Quaternion Colour Texture

L. Shi, B. Funt
School of Computing Science, Simon Fraser University, Vancouver, Canada
Corresponding Author: funt@sfu.ca

ABSTRACT

The quaternion representation of colour is shown to be effective in the context of colour texture region segmentation in digital colour images. The advantage of using quaternion arithmetic is that a colour can be represented and analyzed as a single entity. A basis for the colour textures occurring in a given image is derived via quaternion principal component analysis of a training set of colour texture samples. A colour texture sample is then projected onto a subset of this basis to obtain a concise (single quaternion) description of the texture. The power of this quaternion colour texture representation is then demonstrated by its use in an algorithm that successfully segments an image into regions of different texture.

1. INTRODUCTION

Many different local image features have been proposed for characterizing textures[1,2,3,9]. Since there are many possible texture features, finding a feature vector with good discriminating power is important. For colour imagery, Hoang et. al.[9] showed that using colour and texture in combination results in better discrimination than using the colour and texture features separately.

We propose a new colour texture classification method based on the quaternion representation of a colour image. Quaternions provide a new way to process colour and texture in combination. The advantage of the quaternion representation for colour, as proposed by Sangwine[4,5,8], is that it combines a colour 3-tuple (RGB or LMS) into a single hypercomplex number. A colour can then be processed as a unit, rather than as 3 separate channels. A quaternion has a real part and three imaginary parts and can be written as $q = a + b \cdot i + c \cdot j + d \cdot k$. An RGB colour triple is represented as a purely imaginary quaternion of the form $R \cdot i + G \cdot j + B \cdot k$. Both the Fourier transform[5,6] and principal component analysis[7,8] have been extended to quaternion arithmetic.

Standard principal component analysis (PCA) has been used previously for texture classification in grayscale images[9,11]. Here, we use quaternion principal component analysis (QPCA) with colours encoded as quaternions to calculate a quaternion basis for colour textures. Whether in grayscale or colour, the basic idea is the same; namely, to use PCA/QPCA to obtain a low-dimensional representation of textures sampled from image subregions. PCA/QPCA computes a basis ordered in terms of the amount of variance accounted for in the data. The low-dimensional approximation is constructed by projecting the original input onto the first few basis vectors. Two textures are compared by projecting each onto the first few basis vectors and then comparing the resulting coefficients.

To test whether or not the QPCA representation of colour texture is effective, we use it to segment images into regions of homogeneous texture and make a qualitative comparison to the previously published results[9]. The segmentation is initialized using k-means clustering which is then followed by an iterative stage of region agglomeration.

2. METHOD

In this section, we detail the algorithm for segmenting an image into regions of homogenous colour texture. There are three stages: colour texture feature extraction, feature clustering, and region merging.

The first stage is to extract an orthogonal basis for the colour textures in an image. This basis is calculated by sampling square windows from the image and expressing the contents of each such

window as a vector of quaternions and arranging these vectors as the columns of a matrix. QPCA applied to this matrix yields an orthogonal basis for the contents of the windows ordered in terms of the variance accounted for by each basis vector. Similar to the standard PCA, the dimensionality of the feature space can then be reduced by selecting only the first few bases that account for the majority of the variance. These basis vectors are vectors of quaternions. The contents of any image window can then be approximated concisely by projecting them onto the selected basis. The projection can then be used as the extracted colour texture feature of the window.

For image texture segmentation, we are concerned with representing the textures that occur in the given image, not textures in general. Therefore, the first step is to create a training set of texture samples from the given image. The training set is drawn from image windows of size $w \times w$, where w depends on the image resolution. It is chosen to be large enough to cover a representative texture element and small enough so that it will not generally cover multiple texture elements. In order to reduce the computational cost, we train the texture classifier on a reduced set of possible samples from the input image. This reduced set is used to compute the quaternion texture basis. We tried a sampling of windows centered on random pixels, a fixed set of overlapping windows and a smaller fixed set of non-overlapping windows. Without sampling, a 256 by 256 colour image would generate 65536 feature vectors, and with a window size 15 by 15 the length of each feature vector is 225. Since each element feature vector element is a 4-component quaternion number, the total number of reals QPCA would need to handle is 255x65536x4. We used non-overlapping windows in the experiments reported below.

The RGB pixel values from each window are represented as quaternions and formed into a column vector of quaternions v_q of size w^2. The training set of textures is then represented as a matrix T_q with columns v_q. QPCA decomposes T_q into singular values and their corresponding quaternion eigenvectors U_q. Taking only the first d of the w^2 eigenvectors reduces the dimensionality of the texture model. This of course reduces the accuracy with which the training set can be represented, but the expectation is that only features that are irrelevant to texture discrimination will be lost. Surprisingly, we found experimentally that the best choice was d=1, meaning that the method relies on the first basis vector of U_q only. Therefore, a window's quaternion vector v_q of size w^2 is reduced to size 1, so its texture is describe by a single quaternion. The resulting quaternion need no longer be a purely imaginary quaternion and generally does include a non-zero real part, so effectively texture is being represented by a 4-tuple.

The second stage of the quaternion colour texture analysis is texture clustering based on the features from each sub-window. With d=1, the whole training set T_q is represented by $T_q^1 = U_q^1 \cdot T_q$, where U_q^d represents the first d eigenvectors. Texture classification begins by k-means clustering of the textures in the training set according to there associated reduced feature vectors T_q^1. In other words, the quaternion feature vectors (the columns of T_q^1) are input to the k-means algorithm. The output of the k-means clustering is k centroids describing the mean features of the texture clusters. The parameter k is chosen to be larger than the number of regions expected in the final region segmentation. For example, k=15 is a good choice.

The next step is to classify each image pixel in terms of the texture of w×w window centered on it. The window's contents are represented as a vector of quaternions and then projected onto the basis U_q^1 producing feature vector to v_q^1. The pixel feature vectors are then smoothed spatially. The pixel's texture is assigned a label based on the training set cluster to which it is closest in terms of its Euclidean distance to the cluster's centroid.

The final stage in the texture segmentation process is region merging. The initial clustering was based on k-means clustering with k set to be larger than the anticipated number of final regions. As a result, after the initial clustering some regions still need to be merged. The merging proceeds iteratively and is based on combining statistically similar regions. At each iteration, the two most similar clusters are merged according to a region-similarity measure. The similarity of two regions R_i, R_j with mean feature vectors μ_i and μ_j and covariance matrices Φ_i, Φ_j is defined in[10] by:

$$S_{i,j} = (\mu_i - \mu_j)^T [\Phi_i + \Phi_j]^{-1} (\mu_i - \mu_j) \quad (1)$$

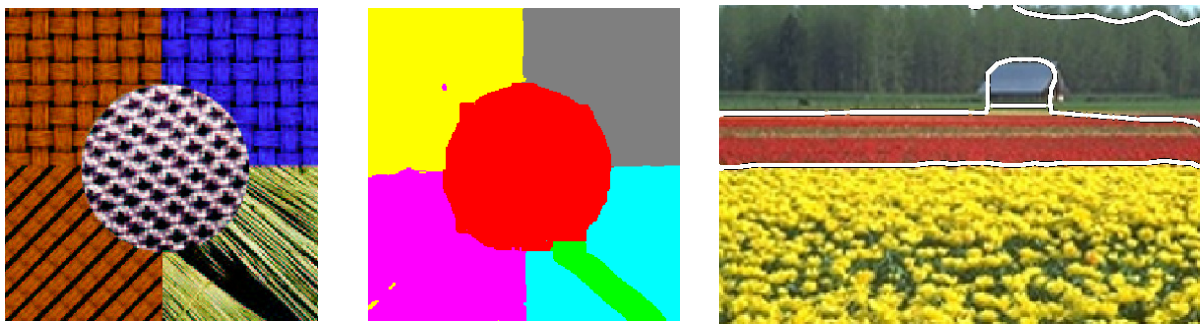
The smaller $S_{i,j}$, the more similar the clusters. At each iteration, the two regions with the smallest $S_{i,j}$ are merged until no $S_{i,j}$ is less than a specified threshold. Finally, due to the fact that the sample window may cover more than one region, a post-processing step is needed to eliminate spurious segmentations at region boundaries. If in a small neighbourhood around a pixel includes 3 or more different texture labels then the pixel in the middle is assigned to the nearest of the other two regions.

3. RESULTS

The results of segmentation using the QPCA based feature extraction method are illustrated in Figure 1. The input image[9] is composed of five colour textures with different colour or patterns. The left hand quadrants have similar colour and texture, although the texture is at a different orientation, while the upper quadrants have exactly the same pattern but different colour. The central circular region is a different colour and texture. Our new method successfully segments the five regions as shown in Figure 1b. The results can be compared to those in[9]. Not shown in Figure 1 is what happens as we lower the similarity threshold so more regions merge: First, the left hand quadrants, which are similar in colour and texture, merge; Second, the shadow area in the lower-right quadrant disappears. With the quaternion representation, it is also possible to change the relative importance assigned to colour versus grayscale texture. When the segmentation is based on intensity alone, the upper quadrants always merge. On the other hand, segmentation based on colour alone always merges the left quadrants.

4. CONCLUSIONS

Quaternions have been used here to represent color in the context of the analysis of color texture. Colour texture features are determined by quaternion principal components analysis of the data from many sub-windows of an image. Experiments show that these features can be used to cluster textures and then segment images into regions of homogenous texture. This demonstrates once again that the quaternion representation of color, which treats color channels as single unit instead of as separate components, truly can be effective.



(a) (b) (c)

Figure 1: Texture segmentation results (colour images reproduced here in grayscale): **(a)** synthetic input image with brown regions on the top left and bottom left, a blue region in the top right, a pale green region in the bottom right and a grayish circular region in the center [9]. **(b)** Segmentation result for (a) showing that the QPCA method successfully separates the top left and top right regions which differ in colour but have similar (although rotated) grayscale structure, and simultaneously separates the top-left and bottom-left regions which differ in grayscale structure but have similar colour. The shadow in the lower right region is identified. The result in (b) is similar to that in [9] (page 272, figure 3(d)) without shadow invariance. **(c)** Result on a natural image (from the Corel database) in which regions of yellow flowers (lower section), red flowers (middle), a gray barn roof, green grass/trees, and blue sky are each segmented. (Parameters used were: image size 192x128, window size 17x17, abutting windows, d=1, initial number of k-means clusters 15, similarity threshold 5.0, Gaussian smoothing σ=4)

References

1. S.E. Grigorescu, N. Petkov, P. Kruizinga, “Comparison of texture features based on Gabor filters,” *IP*(11), No. 10, October 2002, pp. 1160-1167.
2. S. Arivazhagan, L. Ganesan, “Texture segmentation using wavelet transform,” *PRL*(24), No. 16, December 2003, pp. 3197-3203.
3. Z. Lu, W. Xie, J. Pei, and J. Huang. “Dynamic texture recognition by spatiotemporal multiresolution histogram,” In *Proc. IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, 2005.
4. C.J. Evans, T.A. Ell, and S.J. Sangwine , “Hypercomplex Colour-Sensitive Smoothing Filters,” *IEEE International Conference on Image Processing (ICIP)*, 2000, I, pp. 541-544.
5. S.J. Sangwine, and T. A. Ell, “Hypercomplex Fourier Transforms of Colour Images,” *IEEE International Conference on Image Processing (ICIP)*, 2001, I, pp. 137-140.
6. S.C. Pei, J.J. Ding, J.H. Chang, “Efficient implementation of quaternion Fourier transform, convolution, and correlation by 2-D complex FFT,” *IEEE Trans. Signal Process*. 49(2001), No. 11, 2783-2797.
7. S.C Pei, J.H. Chang, J.J. Ding, “Quaternion matrix singular value decomposition and its applications for color image processing,” *IEEE International Conference on Image Processing (ICIP)*, 2003, I, 805-808.
8. N. Le Bihan and S.J. Sangwine, "Quaternion Principal Component Analysis of Colour images," *IEEE International Conference on Image Processing (ICIP)*, 2003, I, 809-812.
9. M. A. Hoang, J. M. Geusebroek, and A. W. M. Smeulders. “Color texture measurement and segmentation,” *Signal Processing*, 2005, 85(2):265-275.
10. H.T. Nguyen, M. Worring, and A. Dev. “Detection of moving objects in video using a robust motion similarity measure,” *IEEE Trans. on Image Proc.*, 2000, 9(1):137-141.
11. D. Alleysson and S. Süsstrunk, “Spatio-chromatic PCA of a Mosaiced color image,” *Proc. IS&T Second European Conference on Color in Graphics, Image, and Vision (CGIV)*, April 2004, pp. 311-314.