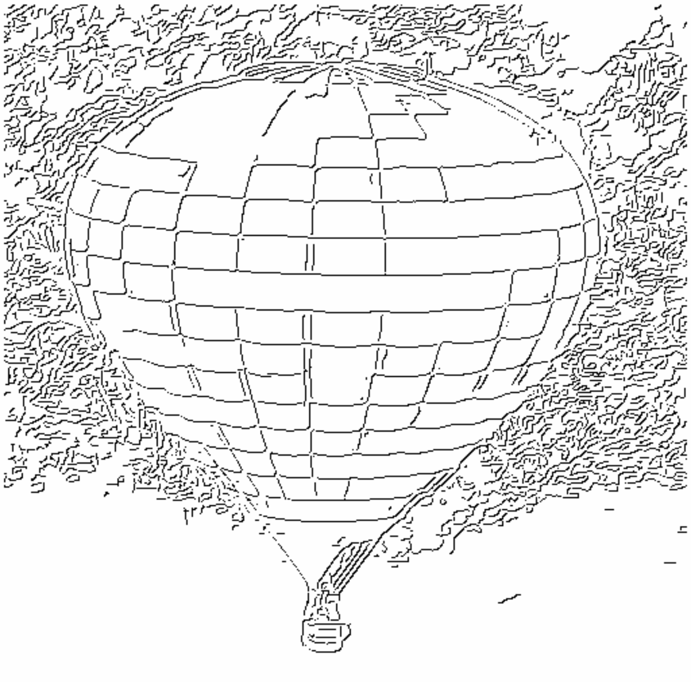

Fig. 7 (e)
Result of applying the proposed method (with the omnidirectional extension of the Demigny filter to color images) with W=5 on the BALLOON image (color)

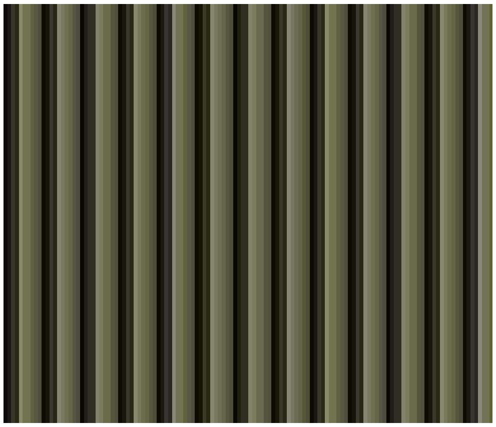

Fig. 8 (a)
VERTICAL BARS Image (color) (original size 128 x 128, re-sized here in the report document)

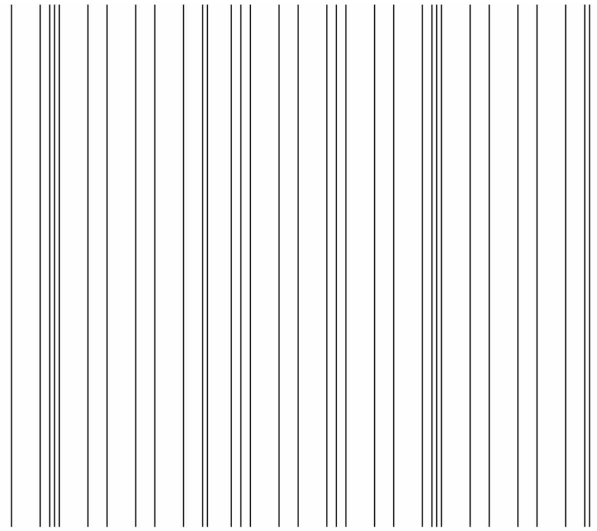

Fig. 8 (b)
Result of applying the traditional method of edge detection for color images on the VERTICAL BARS image (color)

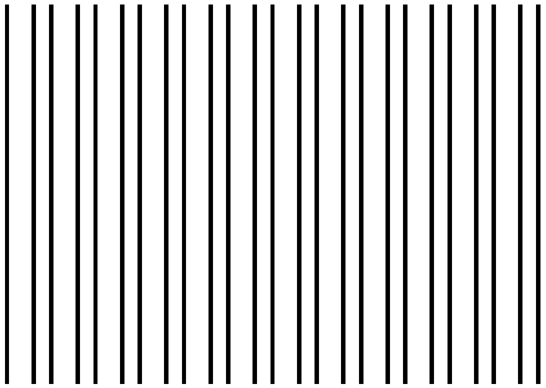

Fig. 8 (c)
Result of applying Sarif et al.'s method with h = 1.25, Tl = 5,
Tu = 30 on the VERTICAL BARS image (color)

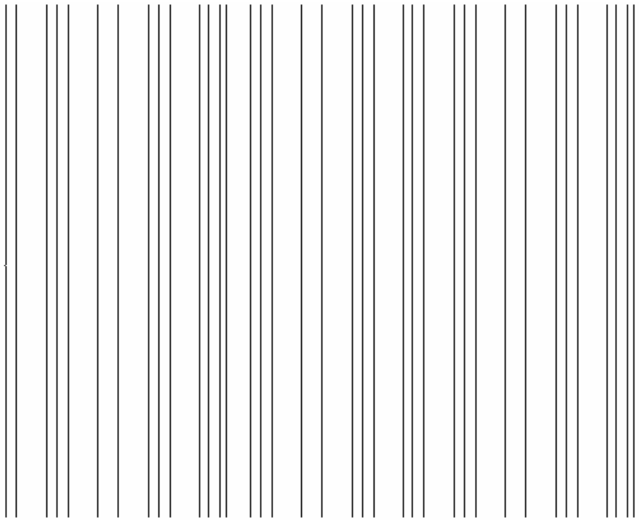

Fig. 8 (d)

Result of applying the proposed method (with the omnidirectional extension of the DOG filter to color images) with W=5 and σ *= 1 on the VERTICAL BARS image (color)*

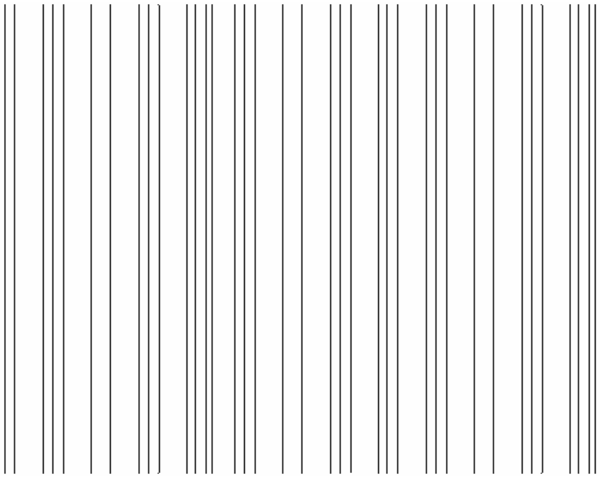

Fig. 8 (e)

Result of applying the proposed method (with the omnidirectional extension of the Demigny filter to color images) with W=5 on the VERTICAL BARS image (color)

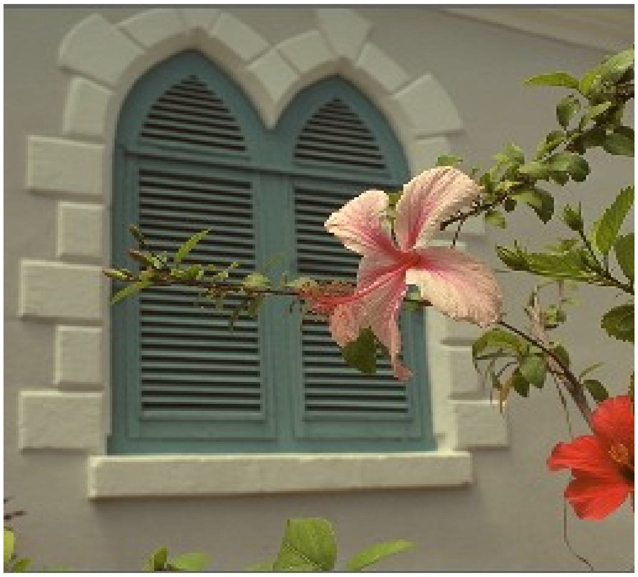

Fig. 9 (a)
WINDOW Image (color) (original size 256 x 256, re-sized here in the report document)

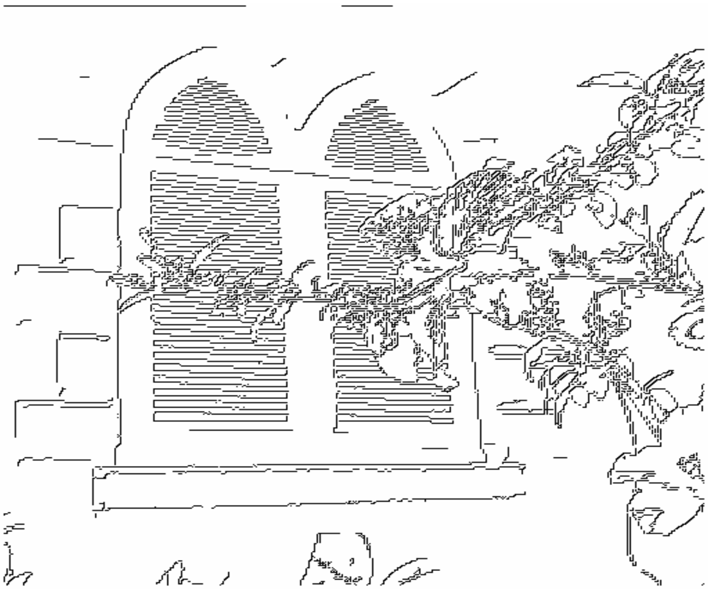

Fig. 9 (b)
Result of applying the traditional method of edge detection for color images on the WINDOW image (color)

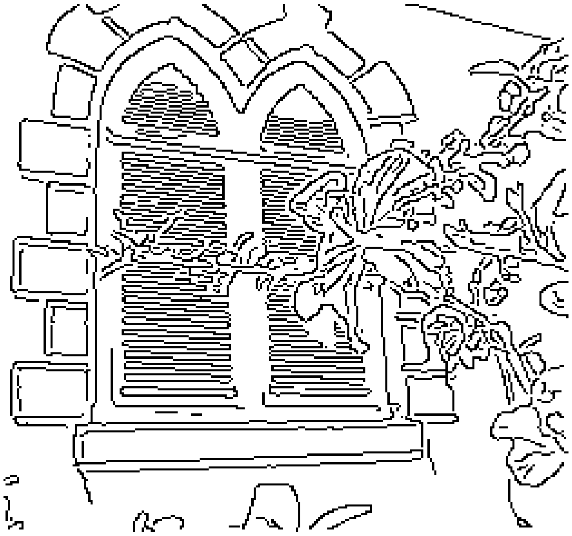

Fig. 9 (c)
Result of applying Sarif et al.'s method with h = 1.25, Tl = 5,
Tu = 30 on the WINDOW image (color)

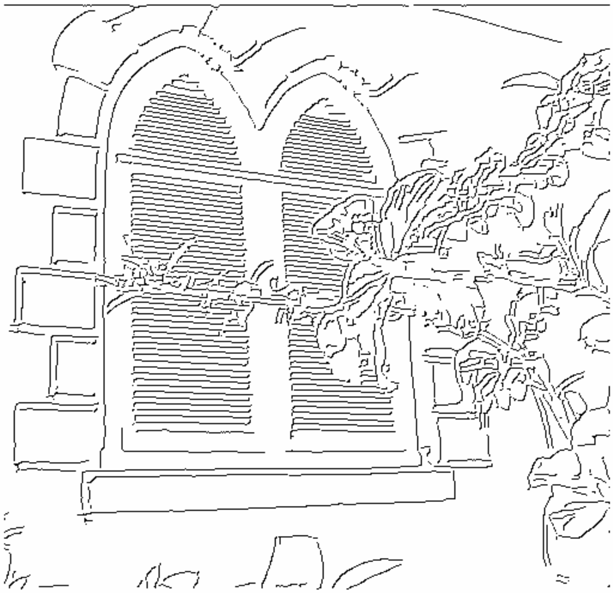

Fig. 9 (d)
Result of applying the proposed method (with the omnidirectional extension of the DOG filter to color images) with W=5 and σ *= 1 on the WINDOW image (color)*

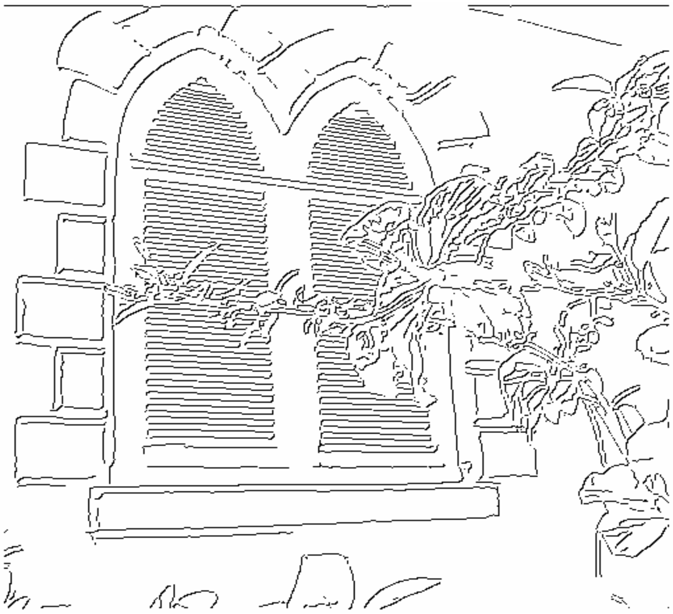

Fig. 9 (e)

Result of applying the proposed method (with the omnidirectional extension of the Demigny filter to color images) with W=5 on the WINDOW image (color)

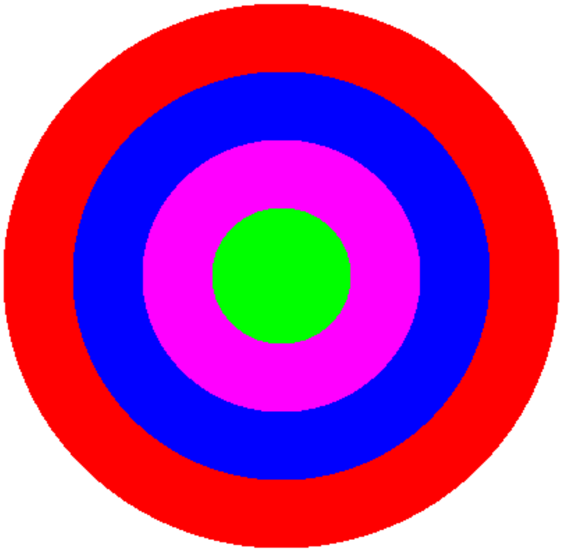

Fig. 10 (a)
CIRCLES Image (color, artificially created) (original size 128 x 128, re-sized here in the report document)

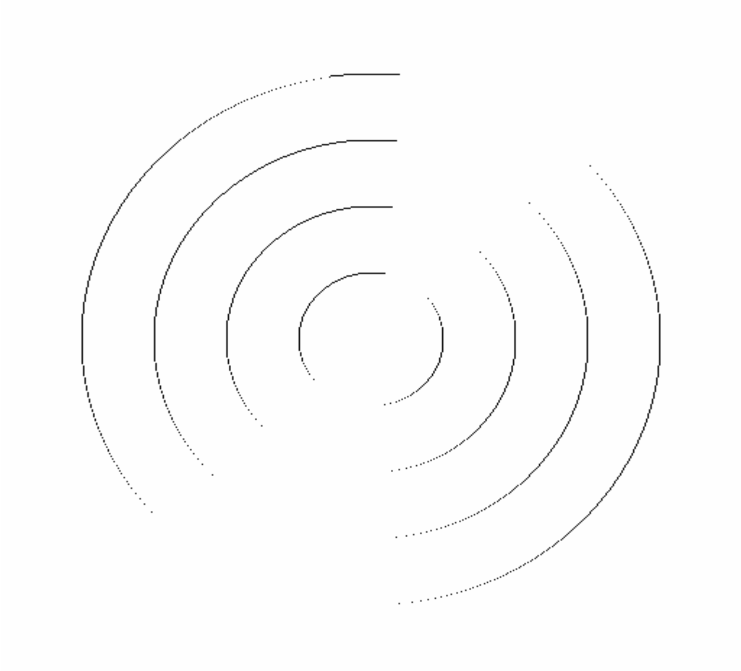

Fig. 10 (b)
Result of applying the traditional method of edge detection for color images on the CIRCLES image (color)

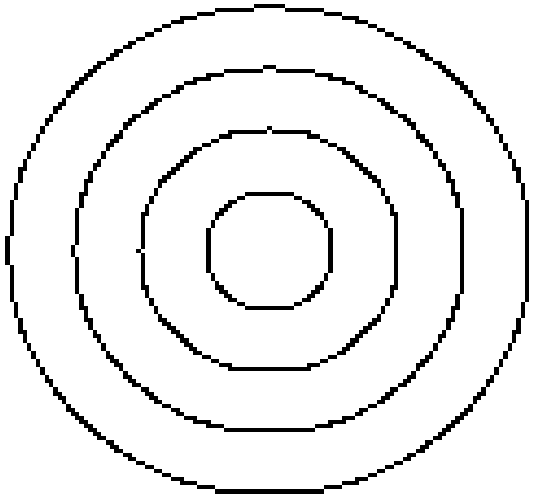

Fig. 10 (c)
Result of applying Sarif et al.'s method with h = 1.25, Tl = 5,
Tu = 30 on the CIRCLES image (color)

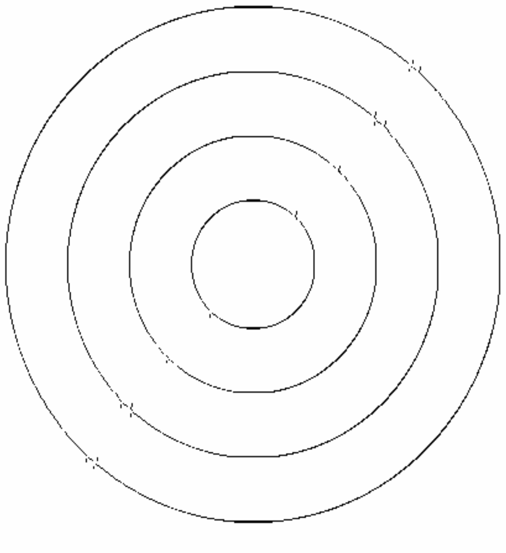

Fig. 10 (d)
Result of applying the proposed method (with the omnidirectional extension of the DOG filter to color images) with W=5 and σ = 1 on the CIRCLES image (color)

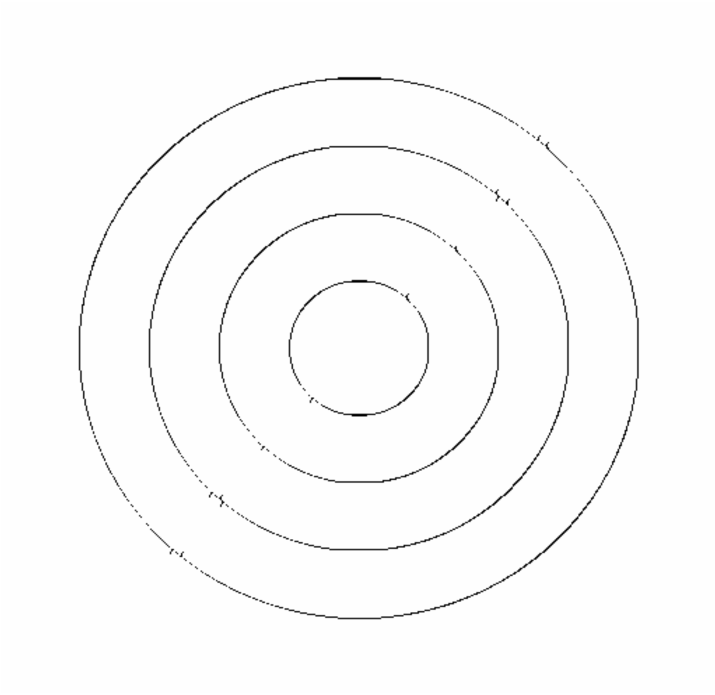

Fig. 10 (e)

Result of applying the proposed method (with the omnidirectional extension of the Demigny filter to color images) with W=5 on the CIRCLES image (color)

4.2 Discussions

First, we compare and study the results of the omnidirectional operator on gray scale images with that of the results of Canny's gradient based approach. It is evident that the omnidirectional approach is better than Canny's gradient based approach. The omnidirectional extension of the DOG filter detects more true edges and less spurious edges than Canny's gradient based approach. The omnidirectional extension of the Demigny filter in turn is better than the DOG extension in detecting more true edges and less spurious edges for both the Taj Mahal and the Lena image. Moreover, the omnidirectional operator has the advantage of automatically selecting the smoothing window and its parameters.

The proposed method along with other methods have been compared on several images color images as well. Methods used for comparisons are *one,* the traditional gradient operator masks used for calculating the partial derivatives followed by gradient magnitude and direction calculation for color images as discussed earlier ; *two,* the method proposed by Sarif et al. [6] ; *three*, our proposed extension of omnidirectional operator (extending the DOG filter) on color images; and *four*, our proposed extension of omnidirectional operator (extending the Demigny filter) on color images. It is clearly evident from the results that the omnidirectional extension (both DOG and Demigny) to color images is better than the traditional gradient based method. The results of applying them on the Lena (color) image testifies this fact. The omnidirectional extensions produces acceptable results in all cases. The best test for the omnidirectional operator is comparing the results obtained after applying all the above mentioned edge detectors on the artificially created concentric circles image. The traditional method misses edges in several directions. However the omnidirectional methods produces acceptable results and is able to detect edges in all directions. However, it must be borne in mind that the method proposed by Sarif et al . [6] produces better results for several images. The statistic used for thresholding in [6] can efficiently handle random noise present in the image. Also an intuitive reason for producing better results in [6] is that it increases the edge response of all the detected edge pixels but not necessarily uniformly.

4.4 Conclusion

The report describes an extension of omnidirectional edge detection filter for gray-level images to color images. At present, we don't have any omnidirectional edge detector for color images. The results and comparisons shows that our proposed omnidirectional filter for color images produces better results than ordinary gradient filter masks that computes gradient in the horizontal and vertical directions only. Comparing the results obtained after applying the edge detectors on the artificially generated concentric circles image, it is quite evident that the omnidirectional edge detectors (both DOG and Demigny extensions) are capable of detecting edges in all directions. The obtained results also show that the omnidirectional edge detectors produces results containing most of the important edges. Moreover as claimed by the the authors in [8], the other advantages of omnidirectional operator over other methods still hold. As claimed in [8], the advantage of the omnidirectional method lies mostly in freeing the user from specifying the smoothing window and its parameter. However when we compare our results with that of [6], the results produced after standardization of edge magnitude as done in [6] is better for several images.

Therefore we conclude that the proposed omnidirectional edge detector for color images produces quite acceptable and nice results for most of the images. Also it is able to automatically choose the smoothing parameters. However, comparing with the results of the method in [6] we see that there are several scopes for improvements. The statistic used for thresholding in [6] can efficiently handle random noise present in the image. Also an intuitive reason for producing better results in [6] is that it increases the edge response of all the detected edge pixels but not necessarily uniformly. So the omnidirectional operator can be improved on these lines and thus can be subjected to further study.

CHAPTER 5

REFERENCES

[1] R.C. Gonzalez and R.E. Woods, Digital Image Processing. Reading, MA: Addison-Wesley,1993.

[2] J. Canny, “A computational approach to edge detection”, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, pp. 679-697, June 1986.

[3] R. R. Rakesh, P. Chaudhuri, C. A. Murthy, “Thresholding in edge detection : A Statistical approach,” IEEE Trans. Image Processing, vol. 13, no. 7, pp. 927-936, July 2004.

[4] D. Demigny, "On optimal linear filtering for edge detection," IEEE Trans. Image Processing, vol. 11, no. 7, pp.728-737, Jul. 2002.

[5] A. P. Paplinski, "Directional filtering in edge detection," IEEE Trans. Image Processing, vol. 7, no. 4, pp. 611-615, April 1998.

[6] Sarif Kumar Naik and C. A. Murthy, “Standardization of Edge Magnitude in Color Images”, IEEE Trans. Image Processing, 2006

[7] Silvano Di Zenzo, “A Note on the Gradient of a Multi-Image”, IBM Rome Scientific Centre, Via del Giorgione 129, Rome, Italy

[8] Sayed Kamaledin Ghiasi Shirazi, Reza Safabakhsh, “Omnidirectional edge detection ”, Computer Vision and Image Understanding vol. 113, pp. 556–564, 2009