A New Approach to Texture Recognition Using Decorrelation Stretching

Micha Kalfon and Moshe Porat

***Abstract*—Texture features have always been a key attribute in image recognition and classification. In this paper we propose two pre-processing methods for enhancing the performance of widely used color texture recognition methods. In the first approach we propose decorrelation stretching for color enhancement, which is known to improve the interpretability of color images. The second method employs Cartoon-Texture decomposition for sharpening the texture component of the image. We show that both methods improve the classification accuracy by 7% and 4% respectively when applied to images prior to extracting auto and cross-correlation features. Our conclusion is that the proposed approach could be helpful in machine vision tasks.**

***Index Terms*—Texture classification, decorrelation stretching, cartoon-texture decomposition, cross-correlation, sharpening.**

I. INTRODUCTION

The importance of texture analysis in image recognition and classification was demonstrated thoroughly in the last decades. The use of color information in texture discrimination has also been studied and several techniques were suggested for extracting texture features from the image color bands [1]-[4]. While some of those techniques naïvely attempt to apply traditional gray-scale texture analysis methods to each color band (i.e., RGB) separately, others have tried to exploit inter-band correlation as a discriminative feature of the color texture. In [1] the values of the auto and cross-correlation function in a small region around the origin are used to represent the color texture,assuming a wide-sense stationary model. Another approach that utilizes the correlation between wavelet coefficients in different color bands was studied in [2]. The work in [3] showed how to extend the well-known Haralick features extracted from co-occurrence matrices (CCM) [4] for capturing the correlation between textures of different color channels.

Decorrelation stretching [5] was introduced as a method for increasing the interpretability of multispectral images. It employs Principal Components Analysis (PCA) in order to stretch the range of points in the RGB color space.

The Cartoon-Texture decomposition problem deals with decomposing an image into piece-wise smooth (cartoon) and oscillatory (texture + noise) components. Most of the algorithms dealing with the problem ([6], [7], [8] to name a few) try to minimize an energy term that penalize oscillatory features in the supposedly piecewise smooth image and vice-versa. The algorithm in [8] decomposes the image by applying low-pass (cartoon) and high-pass (texture) filters. While this filtering is linear it fails to differentiate texture from edges as both are characterized by high spatial frequencies. The algorithm suggested by [9] is based on the model in [8] and suggests a simple yet efficient non-linear filtering scheme for deciding whether a given pixel belongs to cartoon or texture region.

Based on these considerations, we propose in this paper to harness the power of such pre-processing methods for improving the performance of existing texture image classification techniques.

The rest of the paper is organized as follows: In Section II we describe the feature extraction methods employed in the experiments. Section III describes the decorrelation stretching and texture sharpening methods we propose to use for pre-processing. In Section IV we demonstrate the improvement achieved in classification by conducting a retrieval experiment on a subset of the VisTex database. We conclude our results and discuss them in Section V.

II. FEATURE EXTRACTION

The spatial correlation function within and between color bands and its contribution to color texture recognition has been investigated in [1] and [10]. Here, the color texture model follows the assumption that each color band in the image is wide-sense stationary and each pair of bands is jointly wide-sense stationary. Under this assumption, one can define a cross-correlation function $R_{ij}(n,m)$ as follows:

$$R_{ij}(n,m) = E[I_i(p,q) - \mu_i][I_j(p+n, q+m) - \mu_j]. \quad (1)$$

$I_i(n,m)$ is the intensity value of color band i ($i \in \{R,G,B\}$) in pixel (n,m) and μ_i is the average intensity of color band i. Throughout the rest of this section we will assume that the color bands have been normalized to have zero mean, that is $\mu_i = 0$.

While the cross-correlation function $R_{ij}(n,m)$ provides good features for texture recognition, it cannot be used efficiently 'as is', since it has as many values as the dimensions of the texture image itself. This obviously does not allow for efficient retrieval and storage. To overcome this, we further assume that the texture image is a first order wide-sense Markov process. Under this assumption, the intensity of each pixel in color band I can be linearly predicted by its 3 causal neighbors from each color band:

Manuscript received June 15, 2012, revised July 12, 2012.

This research was supported in part by Technion's fund #7110134 and by the Ollendorff Minerva Center. Minerva is funded through the BMBF.

Micha Kalfon is with the Computer Science Department, Technion, Haifa 32000, Israel (e-mail: smichaku@cs.technion.ac.il).

Moshe Porat is with the Electrical Engineering Department, Technion, Haifa 32000, Israel (e-mail: mp@ee.technion.ac.il).

$$\hat{I}_i(m,n) = \sum_{j \in B} \sum_{(p,q) \in D} C_{ij}(p,q) I_j(m-p, n-q), \qquad (2)$$

where $D = \{(0, 1), (1, 0), (1, 1)\}$ and B is the set of color bands determined by the color space in which the image is represented. The coefficients $C_{ij}(p, q)$ are chosen such that the mean square prediction error,

$$\epsilon = E\left[\left(I_i(m,n) - \hat{I}_i(m,n)\right)^2\right], \qquad (3)$$

Is minimized. Setting the derivative with respect to the coefficients $C_{ij}(p, q)$ to zero yields the normal equations:

$$\frac{\partial \epsilon}{\partial C_{ij}(p,q)} = 0. \qquad (4)$$

Substituting $(m, n) = (p, q)$ in the above set of equations gives:

$$E\left[I_i(p,q) I_j(0,0) - \hat{I}_i(p,q) I_j(0,0)\right] = \qquad (5)$$
$$R_{ij}(p,q) - \sum_{k \in B} \sum_{(r,s) \in D} C_{ik}(r,s) R_{kj}(p-r, q-s) = 0\,.$$

For $i, j \in$ B and $p, q \in D$. The coefficients of the linear predictor are therefore determined by the three values (from each color band) of the cross-correlation function around the origin. The first order wide-sense Markov model described above behaves quite well for small local regions and proved itself in the context of texture interpolation and reconstruction [11].

In this work we have used the three values of the auto and cross-correlation function within and between color bands.

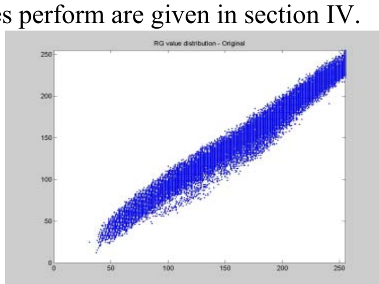
First, each color band is normalized to have zero mean. Then, the three values for each auto and cross-correlation function is computed as discussed above. Finally, each correlation value is normalized with respect to the geometric average of color bands variances:

$$\bar{R}_{ij}(p,q) = \frac{R_{ij}(p,q)}{\sqrt{R_{ii}(0,0) R_{jj}(0,0)}}. \qquad (6)$$

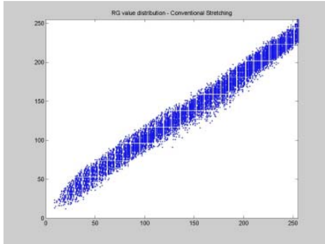
To those we also add the normalized mean value of each color band:

$$\bar{\mu}_i = \frac{\mu_i}{\sum_{k \in B} \mu_k}\,. \qquad (7)$$

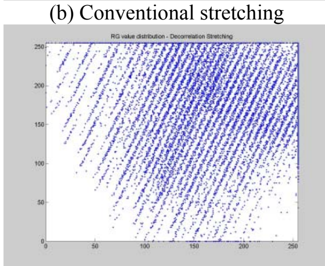
For a 3-dimensional color space (e.g., RGB) this yields a feature vector of 21 elements. Further details on how well these features perform are given in section IV.



(a) Original



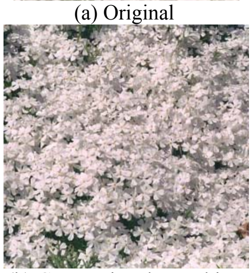
(b) Conventional stretching



(c) Decorrelation stretching

Fig. 1 – Distribution of RG pixel values on the [0..255]2 box. While conventional stretching extends the dynamic range on each axis separately, only a small portion of the plane (two dimensional range) is used. Decorrelation stretching allow for better utilization of the true 2D dynamic range.


(a) Original


(b) Conventional stretching


(c) Decorrelation stretching

Fig. 2 – The effect of conventional and decorrelation stretching on a texture image (Flower0005). images that went through decorrelation stretching better utilize of the color plane and are more descriptive.

III. PRE-PROCESSING ALGORITHMS

A. Decorrelation Stretching

Decorrelation stretching [5] was introduced as a method for enhancing multi-spectral images interpretability on RGB displays. As in the case of gray-scale images, contrast stretching improves image interpretability by expanding the dark-light range of intensities. For color images, a similar result is desirable for the range of colors. Fig. 1a illustrates the distribution of RG pairs in the $[0, 255]^2$ box for a natural color image (Flower0005). It can be seen that the color points are highly correlated and concentrated around the line R=G. Furthermore, a very small subset of the dynamic range ($[0, 255]^2$ box) is used. The conventional contrast stretching method will give poor results in this case as illustrated in Fig. 1b.

Decorrelation stretching deals with the caveats of highly-correlated data by projecting it on its principal axes.

This is achieved by applying the well known Karhunen-Love transform (KLT). Once projected, a conventional stretching method is applied in order to make the data fill the plane. Finally, the inverse KL transform is applied to obtain a representation in the RGB color space. The improved result is illustrated in Fig. 1c. The original image as well as its conventional stretched and decorrelation stretched versions are presented in Fig. 2.

B. Cartoon-Texture Decomposition and Texture Sharpening

In the second pre-processing approach we examine texture sharpening. This approach follows the intuition that if images are to be distinguished by their embedded textures, making those textures more apparent and noticeable would enable better differentiation. This clearly would have been the case had we given the collection of images to a human classifier.

We model the image f as an additive composition of two components, namely $f = u + v$. The first component u is a piecewise-smooth cartoon component while the second component $\boldsymbol{v}$ accounts for the texture and noise. Given the decomposition of a given image f into u and v, we re-mix those components into a new image in which the texture component is amplified $g = u + \beta v$ for $\beta \geq 1$.

The general Cartoon-Texture decomposition problem is defined as the following optimization problem:

(8)

where F_1, F_2 are functionals and X_1, X_2 are spaces for which $F_1(u) < \infty$ and $F_2(v) < \infty$ if and only if (u, v) $X_1 \times X_2$. A good model for the problem would be one for which texture components are penalized by F_1 but not by F_2 and vice-versa. While there exist various options for choosing X_1, X_2, F_1, F_2 (See [9] for a survey of the most popular models) we have chosen to use the algorithm described in [9] for its efficient implementation. The algorithm is based on the $H^1 - H^{-1}$ model [12], which optimizes the following term in the Fourier domain:

$$\min_{(u,v)\in H^1\times H^{-1}} (\sigma^4 \int |\omega\hat{u}(\omega)|^2 + \int \left|\frac{\hat{v}(\omega)}{\omega}\right|^2). \tag{9}$$

The solution to this optimization problem is given by low-pass and high-pass filtering, that is $\hat{u} = \hat{L}_\sigma f$ and $\hat{v} = \hat{f} - \hat{u}$, where

$$\hat{L}_\sigma(\omega) = \frac{1}{1 + (\sigma|\omega|)^4} . \tag{10}$$

The solutions for the $H^1 - H^{-1}$ model in the image domain are:

$$(u, v) = (L_\sigma * f, f - L_\sigma * f) . \tag{11}$$

The σ parameter is a threshold for deciding whether a given frequency ω belongs to u or v. Frequencies significantly smaller than σ^{-1} will be passed only by the low-pass filter L_σ and will therefore appear in u. Similarly, frequencies significantly larger than σ^{-1} will be passed only by the high-pass filter and will therefore appear in v. A shortcoming of this model is that it does not distinguish between edges and texture as both are considered to be of high frequencies.

The algorithm in [9] is based on the $H^1 - H^{-1}$ model and suggests a non-linear filtering scheme, which better distinguishes between edges and texture. For each pixel in the image, a characteristic value is derived for determining whether it belongs to a cartoon or textured region. The local total variation (LTV) of a pixel x in an image f is defined for an averaging kernel G as:

$$LTV_f(x) = (G * |\nabla f|)(x) . \tag{12}$$

The assumption is that for cartoon regions, the local total variation does not decrease by low-pass filtering whereas for textured regions the local total variation decreases very fast under low-pass filtering. The relative reduction rate $\lambda(x)$ is defined as:

$$\lambda(x) = \frac{LTV_f(x) - LTV_{L*f}(x)}{LTV_f(x)}, \tag{13}$$

where L_σ is a low-pass filter. If $\lambda(x)$ is close to 0 the relative reduction is small, which is an indication that x belongs to a cartoon region. When $\lambda(x)$ is close to 1 the relative reduction is rather large, which indicates that x belongs to a textured region. The cartoon and texture components are constructed as a weighted average of the original and low-pass versions of the image:

$$\begin{aligned} u(x) = w\big(\lambda(x)\big)(L_\sigma * f)(x) \\ + \Big(1 - w\big(\lambda(x)\big)\Big) f(x) v(x) \\ = f(x) - u(x) . \end{aligned} \tag{14}$$

$w(x)$ is a soft threshold function and defined as follows for the experiments:

$$w(x) = \begin{cases} 0 & x \leq 0.25 \\ (x - 0.25)/0.25 & 0.25 \leq x \leq 0.5 \\ 1 & x \geq 0.5 \end{cases} . \tag{15}$$

The final step is the re-combination of cartoon and texture components to get a texture amplified (sharpened) version of the original:

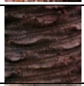
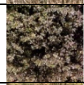
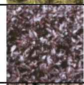
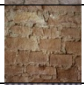
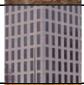
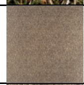
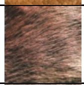
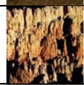
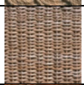
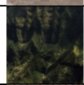
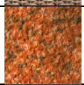
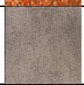
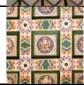
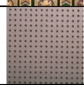
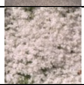
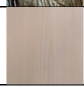
$$g = u + \beta v , \beta > 1 . \tag{16}$$

In the experiments we used a gain factor of $\beta = 4$.

IV. EXPERIMENTAL EVALUATION

We now describe an experimental evaluation to confirm the relevance and benefits of our proposed preprocessing methods.

TABLE I : CLASSIFICATION EXPERIMENT ACCURACIES (PERCENTAGE) FOR ORIGINAL (RAW), DECORRELATION STRETCHED (DS) AND TEXTURE SHARPENED (TS) SUB-IMAGES.

Image		*RAW*	*DS*	*TS*	Image		*RAW*	*DS*	*TS*
	Bark.0000	31.25	100.00	37.50		Food.0008	87.50	81.25	93.75
	Bark.0006	100.00	100.00	100.00		Grass.0001	100.00	100.00	93.75
	Bark.0008	81.25	87.50	81.25		Leaves.0008	100.00	100.00	100.00
	Bark.0009	75.00	87.50	93.75		Leaves.0010	87.50	100.00	75.00
	Brick.0001	81.25	100.00	93.75		Leaves.0011	100.00	100.00	100.00
	Brick.0004	43.75	75.00	75.00		Leaves.0012	93.75	50.00	75.00
	Brick.0005	75.00	100.00	87.50		Leaves.0016	62.50	100.00	100.00
	Buildings.0009	68.75	100.00	68.75		Metal.0000	100.00	100.00	100.00
	Fabric.0000	93.75	87.50	100.00		Metal.0002	100.00	100.00	100.00
	Fabric.0004	62.50	75.00	81.25		Misc.0002	93.75	100.00	100.00
	Fabric.0007	93.75	100.00	93.75		Sand.0000	100.00	100.00	100.00
	Fabric.0009	100.00	100.00	100.00		Stone.0001	75.00	81.25	75.00
	Fabric.0011	93.75	100.00	87.50		Stone.0004	87.50	93.75	68.75
	Fabric.0014	93.75	100.00	100.00		Terrain.0010	93.75	100.00	87.50
	Fabric.0015	100.00	100.00	100.00		Tile.0001	62.50	100.00	93.75
	Fabric.0017	93.75	100.00	100.00		Tile.0004	100.00	100.00	100.00
	Fabric.0018	100.00	93.75	100.00		Tile.0007	100.00	100.00	100.00
	Flowers.0005	93.75	93.75	100.00		Water.0005	75.00	75.00	93.75
	Food.0000	100.00	100.00	100.00		Wood.0001	87.50	100.00	68.75
	Food.0005	100.00	100.00	100.00		Wood.0002	100.00	100.00	100.00
						Average	**87.19**	**94.53**	**90.62**

A. Experimental Setup

To create our experiment dataset we have used 40 512 × 512 images from the VisTex [13] database. Each image is split into sixteen 128 × 128 non-overlapping sub-images all of which are considered to belong to the same texture class. The experiment was conducted on the original, decorrelation stretched and texture sharpened versions of each sub-image. Prior to feature extraction we have converted each image from the RGB space to the K-L space [14] by applying the following transformation matrix:

$$T = \begin{pmatrix} 1/_3 & 1/_3 & 1/_3 \\ 1/_2 & 0 & -1/_2 \\ -1/_2 & 1 & -1/_2 \end{pmatrix}. \qquad (17)$$

This color transformation was used in previous segmentation and texture related work ([14], [2]) for decorrelating the color information in the RGB space. In our experiments, extracting features in the K-L space gave the best results among other spaces we have examined such as RGB, YIQ, Lab, CIE-XYZ.

From each sub-image we have extracted the 21 elements feature vector discussed in Section II. A 16-Nearest-Neighbor classifier with Euclidean vector distance is then used for classifying each sub-image. A sub-image is considered to have 100% or 0% accuracy rate depending on whether it was successfully classified to its original parent image or not.

B. Results

Table I shows the classification accuracy rates achieved for the original, decorrelation stretched and texture sharpened versions. Both decorrelation stretching and texture sharpening have improved the average accuracy by 7% and 4% respectively. Examining individual image classes reveals that the contribution of decorrelation stretching is more significant for monochromatic and highly color-correlated images (Bark.0000, Buildings.0009, Tile. 0001).

V. CONCLUSIONS AND DISCUSSION

Most of the previous work in the field of texture classification had focused on the process of feature extraction. While choosing the right features is crucial for the performance of a classifier, pre-processing the images and preparing them for the classification task is shown in this paper to have a major impact on the classification results. In our experiments we have shown that a rather common texture classification algorithm based on a 21-elements RGB cross-correlation feature vector can be significantly enhanced by applying low complexity pre-processing tools to the input images. This calls for the application of the pre-processing approach as a means for improving the performance of fast texture classifiers.

Both methods proposed in this paper were developed originally for improving the ability of a human viewer to interpret images. Our conclusion is that this new approach to texture analysis could practically become an integral part of most texture recognition classifiers as it adds only negligible complexity to the process while significantly improving the classification results.

REFERENCES

[1] G. Healey and L. Z. Wang, “Illumination-invariant recognition of texture in color images,” *Journal of the Optical Society of America*, vol. 12, no. 9, pp. 1877–1883, Sep. 1995.

[2] G. V. D. Wouwer, P. Scheunders, S. Livens, and D. V. Dyck, “Wavelet Correlation Signatures for Color Texture Characterization,” *Pattern Recognition*, vol. 32, no. 3, pp. 443–451, Mar. 1999.

[3] C. Palm, “Color Texture Classification by Integrative Co-occurrence matrices,” *Pattern Recognition*, vol. 37, no. 5, pp. 965–976, May 2004.

[4] R. M. Haralick, K. S. Shanmugan, and I. Dunstein, “Textural Features for Image Classification,” *IEEE Transaction on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[5] A. R. Gillespie, A. B. Kahle, and R. E. Walker, “Color enhancement of highly correlated images. I. Decorrelation and HSI contrast stretches,” *Remote Sensing of Environment*, vol. 20, no. 3, pp. 209– 235, 1986.

[6] L. I. Rudin, S. J. Osher, and E. Fatemi, “Nonlinear total Variation Based Noise Removal Algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, pp. 259–268, 1992.

[7] S. Osher, A. F. Sol´e, and L. A. Vese, “Image Decomposition, Image Restoration, and Texture Modeling using Total Variation, Minimization and the H^{-1} Norm,” in *Proc. International Conference on Image Processing*, pp. 689–692, 2003.

[8] Jackie Shen, “Piecewise $H^{-1}+H^{0}+H^{1}$ Images and the Mumford-Shah-Sobolev Model for Segmented Image Decomposition,” *Appl. Math. Res. eXpress*, no. 4, pp. 143–167, 2005.

[9] A. Buades, T. M. Le, J. Morel, and L. A. Vese, “Fast Cartoon + Texture Image Filters,” *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 1978–1986, 2010.

[10] R. Kondepudy and G. Healey, “Modeling and Identifying 3-D Color Textures,” in *Proc. International Conference on Computer Vision and Pattern Recognition*, pp. 577–582, 1993.

[11] S. Nemirovsky and M. Porat, “On Texture and Image Interpolation using Markov Models,” *Signal Processing: Image Communication*, vol. 24, no. 3, pp. 139–157, Mar. 2009.

[12] Y. Meyer, “Oscillating Patterns in Image Processing and Nonlinear Evolution Equations,” *The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, Boston, MA, USA: American Mathematical Society, 2001.

[13] Vistex: Vision Texture Database, MIT Media Lab, [Online]. Available: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/, 1995.

[14] Y. Ohta, T. Kanade, and T. Sakai, “Color Information for Region Segmentation,” *Computer Graphics and Image Processing*, vol. 13, pp. 222–241, 1980.