ADVANCEMENT IN COLOR IMAGE PROCESSING USING GEOMETRIC ALGEBRA

Biswajit Mishra, Peter Wilson and Bashir M. Al-Hashimi

Electronic Systems and Devices, University Of Southampton
Highfield, SO17 1BJ, Southampton, United Kingdom
phone: + (44) 238059 6665, fax: + (44) 238059 2901, email: {bm2,prw,bmah}@ecs.soton.ac.uk
web: www.ecs.soton.ac.uk

ABSTRACT

Due to the compact representation of vectors within *n* dimensional space Geometric Algebra (GA) is a preferred framework for signal representation and image representation. In this context the R, G, B color channels are not defined separately but as a single entity. As GA provides a rich set of operations, the signal and image processing operations becomes straightforward and the algorithms intuitive. From the experiments it is also concluded that the convolution operation with the rotor masks within GA belong to a class of linear vector filters and can be applied to image or speech signals. The usefulness of the introduced approach was demonstrated by analyzing and implementing two different types of edge detection schemes.

1. INTRODUCTION

Over the past few years, there has been a significant progress in color image processing [1] [2] [3]. Color signal representations also have undergone a sea change from being mere algebraic equations and matrix representation to more sophisticated vector valued representation. This is because the color image consists of three color planes in RGB and is inherently vector in nature. Therefore for color images the vector valued image processing techniques are more suitable. Though these techniques are crucial and appropriate, are not very popular because of the complexity and lack of a unified framework. They rely on scalar image processing functions based on monochrome techniques.

The RGB color vector has a magnitude and direction and hence can be represented as an entity having a luminance and chrominance component. However this is not a direct conversion and involves a transformation of one vector space (RGB) to another vector space (Hue-Saturation-Intensity or HSI) and the transformation is nonlinear in nature. Most of the conversions within the color space are lossy conversions due to this nonlinearity, so the algorithms are not portable and use a lot of vector additions and complex vector multiplications that also pose fundamental limitation.

In both gray scale and color images edge detection is one of the most basic operation in image processing. The edge in gray scale images is defined in an achromatic way as the discontinuity in the brightness function. The color edge is an extension to this *i.e.* to find the discontinuities along the adjacent regions of a color image in the 3D color space dependent on some function of metric distance of the color space, some function of merging of the derived edges and some function of imposing uniformity constraints to utilize all three independent edges concurrently. However, the fundamental problem associated with color edge detection is due to the uncorrelated monochrome techniques applied to the three correlated color channels. For example to smooth along a particular color component within the image, component-wise filtering will give an incorrect result. The reason is due to the nonlinear filtering of the smoothing operation. With component-wise filtering the median pixel resulting from each component-wise operation may not yield the same median pixel on each color plane (R, G or B) and hence the results may contain noise and/or incorrect pixel value. But with median filter [4], the operation picks the correct pixel because it chooses the correct median value of the pixel using the distance information as it treats the color as a 3D vector. There also exist alternative methods where vector methods are used for non linear filtering [5] operations which give better results and is preferred to component wise filtering. In this paper, a new method using Geometric Algebra is discussed which defines color as a single vector entity for color image processing. Based on it a linear filter is proposed which is applied to edge detection algorithms.

This paper is organized as follows: Section 2 discusses some previous work on vector filtering methods. Geometric Algebra (GA) is introduced in Section 3. In Section 4, the concepts of rotations and *rotors* in GA are discussed. In Section 5 the color is defined as a single entity as a vector consisting of the individual elements of the *multivector* defined within GA. The physical significance of the transformation of the color space is derived and discussed in Section 6. Two different edge detection techniques using GA are discussed in Section 7. Finally in Section 8, the results are summarized.

2. COLOR IMAGE PROCESSING AND VECTOR FILTERING

The standard technique to identify the critical features of the image in image processing applications is to use the rotational and curvature properties of the vector fields [6]. Therefore, the convolution and correlation techniques are quite common to image processing algorithms on scalar fields. These techniques have been extended to vector fields for visualization, signal analysis and applications such as detection of patterns within DNA images [2]. Independently in [7], the author developed a hypercomplex fourier transform of a vector field for image processing. This was extended in [8] [9] for image processing by GA methods. In [10] the authors have developed the GA or Clifford convolution fourier transform method for pattern matching on vector fields which are used for visualization of 3D vector field flows. It has been shown that the conventional feature detection fourier methods based on scalar techniques are simple extensions of fourier transforms based on GA. It has also been proven in these works that the GA convolution techniques are superior because of the unified notation for convolution of scalar, vector and *multivector* fields.

3. GEOMETRIC ALGEBRA IN 3D

This section describes the basics of GA in 3D. Expressions within GA embed and extend existing theories and methods to express geometric relations without the need for special case considerations in higher dimensions with simple yet powerful techniques [11] [12] [13].

Let the Euclidean vector space E^3 be defined by the orthonormal *basis* vectors e_1, e_2 and e_3. In GA this 3D space in E^3 can be decomposed into a linear space spanned by the following elements $1, e_1, e_2, e_3, e_1 \wedge e_2, e_2 \wedge e_3, e_3 \wedge e_1, e_1 \wedge e_2 \wedge e_3$ in G^3. The individual element of this linear subspace are called the *blades* and the elements of the algebra are multivectors. The *blade* signifies the subspace *i.e* scalar, vector or *bivector*. Hence the linear subspace consists of a scalar (1) a 0 grade *blade* element, vectors (e_1, e_2, e_3) 1 grade *blade* element, *bivector* ($e_1 \wedge e_2$, $e_2 \wedge e_3$, $e_3 \wedge e_1$) a 2 grade *blade* element (the directed area segment formed from sweeping e_1 along e_2 for $e_1 \wedge e_2$ which has orientation and is a 2-dimensional subspace.) and $e_1 \wedge e_2 \wedge e_3$ a *trivector* or the pseudoscalar. The collection of the subspaces define the complete space in G^3. The multiplication is associative, bilinear and is defined by the following rules:

$$\boldsymbol{1e}_i = \boldsymbol{1e}_i; \quad \boldsymbol{e}_i\boldsymbol{e}_i = \boldsymbol{1}; \quad \boldsymbol{e}_i\boldsymbol{e}_j = -\boldsymbol{e}_j\boldsymbol{e}_i$$

In GA it is possible to add different *grade* vectors to form a *multivector*. This reveals a great power within it by expressing the different *grade* vectors in a single product. For example, the *multivectors* in G^2 in E^2 would be the linear combination of the *blades* and will contain a scalar, vector and *bivector* parts:

$$\underbrace{a_0}_{\text{scalar}} + \underbrace{a_1e_1 + a_2e_2}_{\text{vector}} + \underbrace{a_3e_1 \wedge e_2}_{\text{bivector}}$$

In GA the multiplication of any two vectors (or multivector) $\boldsymbol{a}$ and $\boldsymbol{b}$ yields the geometric product given by: $ab = a \cdot b + a \wedge b$. which consists of the inner product(or dot product) and the outer product. It gives the information about the magnitude and orientation of the vector. For example, if a and b are collinear, then $a \wedge b = 0$ and the geometric product which now has only the dot product, gives the magnitude of the vectors. If a and b are perpendicular, then $a \cdot b = 0$, the geometric product now consisting of only the outer product $a \wedge b$ gives the orientation of the *bivector*. If the vectors a and b are neither of the two extremes then the geometric product will give information something in between. In essence this is the most important element of this algebra and all the other meaningful operations are derived from this geometric product.

4. ROTATIONS IN THE GEOMETRIC ALGEBRA 3-D SPACE

The rotations in GA are represented by a pair of reflections. For any vector a reflected in a plane perpendicular to an unit vector n is $-nan$ and followed by another reflection in another plane perpendicular to m results in another vector given by $-m(-nan)m = (mn)a(nm) = Ra\tilde{R}$ [14]. This product R is a *multivector* and is called a rotational element or Rotor and satisfies $R\tilde{R} = 1$, where $\tilde{R}$ is the conjugate of R. The equation $Ra\tilde{R}$ works for any dimension, any *grade* and any objects.

Therefore taking only the scalar and *bivector* parts, a general rotation in 3-D can be written as shown in eqn (1):

$$R = exp(n\frac{\theta}{2}) = \cos\frac{\theta}{2} + n\sin\frac{\theta}{2} \tag{1}$$

where θ represents a rotation about an axis parallel to unit vector n and the rotation axis n is given by: $n_1e_2e_3 + n_2e_3e_1 + n_3e_1e_2$ is spanned by the *bivector basis*.

5. COLOR AS A VECTOR

It is fair to assume that the human eye doesn't process different colors in RGB images (fig 1) separately but processes like a continuous vector valued approach in the 3-D Euclidean space. In this regard the *bivector* representation of color vectors in GA fits neatly for the 3-D Euclidean space. Then the color information of $(r,g,b)^T$ vector of the color image $c_{m,n}$ can be written as:

$$c_{m,n} = r_{m,n}e_2e_3 + g_{m,n}e_3e_1 + b_{m,n}e_1e_2. \tag{2}$$

where $r_{m,n}$, $g_{m,n}$ and $b_{m,n}$ are the RGB vectors of the image $c_{m,n}$. Alternately, the following analysis and derivations in the next few sections holds good for the color vector $c_{m,n} = r_{m,n}e_1 + g_{m,n}e_2 + b_{m,n}e_3$ definition.

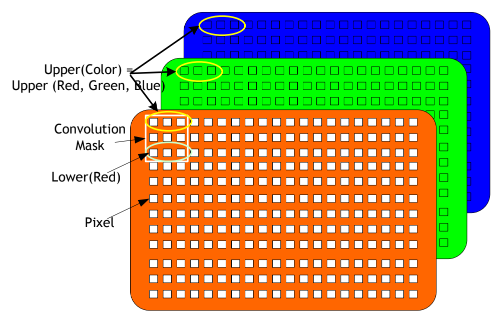

Figure 1: R-G-B plane

6. TRANSFORMATION AND DIFFERENCE SUBSPACE

Though most of the algorithms extend the conventional gray scale algorithms to the color images, very few have treated color as a vector or a single entity. In this paper the color is defined as a single entity (eqn 2) and treat the image as a superset of this entity. Therefore the color edge detection and image processing algorithms with this type of approach becomes straightforward and intuitive.

Let μ be the rotation axis spanned by the *bivector basis* $e_1e_2 + e_2e_3 + e_3e_1$ in the color cube (fig 2) given by:

$$\mu = e_1e_2 + e_2e_3 + e_3e_1$$

For a unit transformation on the color vector the normalized color representation is chosen. This ensures that the orientation information is kept while the distance information is normalized. For a unit transformation on the normalized color, the rotation vector is given by:

$$\mathbf{R} = \cos\theta + \frac{1}{\sqrt{3}}\mu\sin\theta = \cos\theta + \frac{e_{12}+e_{23}+e_{31}}{\sqrt{3}}\sin\theta$$
$$\tilde{\mathbf{R}} = \cos\theta - \frac{1}{\sqrt{3}}\mu\sin\theta = cos\theta - \frac{e_{12}+e_{23}+e_{31}}{\sqrt{3}}\sin\theta$$

These rotations given by $\mathbf{R}$ and $\tilde{\mathbf{R}}$ rotates any vector by an angle θ in 3D about an axis parallel to the rotation axis. Hence the unit transformation or the convolution operation on a color element C is given by:

$$\begin{aligned}\mathbf{R}C\tilde{\mathbf{R}} &= (c\theta + \frac{1}{\sqrt{3}}\mu s\theta)(C)(c\theta - \frac{1}{\sqrt{3}}\mu s\theta) \\ &= \underbrace{c^2\theta(C)}_{\text{I}} - \underbrace{\frac{c\theta s\theta}{\sqrt{3}}(C)\mu}_{\text{II}} + \underbrace{\frac{c\theta s\theta}{\sqrt{3}}\mu(C)}_{\text{III}} + \underbrace{\frac{s^2\theta}{\sqrt{3}}\mu(C)\mu}_{\text{IV}}\end{aligned}$$

where $C = re_{23} + ge_{31} + be_{12}$ and $c\theta$, $s\theta$ is $\cos\theta$ and $\sin\theta$ respectively.

The part of II in the above equation simplifies to

$$-r-g-b-(b-g)e_{23}-(r-b)e_{31}-(g-r)e_{12}$$

and similarly the part of III equates to

$$-r-g-b+(b-g)e_{23}+(r-b)e_{31}+(g-r)e_{12}$$

Finally, the part IV equates to

$$re_{23}+ge_{31}+be_{12}-2(re_{31}+re_{12}+ge_{23}+ge_{12}+be_{23}+be_{31})$$

Putting it altogether, eqn 3 reduces to as follows:

$$\underbrace{\cos 2\theta(re_{23}+ge_{31}+be_{12})}_{\text{A}} + \underbrace{\frac{2}{3}\sin^2\theta\mu(r+g+b)}_{\text{B}}$$
$$\underbrace{+\frac{1}{\sqrt{3}}\sin 2\theta[(b-g)e_{23}+(r-b)e_{31}+(g-r)e_{12}]}_{\text{C}}$$

The first element 'A' of the above equation is the RGB space component, the second element 'B' is the intensity component and 'C' element is the color difference or the projection of the tristimuli in the Maxwell triangle. This is also known as the chromaticity (due to hue and saturation) of the vector.

Interestingly from the above equations if we rotate the vector by an angle $\theta = \frac{\pi}{4}$, the above equation reduces to only two components as shown in eqn 3 and eqn 4, where the space component 'A' is cancelled. This transformation is similar to a RGB space being transformed to an HSI like space and the remaining two components describes the luminance and chrominance of the image. For $\theta = \frac{\pi}{4}$

$$\mathbf{R}C\tilde{\mathbf{R}} = \underbrace{\tfrac{1}{3}\mu(r+g+b)}_{\text{luminance}} + \underbrace{\tfrac{1}{\sqrt{3}}[(b-g)e_{23}+(r-b)e_{31}+(g-r)e_{12}]}_{\text{chrominance}} \quad (3)$$

and similarly the opposite rotation is given by:

$$\tilde{\mathbf{R}}C\mathbf{R} = \underbrace{\tfrac{1}{3}\mu(r+g+b)}_{\text{luminance}} - \underbrace{\tfrac{1}{\sqrt{3}}[(b-g)e_{23}+(r-b)e_{31}+(g-r)e_{12}]}_{\text{chrominance}} \quad (4)$$

If the color vector is homogeneous then addition of two transforms equates to:

$$\mathbf{R}C\tilde{\mathbf{R}} + \tilde{\mathbf{R}}C\mathbf{R} = \frac{2}{3}\mu(r+g+b) \quad (5)$$

which yields the equation for the intensity of the image.

Subtracting the two transforms gives:

$$\mathbf{R}C\tilde{\mathbf{R}} - \tilde{\mathbf{R}}C\mathbf{R} = \frac{2}{\sqrt{3}}[(b-g)e_{23}+(r-b)e_{31}+(g-r)e_{12}] \quad (6)$$

which is the difference of the color vector in the image or the change in chromaticity or shift in hue.

From the above it can be concluded that if the color vectors are different in both the unit transformations then they don't cancel out and that would result in a hue or chromatic shift. On the other hand if the color is same then only the intensity component will be left out. This phenomenon generates edges where sharp changes of color occurs.

7. EXPERIMENTAL RESULTS

7.1 Color Difference Edge Detection by Rotor Convolution

The edge detection process involves convolving masks $m_L(x,y)$ (for left) and $m_R(x,y)$ (for right) of the size $X \times Y$ with the image $c(m,n)$ of dimension $(m \times n)$. In a rotor based approach the convolution involves the geometric product of the vector with the rotor as shown in the convolution eqn 7:

$$\hat{c}(m,n) = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} m_L[x,y]c[m-x,n-y]m_R[x,y] \quad (7)$$

The horizontal left and right masks for the rotor convolution are defined in eqn 8. The vertical masks are obtained by interchanging the rows and columns.

$$m_L = \begin{bmatrix} R & R & R \\ 0 & 0 & 0 \\ \tilde{R} & \tilde{R} & \tilde{R} \end{bmatrix}, m_R = \begin{bmatrix} \tilde{R} & \tilde{R} & \tilde{R} \\ 0 & 0 & 0 \\ R & R & R \end{bmatrix} \quad (8)$$

And the rotors are given by:

$$\mathbf{R} = se^{n\pi/4} = s[cos(\pi/4) + nsin(\pi/4)] \quad (9)$$

where n is the unit vector and is given by: $n = (e_2e_3 + e_3e_1 + e_1e_2)/\sqrt{3}$ and $s = 1/\sqrt{6}$ is the scale factor.

As shown in the fig 2 the color vector is split into two components, ($c_\parallel$ is the component parallel to the gray axis and the perpendicular component is $c_\perp$). When the masks are applied on the color vector only the perpendicular component $c_\perp$ is affected but the parallel component $c_\parallel$ is unchanged [14]. After the convolution the perpendicular component is rotated by an amount specified by the rotor R and by an angle 2θ. As $\mathbf{R}$ is a rotor in a right hand screw sense, the rotor $\mathbf{Ra\tilde{R}}$ would rotate the color vector by the same amount as would the rotor $\mathbf{\tilde{R}aR}$ by the same amount but opposite in direction. Hence, if the color vectors are homogeneous then after the masks are applied, both the components would cancel out (as derived in eqn 5) and the pixel point would fall somewhere on the gray axis or simply perceived as the intensity of the image. However if the color components are

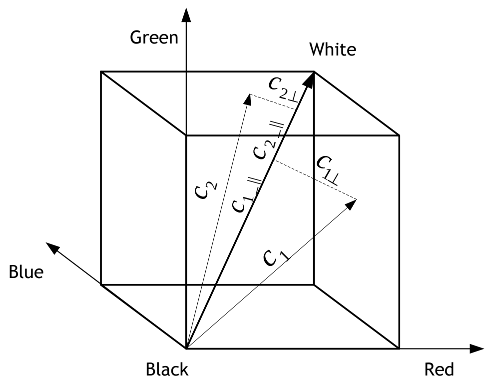

Figure 2: RGB vectors and the Color Cube.

not homogeneous then the color vector will be rotated by an unequal amount by the two rotors. Thus the resultant vector would lie somewhere else in the color cube, and not on the gray axis(eqn 6).

As shown in fig 3 the rotor convolution was applied on the test image of the "color blocks"[1] which has an 8×8 array of colored squares. The result of the filtered image is shown in fig 4. As discussed above, the filtered image has gray areas where the squares had uniform color and colored areas where there was a change of color between the blocks. As only the horizontal masks are applied the change in color is only seen on the horizontal edges. Also the edges between the black and white blocks remain gray but the edges change to color where there was a color difference across the edge. This is because the rotor masks would cancel out the homogeneous components resulting in gray axis and have a color component present if the color components are not homogeneous(as in eqn 5 and eqn 6). This signifies the operation

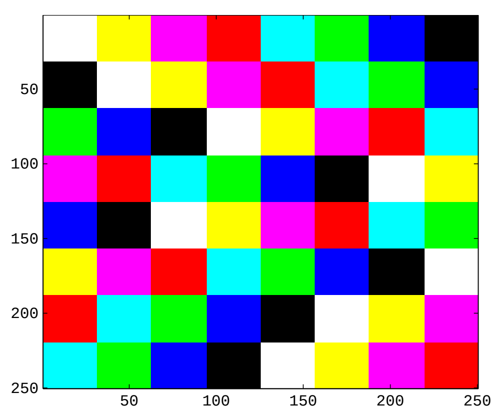

Figure 3: Original Color Block.

is a shift in hue (chromaticity) of the image. This type of change can also be observed on the filtered images of standard lenna (fig 5). The areas where the color change was flat or smoothly varying, the filtered image became grayish, otherwise colors can be observed in regions like the top edges of the hat of the image.

[1] taken from http://privatewww.essex.ac.uk/§js/research/colour_test_images.html

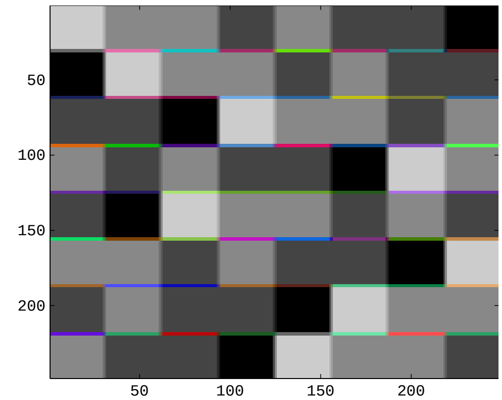

Figure 4: Color Blocks after rotor convolution.

Figure 5: Lenna Image after Rotor Convolution.

7.2 Color Sensitive Edge Detection - Red to Blue

As discussed in the previous section, the rotor edge detection method detects only the chromatic edges. In this section detection of homogeneous regions of particular colors ($C_1 \rightarrow C_2$) is discussed. Here the edges between these two colors are determined by the rotor methods. Only synthetic images are considered to demonstrate the feasibility of the technique on synthetic images. For natural images the criterion is slightly different but is a straightforward extension to the synthetic technique.

For this example, a unit *bivector* basis is considered which preserves the orientation information of the vector. For example, let C_1 be a color *vector* then μC_1 is a normalized color *vector* of C_1 where μ is given by $\mu = \frac{\boldsymbol{e}_2\boldsymbol{e}_3+\boldsymbol{e}_3\boldsymbol{e}_1+\boldsymbol{e}_1\boldsymbol{e}_2}{\sqrt{3}}$.

To find the discontinuity between two regions C_1 to C_2 the convolution uses the following rotor filter.

$$m_L = \frac{1}{\sqrt{6}} \begin{bmatrix} \mu C_1 & \mu C_1 & \mu C_1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} . m_R = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ \mu C_2 & \mu C_2 & \mu C_2 \end{bmatrix} \quad (10)$$

For the convolution operation, the RGB image is converted to a format where the GA operations can be understood. This convolution operation on an image $c(m \times n)$ is given by eqn 7. This results in convolution operation giving non-zero scalar part and zero vector part in special cases. For example, if the masks are multiplied with the following image part (see fig 6) where $C_1 = \boldsymbol{e}_2\boldsymbol{e}_3$(red) and $C_2 = \boldsymbol{e}_3\boldsymbol{e}_1$(blue) then; if

$\mu C_1 = \boldsymbol{e}_2\boldsymbol{e}_3$ and $\mu C_2 = \boldsymbol{e}_3\boldsymbol{e}_1$, then the convolution equation would result in a scalar value $\frac{1}{6}(-3-3) = -1$ or generally is expressed as $\frac{1}{2}(|C_1| + |C_2|)$.

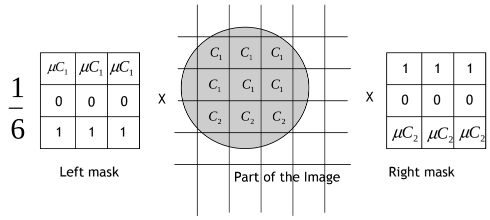

Figure 6: Red to Blue, Convolution by left and right mask

This is where the pixel values will have a non zero scalar value and zero vector value. The scalar value is the intensity of the pixel due to the convolution and is the whiter regions shown in fig 7 at the step edges. In all other places of the image the scalar and vector will have non zero value. In fig 7, only the scalar values are plotted and the vector values are ignored to simplify the discussion.

As shown in fig 4 there is only one region near (6,6) block where the red to blue ($C_1 \rightarrow C_2$) color transition occurs. When convolving with the mask (eqn 10) the vector part will be zero in this region resulting in brightest pixel due to scalar values (see fig 7 marked region). These white pixels show the red to blue transition. The color mask convolution will have a different result when the color transition (e.g. $C_2 \rightarrow C_1$) is opposite. Also the varying degree of the pixel brightness, light and dark gray, in other block edges is due to the varying degree of red and blue components present.

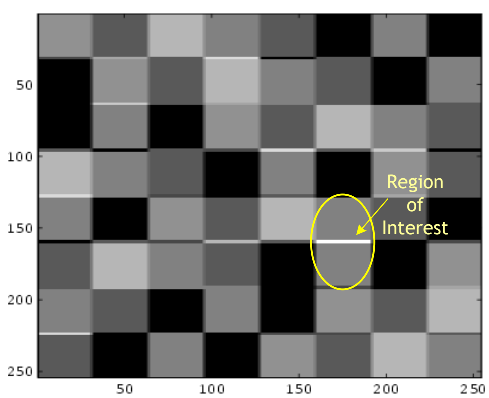

Figure 7: Convolution to detect Red to Blue.

8. CONCLUSION

The GA representation for the color vectors is shown to be a convenient and intuitive way for signal and image processing applications. Experiments showed that this kind of edge detection is compact and is performed wholly on *bivector* color images. It is also concluded that the convolution operation with the rotor masks belong to a class of linear vector filters and can be applied to image or speech signals [2]. The paper also presented an overview of the convolution operations involving rotors for image processing applications. The usefulness of the introduced approach was demonstrated by analyzing and implementing a computation intensive edge detection algorithm. The discussion also touches upon the qualitative analysis of the edge detection algorithm by GA methods.

REFERENCES

[1] A.Koschan and M.Abidi, "Detection and classification of edges in color images," *IEEE Signal Processing Magazine*, vol. 22, no. 1, 2005.

[2] R.Lukac, B.Smolka, K.Martin, K.N.Plataniotis, and A.N.Venetsanopoulos, "Vector Filtering for Colour Imaging," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 74–86, 2005.

[3] H. J. Trussel, E. Saber, and M. Vrhel, "Color image processing," *IEEE Signal Processing Magazine*, vol. 22, no. 1, 2005.

[4] J.Astola, P.Haavisto, and Y.Neuovo, "Vector Median Filters," *Proceedings of the IEEE*, vol. 78, no. 4, 1990.

[5] P.E.Trahanias and A.N.Venetsanopoulos, "Color edge detection using vector order statistics," *IEEE Transactions on Image Processing*, vol. 2, no. 2, 1993.

[6] R. Machuca and K. Phillips, "Applications of Vector Fields to Image Processing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 3, pp. 316–329, 1983.

[7] S.J.Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex numbers," *IEE Electronics Letters*, vol. 32, no. 21, pp. 1979–1980, 1996.

[8] E.Bayro-Corrochano and S.Flores, "Color Edge Detection Using Rotors," in *Applications of Geometric Algebras in Computer Science and Engineering*, C. D. Leo Dorst and J. Lasenby, Eds. Birkhäuser, 2002.

[9] B.Mishra and P.Wilson, "Color Edge Detection Hardware based on Geometric Algebra," in *3rd European Conference on Visual Media Production CVMP 2006*. IET, London, 2006, pp. 115–121.

[10] J. Ebling, "Clifford Fourier Transform on Vector Fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 469–479, 2005.

[11] D. Hestenes, *New Foundations for Classical Mechanics*. D.Reidel Publishing Company, 1986.

[12] L.Dorst and S.Mann, "Geometric Algebra: A Computational Framework for Geometrical Applications(I)," *IEEE Computer Graphics and Applications*, 2002.

[13] R. Goldman, "Computer Graphics in its Fifth Decade: Ferment at the Foundations," in *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*. Washington, DC, USA: IEEE Computer Society, 2003, p. 4.

[14] J. Lasenby, W. J. Fitzgerald, A. N. Lasenby, and C. J. L. Doran, "New Geometric Methods for Computer Vision: An Application to Structure and Motion Estimation," *Int. J. Comput. Vision*, vol. 26, no. 3, pp. 191–213, 1998.