HYPERCOMPLEX COLOR AFFINE FILTERS

Todd A. Ell

University of Essex
Department of Electronic Systems Engineering
Colchester C04 3SQ, U.K.
Email: t.ell@ieee.org

ABSTRACT

In linear gray-scale image convolution filters, pixel values can only be scaled. But, in color image convolution filters, which treat color image pixels as vectors, pixel values can be submitted to a host of affine transforms: scaling, reflection, rotation and shearing. Hypercomplex affine transforms are presented which extend linear color image convolution filtering techniques.

Index Terms— quaternion, affine transform, color image, convolution filter

1. INTRODUCTION

Fundamental to image processing is the convolution operator. At its basic arithmetic level convolutions consist of shifted (in image index-space) and scaled (in pixel value-space) accumulations based on the relative location of scalar values in the convolution filter mask image. Typical color-image processing consists of separating the color channels of the image and applying scalar convolution filters to each channel then merging the channels into the final image. This restricts the convolution operator to the scaling of pixel-values. By treating the pixel values as hypercomplex vectors the convolution operator also has available stretching, reflecting, rotating and multiple shear operations.

Examination of the 1D discrete convolution operation will provide insight for later detailed discussions. The 1D discrete convolution is defined as:

$$Y_n = \sum_{k=1}^{K} m_k x_{n-k}, \quad n = 1, 2 \ldots N$$

Each summand, $m_k x_{n-k}$, is a linear map that scales the image pixel value, x_{n-k}, by the mask value m_k. Let $y_{n,k} = m_k x_{n-k}$, then the convolution is simply the sum of multiple linear maps:

$$Y_n = \sum_{k=1}^{K} y_{k,n}$$

Over the real numbers, the classical form for a linear map is $y = mx$, where $y, m, x \in \mathbb{R}$. In contrast the general linear map over the quaternions, $\mathbb{H}$, takes the form

$$y = L_1 x R_1 + L_2 x R_2 + L_3 x R_3 + \ldots = \sum_{n=1}^{\infty} L_n x R_n$$

where $y, x, L_n, R_n \in \mathbb{H}$. Each summand, as expressed in the equation above, cannot be reduced to a single monomial as is the case for real-valued maps because quaternion multiplication is not commutative. This creates difficulties in manipulating the convolution operators, but for the same reason expresses powerful utility. The real linear map can only encode pixel scaling (i.e., pixel x is scaled by m). The quaternion linear map, by contrast, can encode dilations, reflections, rotations and shears. Understanding the unique linear forms for each of these quaternion maps is recommended if linear hypercomplex convolution filters are to be useful as a tool for color-image processing.

2. QUATERNIONS

In this work we use the hypercomplex numbers of Hamilton [1], namely the quaternion 4-tuple (w, r, g, b) denoted in hypercomplex form as: $q = w + ir + jg + kb$ where $w, r, g, b \in \mathbb{R}$ and the hypercomplex operators follow the rule

$$ijk = i^2 = j^2 = k^2 = -1$$

All quaternions can be split into scalar and vector parts, i.e.,

$$q = S[q] + V[q]$$

where $S[q] = w$ and $V[q] = ir + jg + kb$. Conjugation is denoted with an over-bar which negates the vector part. Color pixel values are encoded into the vector part of a quaternion making a pure quaternion. We denote pure unit length quaternions as μ, scalars as α and β , and finally pixel values as p.

3. LINEAR QUATERNION AFFINE MAPS

It is well known that vector rotation and dilation (stretching and compression) in 3D can be encoded as a linear quaternion monomial map. Less known is their ability to encode reflections [2], which results from the fact that any rotation can be decomposed into a pair of nested reflections. Even more obscure is the use of multi-nomial linear quaternion maps to encode shears. Some works in the literature stop at monomial maps and hence assume only rotations, reflections and simple dilations can be encoded with any linear quaternion map. See, for example, Kuipers [3, p. 345]. Reflections are formed as monomial maps, but complex dilations (beyond scaling toward or away from the origin) and shears require binomial and tetranomial maps, respectively. It should be noted that all the operators presented herein map vectors to vectors, which is necessary if the convolution operator is to map images into images. Also note that the list of maps presented here is by no means exhaustive; this short list is simply used to illustrate their effectiveness.

The author is a Visiting Fellow at the University of Essex. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council under Grant GR/S58621/01.

3.1. Rotations

Rotations of three-dimensional vectors p are encoded with a unit quaternion q in the well known linear quaternion equation [4]

$$R_q[p] = qp\bar{q}$$

The axis of the rotation corresponds to the eigen-axis of the quaternion q and the angle of rotation is twice the eigen-angle of q. The composition of two rotations $R_q[\,]$ and $R_r[\,]$ is given by

$$R_r[R_q[p]] = r(qp\bar{q})\bar{r} = (qr)p\overline{(qr)}$$

Finally, the inverse rotation, $R_q^{-1}[\,]$, of $R_q[\,]$ is given by

$$R_q^{-1}[p] = \bar{q}pq$$

The rotation monomial map has been used to construct a number of convolution filters that perform color-edge detection [5].

3.2. Reflections

Planar reflections, denoted $R_\mu[\,]$, of three-dimensional vectors p across a plane defined by its normal unit vector μ (i.e., a pure unit quaternion) are encoded in the linear quaternion equation

$$R_\mu[p] = \mu p \mu$$

Composite reflections create rotations. For example, the composition of two reflections $R_{\mu_1}[\,]$ and $R_{\mu_2}[\,]$ is

$$\begin{aligned} R_{\mu_2}[R_{\mu_1}[p]] &= \mu_2(\mu_1 p \mu_1)\mu_2 \\ &= -(\mu_2\mu_1)p\overline{(\mu_2\mu_1)} = -qp\bar{q}. \end{aligned}$$

where $q = \mu_2\mu_1$ provided that $\mu_2 \neq \mu_1$. Reflections are their own inverse operations, so applying the same reflection twice (i.e., $\mu_2 = \mu_1$) returns the original vector.

3.3. Shears

This section introduces, for the first time, the quaternion shear operator. There are multiple ways to shear a 3D object. We address two particular shears: the axial- and beam-shear. Shears cannot be built from monomial maps, they require at least a binomial form. Much like rotations being decomposed into a pair of reflections, rotations can also be decomposed into three 2D beam-shears. This property, in matrix form, is used to perform fast 3D volume rotations [6].

The simplest shears are axial neutral shears. They are axial neutral since the axis line has no shear effect. The axial shear equation is

$$S_{\alpha,\mu_3}[p] = p\left(1 - \tfrac{\alpha}{2}\mu_3\right) - \tfrac{\alpha}{2}\mu_2 p \mu_1,$$

where μ_3 is the axis of the shear and α is the shear factor. It is required that $\mu_1 \perp \mu_2 \perp \mu_3$ and $\mu_1\mu_2 = \mu_3$ (i.e., they form an orthogonal right-handed triad). This requirement ensures that vectors are mapped to vectors. Note that this equation reduces to the identity when $\alpha = 0$. Reversing the shear factor α reverses the axis of the shear (reversing μ_3 must be matched by a reversal of μ_1 to maintain the right-handed triad). Reversing the shear factor also forms the inverse shear.

The next level of shears is the 2D beam-shears as defined by Chen and Kaufman [7]. The quaternion binomial form of this shear is

$$\begin{aligned} S_{\alpha,\beta,\mu_3}[p] = p\left(\tfrac{1}{2} + \tfrac{\alpha}{2}\mu_3\right) - \tfrac{\alpha}{2}\mu_2 p \mu_1 \\ + \left(\tfrac{1}{2} + \tfrac{\beta}{2}\mu_2\right)p - \tfrac{\beta}{2}\mu_1 p \mu_3. \end{aligned}$$

where again μ_1, μ_2 and μ_3 form a right-handed triad. Notice that this is a tetranomial equation; it cannot be reduced to fewer terms.

3.4. Dilations

In this section we will introduce, for the first time, two dilation operations: axial- and radial-dilation. The first is rectilinear, and as the name implies, the second is radial. An axial-dilation is given by the binomial equation as

$$D_{\mu,\alpha}[p] = \tfrac{1+\alpha}{2}p + \tfrac{1-\alpha}{2}\mu p \mu$$

where the axis of dilation or compression is defined by μ. Compression occurs when dilation factor, α, satisfies $|\alpha| < 1$, dilation occurs when $|\alpha| > 1$ and identity occurs when $\alpha = 1$. Negative values for α yield vector reversal, hence simple reversal (with no compression or dilation) occurs when $\alpha = -1$. The same equation with a reciprocal factor defines the inverse map.

Radial dilation is given by the binomial equation

$$D_{\mu,\alpha}[p] = \tfrac{2+\alpha}{2}p + \tfrac{\alpha}{2}\mu p \mu$$

which expands space outwards from an invariant line defined by a unit vector μ. This equation reduces to an identity when $\alpha = 0$. Dilation occurs when $\alpha > 0$. Compression occurs when $\alpha < 0$. As a result the image is complemented (in an opposing color sense) when $\alpha = -2$. The same equation with a reciprocal expansion factor defines the inversion.

4. SOME EXAMPLES

This section makes concrete the discussion of the previous section by applying each of the maps to two different images. The first image is the well known 'Lena' image. The second is a test image consisting of equally spaced pixel values along the edge of the RGB color cube but scaled to 90% of full scale. Each image is represented in a gray-centered RGB color space where the origin is centered at mid-gray. This color-space allows for an efficient, color preserving, clipping algorithm that is necessary when the transforms map a color outside the color-cube. For details see Sangwine and Ell [8].

The first two columns of Figure 1 show the result of applying each linear map, starting with the original images in the first row. The third column in the figure consists of a 3D color scatter plots showing the locations, in a color-cube, of each pixel in the Lena and test images. Each scatter-point is colored to represent the corresponding pixel color from the image. As can be seen the test image is used to emphasize the various distortions caused by the linear maps. Vector coordinates are ordered: red, green, blue. The parameters for each map are:

Rotation:	$\mu = i,\ angle = -15^\circ$,
Reflection:	$\mu_1 = (i-j)/\sqrt{2}$,
Axial-Shear:	$\mu_1 = i, \mu_2 = k, \alpha = 0.1$,
Axial-Dilation:	$\mu_1 = k, \mu_2 = i, \alpha = 0.4$,
Radial-Dilation:	$\mu_1 = (i+j+k)/\sqrt{3}, \alpha = -0.6$, and
Beam-Shear:	$\mu_1 = i, \mu_2 = j, \alpha = 0.2, \beta = 0.1$.

Calling attention to the radial-dilated image, one finds that if the axis of dilation is the grey-line (as in this example) and if the compression factor is 100%, then the resulting image would consist only of the luminance data from the original image. Likewise, if the axial-dilation along the same axis is compressed 100% (making it a plane-projection operator) then the resulting image would consist only of the chrominance data from the original image. These two maps alone can be used to build pre-filters that independently separate the luminance and chrominance channels of an image without the need to pre-transform the images to a luminance-chrominance

color-space representation. The results of these pre-filters could then be cascaded with another set of post-filters. The final results could be output directly in RGB color space by simply connecting those filters in parallel, i.e., summing their outputs. This would all be done using linear quaternion algebra, without the need for procedural algorithms. However, this does not mean the post-filters can only affect the data channel they are processing. Even though, for example, the luminance post-filter receives only luminance data it is still a full color filter and its outputs can contain chrominance data.

It should be pointed out that all of these maps can be described in terms of a point-mask two-sided convolution operation. These maps represent the color-space extension to point-mask gray-scale image scaling operations. As such, they should provide a rich source of color-space convolution filters when the convolution masks are given broader support, i.e., they are no longer point-masks but have pixel extent such as standard 3×3 pixel masks. Initial steps in the design of these more complicated masks are given in the previous works of the author with other collaborators [9], [10], [11], [12].

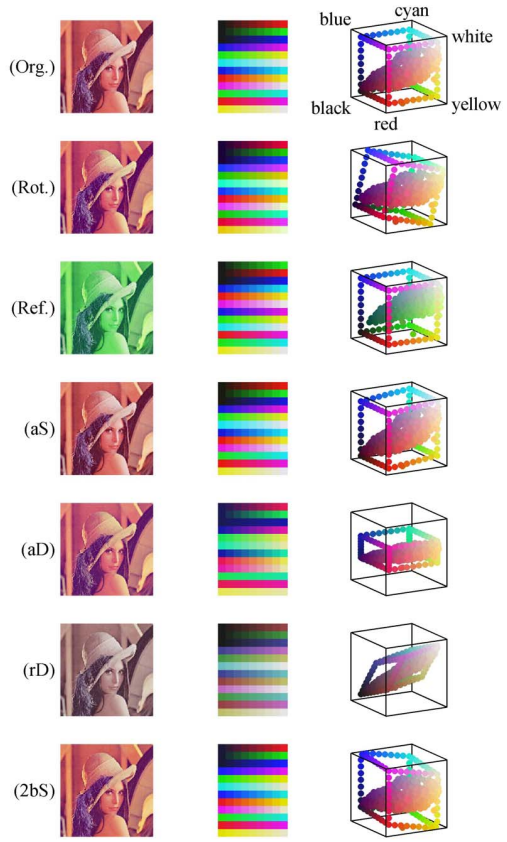

Fig. 1. Quaternion linear transforms of 'Lena' and color-cube test images. The 1st and 2nd columns contain the transformed images, and the 3rd column shows a color-cube scatter plot of both images' contents. Transforms of the original image (Org.) are by rows: rotation (Rot.), reflection (Ref.), axial-shear (aS), axial-dilation (aD), radial-dilation (rD), and 2D-beam-shear (2bS).

5. AN APPLICATION

Since the quaternion convolution is linear, cascade and parallel connections of the simpler affine maps can be used to construct more complex filtering operations. For example, the simplest edge-detection filter is achieved using opposing scale factors across a two-pixel mask as

$$y_k = m_1 p_k + m_2 p_{k-1},$$

where $m_2 = -m_1$. The scale-factor pair, $\{m_1, m_2\}$, may be replaced with any pair of inverse affine maps

$$\left\{L, L^{-1}\right\} \in \left\{ \begin{array}{c} \left\{R_q, R_q^{-1}\right\}, \left\{R_\mu, R_\mu^{-1}\right\}, \\ \left\{S_\alpha, S_\alpha^{-1}\right\}, \left\{D_{\mu,\alpha}, D_{\mu,\alpha}^{-1}\right\} \end{array} \right\}$$

giving a generalized 'edge-detector'

$$y_k = L\left(p_k\right) + L^{-1}\left(p_{k-1}\right).$$

In this fashion specific patterns of transition in an image may be highlighted using a single convolution filter. This would find usefulness, for example, in wind tunnel imagery where identifying specific turbulence patterns is important in the design of efficient aero-dynamic shapes and brute-force procedural search algorithms of high resolution images is too time consuming or obscured by other affects.

6. CONCLUSIONS

The hypercomplex convolution operator has significance for designing color-image filters. The linear transforms presented embrace many color-space operations such as reflections, rotations, dilations and shears that map color images into color images. All of these color-space operations can be performed with *linear* quaternion valued equations. This is important so that convolution is index-space shift invariant and color-space linear which preserve the mapping of sinusoidal inputs to sinusoidal outputs. The class of hypercomplex convolution filters is of great utility for this reason.

7. REFERENCES

[1] W. R. Hamilton, *Elements of Quaternions*, Chelsea Publishing Company, 1996.

[2] H. S. M. Coxeter, "Quaternions and reflections," *American Mathematical Monthly*, vol. 53, pp. 136–146, Mar. 1946.

[3] Jack B. Kuipers, *Quaternions and Rotation Sequences*, Princeton University Press, 1999.

[4] S. L. Altmann, *Rotations, Quaternions, and Double Groups*, Clarendon Press, Oxford, 1986.

[5] S. J. Sangwine and T. A. Ell, "Colour image filters based on hypercomplex convolution," *IEE Proceedings – Vision, Image and Signal Processing*, vol. 147, no. 2, pp. 89–93, Apr. 2000.

[6] T. Toffoli and J. Quick, "Three-dimensional rotations by three shears," *Graphical Models and Image Processing*, vol. 59, pp. 89–96, March 1997.

[7] B. Chen and A. Kaufman, "3d volume rotation using shear transformation," *Graphical Models*, vol. 62, pp. 308–322, 2000.

[8] S. J. Sangwine and T. A. Ell, "Gray-centered RGB color space," in *Second European Conference on Color in Graphics, Imaging and Vision (CGIV 2004)*, Technology Center AGIT, Aachen, Germany, 5–8 Apr. 2004, pp. 183–186, The Society for Imaging Science and Technology.

[9] S. J. Sangwine, B. N. Gatsheni, and T. A. Ell, "Linear colour-dependent image filtering based on vector decomposition," in *Proceedings of EUSIPCO 2002, XI European Signal Processing Conference*, Toulouse, France, 3–6 Sept. 2002, vol. II, pp. 274–277, European Association for Signal Processing.

[10] S. J. Sangwine, B. N. Gatsheni, and T. A. Ell, "Vector amplification for color-dependent image filtering," in *IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain, 14–17 Sept. 2003, vol. 2, pp. 129–132, Institute of Electrical and Electronics Engineers.

[11] C. J. Evans, S. J. Sangwine, and T. A. Ell, "Colour-sensitive edge detection using hypercomplex filters," in *Proceedings of EUSIPCO 2000, Tenth European Signal Processing Conference*, Moncef Gabbouj and Pauli Kuosmanen, Eds., Tampere, Finland, 5–8 Sept. 2000, vol. I, pp. 107–110, European Association for Signal Processing.

[12] C. J. Evans, T. A. Ell, and S. J. Sangwine, "Hypercomplex color-sensitive smoothing filters," in *IEEE International Conference on Image Processing (ICIP 2000)*, Vancouver, Canada, 11–14 Sept. 2000, vol. I, pp. 541–544, Institute of Electrical and Electronics Engineers.