GENERATING TRUE COLOR PAPER TEXTURES OF HISTORICAL DOCUMENTS

Cláudio S.V.C.Cavalcanti, Carlos A.B.Mello[1] *and Milena P.S.Rodrigues*
Escola Politécnica de Pernambuco – Universidade de Pernambuco - Brazil
[1]cabm@upe.poli.br

ABSTRACT

This paper presents two methods for automatic generation of true color paper textures of historical documents. One uses a 256-gray level image of the texture, some features inherent to the original paper itself and Neural Networks to colorize. The other generates a synthetic version of the texture based on the rebuilding of its histogram. Both methods produce images which are perceptually close to the originals by visual inspection and quantitatively by the use of Analysis of Variance. With the colorizing process it is possible to create a cluster of textures and use it to generate different textures from just one sample. The synthesis process allows the complete generation of the image of the document achieving high compression rates.

1. INTRODUCTION

Texture, although not formally defined, has its major application on 3D computer graphics to give a more natural look to artificial models. In this paper, we analyze two algorithms to generate true color paper texture. The algorithms presented were tested on textures extracted from documents from the end of the 19th century and beginning of the 20th century and they are part of PROHIST Project. The documents are digitized with 200 dpi resolution, in *true color*, and stored in JPEG file format with 1% loss for preservation purposes. Figure 1 presents a sample texture from the bequest.

The major goals of the project are the preservation of the documents and to make easy the broadcast of the files. For this second purpose a system to create synthetic versions of the documents were defined [8]. In that system, the synthesis comes in two ways: 1) thru the superposition of an image with the text of the document with a synthetic texture and 2) with the synthesis of the text image and the texture and further superposition. In the two proposed methods, the texture is synthesized.

It is defined herein two methods for true color texture generation: one by colorizing a gray level version of the texture and another by generating the complete texture using some singular features of the paper. The algorithms are applied to a set of 50 and 200 textures, respectively.

2. MATERIALS AND METHODS

Figure 1 presents a sample paper texture and its RGB histogram. The histogram of each RGB component is a gaussian-like function which can be confirmed by the evaluation of the entropy [1] as it is a measure of deformation from perfect gaussian curves [7].

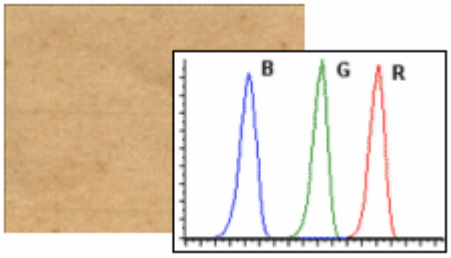

Figure 1. Sample texture and its histogram.

2.1. The Colorizing Algorithm

One can analyze in more details the histograms of the green component and the luminance of the images (Figure 2). It can be noticed that these two functions are very similar as the texture have predominance in the yellow and brown tones. This similarity can be confirmed by the entropy of the two histograms. The entropy values of 50 textures for the green component and the luminance were evaluated and analyzed using ANOVA (Analysis of Variance) which resulted in differences statistically not-significant (Prob>F = 0.2231). The PSNR (Peak Signal-to-Noise Ratio) was also evaluated and the similarities were confirmed again.

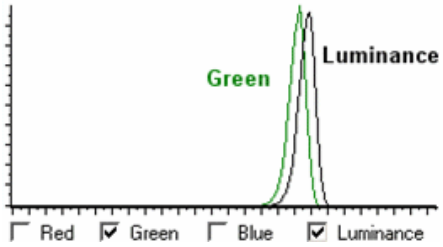

Figure 2. Histograms of the green component and the luminance of the sample texture of Figure 1.

This resemblance is used in the generation of an image with only the green tones. This image is called a *green scale* image which has up to 256 colors and is most likely grayscale images. Instead of storing a grayscale version of the image, we propose the storage of this green

scale image and further the use of the colorizing process as it is described in the next section.

The colorizing scheme works with two synthetic matrixes based on the green scale image defined before.

As shown in Figure 2, the luminance function approaches to the green component's function. The luminance (L) value is evaluated as [9]:

$$L(color) = 0.176R + 0.81G + 0.011B \quad (1)$$

where R, G and B are the red, green and blue components of the color. It can be noticed by this equation that the blue component has little influence on the luminance value. So the behavior of the blue histogram can follow the green one. It is only necessary a displacement (since the blue histogram is darker than the green one as can be seen on Figure 1: the blue's curve is leftmost than the green's curve which is a common feature all textures of the bequest). This means that we only need to store the mean of the blue histogram (*meanB*) in order to generate its complete histogram. Each point *i* of the final histogram of the blue component can be evaluated by the equation:

$$B(i) = G(i) - (meanG - meanB).$$

where *meanG* is the mean of the green histogram and B(i) and G(i) are the blue and green values in the position *i* of each histogram. It must be remembered that meanG does not need to be stored as the green scale image is being used.

The red matrix is the harder one to deal. The first reason is that small differences to the original values result on big statistical differences. Another reason is the difference between the green and red matrixes features. A single displacement (as the one done on the blue matrix) does not work properly. The main idea for facing this problem is the achievement of mathematical equations for expressing the irregular variation of the colors based on its neighborhood.

For the definition of the red tones, a neural network is used. After all the training process, a neural network [5] with two hidden layers should be able to generalize any continuous function [2][6]. If the behavior of the histogram can be represented by a continuous function, this neural network's function may approximate it.

So every pixel of the image and its 8-connected neighbors are going to be used as input to the neural network. Two 3x3 matrixes are used: one for the green components and another for the blue ones.

A MLP (Multi-Layer Perceptron) [5] with 9 nodes at the input layer (for each entry of the matrixes before), two hidden layers with 9 nodes and 6 nodes respectively, and 1 node at the output layer, is designed for the training process. The output of the network is the red value of the pixel. A supervised learning process is then used.

The training and test sets are extracted from each image. For each four masks of 3x3 pixels, three are used in the training set and the other one in the test set for validation. This is repeated for every pixel of the image.

Due to the large amount of data in the training phase, one network is trained using a small set of different samples from three different clusters of textures. The definition of the clusters is based on the most frequent hue value of the textures. Cluster 1 has the textures with hue value less than or equal to 40 (soft brown textures), cluster 2 has textures with hue value between 40 and 50 (dark brown textures) and cluster 3 groups textures with hues greater than 50 (yellow textures) Five images of each cluster are chosen for the training phase.

For the network learning process, it is used the Rprop [10] which is a version of the Back-Propagation algorithm [5]. It is used for the simulation the RpropMap which improves the dynamic decay adjustment to the standard Rprop algorithm. This is a supervised learning algorithm which does not require a validation set.

All the data is normalized in means to maintain the data standards for the neural network. After the training process, the network is ready for the evaluation of the red central pixel of each mask. An image in green scale can have its red component evaluated by this trained neural network.

2.2 The Synthesizing Algorithm

Samples of the texture of 200 color images were collected and analyzed. The mean, standard deviation and entropy of the histograms of R, G and B components were extracted. Figure 1 presented an example of a paper texture and its RGB histogram. As said before, the entropy value can confirm the similarities between the histograms and gaussian functions. This information is used in the automatic generation of the histograms. Even more, as shown in Figure 2, the luminance histogram is also a gaussian-like function. As the textures are in the most part yellowed, it has little influence of the blue tone. So the system generates automatically the histograms of the red and green tones, and the histogram of the luminance. With these values it is possible to evaluate the value of the blue component (from Eq. 1).

The mean and standard deviation values are used to specify the color space of the image which has values between *mean–standard_deviation* and *mean + standard_deviation* for each of the three red, green and gray components (R, G and Gr, from now on). However, not all colors generated in this space are part of the original texture. The *hue* is the feature used to define which colors must be in the texture.

For all the set of 200 images analyzed, it was observed that there is a predominant hue value (called *hue_max*). In some cases, this value is found in more than 40% of the colors of an image. A new image is then created where each pixel has its color determined by a hue value, no longer by its RGB components alone. The *hue_max* of this new image is evaluated and this value is the factor that defines if a color formed by some triple

RGB can be accepted or not in the synthetic image. The most frequent hue value works as a boundary for the possible colors that can be generated in the paper image.

The entropy of the hues (*h_hue*) in the image is also evaluated. As before, the entropy is evaluated using as logarithmic basis the product of the dimensions of the image. The need for h_hue is explained below.

The values of the mean and standard deviation of the histograms of the RGB components, *hue_max* and *h_hue* are all the information needed to create the synthetic image of the texture of the paper. The dimensions of the image are also stored totalizing 40 bytes only.

The histogram of the synthetic image is now created using the mean and standard deviation of the histograms of the original image. The gaussian-like functions for the *RGGr space* is expanded by the RGGr values bounded by $mean \pm factor*|variance|$ for each color tone. Although all the data stored are from the gray histogram, it is possible to evaluate the blue value using Eq. 2, so we are going to consider again the use of the RGB components without any loss. The *factor* variable is defined by an interactive process where a gaussian function is created for the histogram using the mean and standard deviation of each of the RGB components. *factor* reaches its final value whenever the addition of the amplitudes of the histogram is either greater or equal to the number of pixels of the image or it reaches 20% of this number (in this case a correction is done to complete the number of pixels). The use of the variance instead of the standard deviation "stretches" the amplitude of the mean value of the gaussian distribution.

The entropy of the hues is used to determine how the distribution approaches to a gaussian. A triple (R, G, B) from the *RGB space* can be in the final image if its hue value is between *hue_max – delta* and *hue_max + delta*, where if h_hue>0.17, then delta = 10, else delta = 1. These values were found empirically and are called the *hue space*.

Parameter h_hue is also used to define how many colors of the *RGB space* will have the hue defined in the *hue space*. The entropy h_hue and the number of colors in *hue space* are related by a first order function:
*number_colors_in_hue_space =152.64*h_hue + 16.0089*.
Functions of higher orders were tested with no satisfactory results.

Whenever the number of colors in the *hue space* reaches its maximum, the rest of the image is filled with colors from the *RGB space* even in the case they do not belong to the *hue space*.

The color of each pixel is determined by pseudo-random searches in the color table until the maximum number of colors in the *hue space* is reached. The colors are then selected from the *RGB space* with no restriction about its hue value. No structural elements were applied.

3. RESULTS

3.1 For the Colorizing Scheme

An example of the colorizing process can be seen in Figure 3. To analyze the colorized textures, several measures are evaluated from classical ones to new fidelity indexes evaluated between two images (a reference one and a target one). It is evaluated: from [4], Entropy, Angular Second Moment and Inverse Difference Moment; and from [3], Difference and Sum Entropy, MeanX and MeanY, VarianceX and VarianceY, Energy, all based in Gray-Level Co-Occurrence Matrix [9]. The contrast of the images is also analyzed. All these features are evaluated for each of the RGB component and inserted in ANOVA. As the green scale image is the base for the colorization process, only the measures for the red and blue are evaluated.

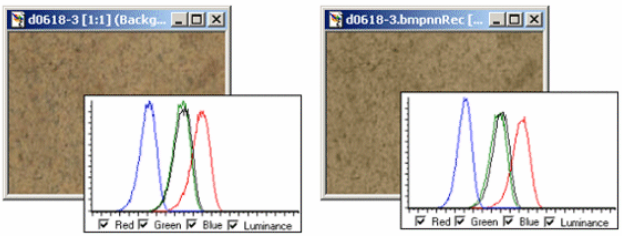

Figure 3. (left) Original texture and (right) colorized one with both histograms.

Table 1 presents ANOVA results where, in the conclusion column, NS means differences statistically not-significant and S means differences statistically significant. The algorithm was applied to a set of 50 samples textures.

A fidelity index defined in [11] is also evaluated. As the final texture is colorized using a green scale image, the comparison between the green histograms of the original and the colorized images is perfect. The blue histograms achieved an almost perfect match for the most part of the textures. For the red histograms, the results were non-conclusives as some images presented high scores and others did not. In means to have a possible comparison, it was evaluated the fidelity index for the textures stored in a GIF file format. This format was chosen because of its particularity of working with 256-color images only. So, it could be used to store the green scale image. In this case, for the fidelity analysis, our process generated colorized textures more similar to the original ones than if they were stored in GIF file format.

3.2 For the Synthesis Scheme

Although visually the synthetic images are far more distant from the original textures than in the colorized version presented in the previous section, their features are quite similar. Figure 4 presents a sample texture, its synthetic

version and their histograms. The textures were analyzed thru ANOVA for the same features as before. The only features that presented differences statistically significants are Entropy (for B), Difference Entropy (for G and B), Sum Entropy (for R, G and B), VarianceX (for B), VarianceY (for B), Contrast (for R and G) and Skewness and Kurtosis (both for Blue component).

Table 1. ANOVA test applied to the red and blue histograms for the colorizing process.

Measure	Red			Blue		
	F(X)	p	Result	F(X)	p	Result
Entropy	6.7318	0.0117	NS	0.0436	0.8353	NS
Angular Second Moment	11.194	0.0014	NS	0.4934	0.4849	NS
Inverse Differential Moment	11.058	0.0014	NS	0.4502	0.5046	NS
Difference Entropy	18.078	6.8e-5	S	0.0627	0.8031	NS
SumEntropy	6.9310	0.0105	NS	0.0234	0.8578	NS
Mean X	0.3580	0.5517	NS	1.1e-4	0.9918	NS
Mean Y	0.3585	0.5514	NS	8.7e-5	0.9926	NS
Variance X	0.5177	0.4744	NS	0.2429	0.6237	NS
Variance Y	0.5311	0.4687	NS	0.2433	0.6235	NS
Energy	11.397	0.0012	NS	0.7742	0.3821	NS
Contrast	2.0653	0.1554	NS	0.0282	0.8670	NS
Skewness	0.9519	0.3328	NS	0.2624	0.6102	NS
Kurtosis	0.8606	0.3570	NS	0.3186	0.5744	NS

4. CONCLUSIONS

Two methods to generate true color paper texture for historical documents were presented. Both of them can be used in a system for automatic generation of a synthetic image of the document in means to easily broadcast a file with thousands images. The first method uses a green scale version of the image and neural networks to colorize this image. The second one completely generates the texture from the synthesis of its histogram and the most frequent hue value to determine which colors belong to the final texture. However, instead of generating the histogram of the blue component, the method builds the histogram of the luminance of the image as there is little influence of the blue color in the textures. With the red, green and luminance histograms, the blue color can be evaluated and the hue defines how the three components can be combined to form the correct colors of the texture. Both algorithms were applied to a set of 50 images achieving high-quality results by qualitative (visual inspection) and quantitative means. ANOVA was applied with the evaluation of thirteen measures analyzed from the original textures and the colorized ones.

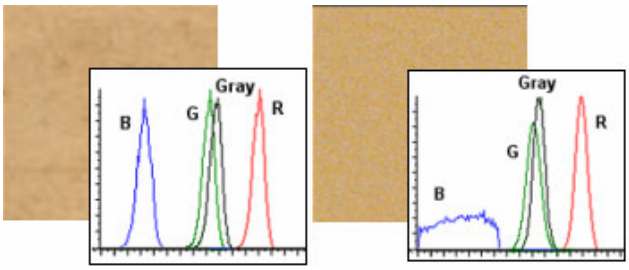

Figure 4. (left) Original texture and its histogram and (right) synthetic texture and its histogram

5. ACKNOWLEDGMENTS

This research is partially sponsored by CNPq (PDPG-TI 55.0017/2003-8), FACEPE (BIT – Bolsa de Incentivo Tecnológico) and University of Pernambuco (PIBIC-CNPq/UPE).

6. REFERENCES

[1] N.Abramson. Information Theory and Coding. McGraw-Hill Book Company, 1963.

[2] G. Cybenko. *Approximation by superpositions of a sigmoid function*. Math.of Control Signals and Systems, Vol 2, 1989.

[3] K.Franke, O.Bunnemeyer, T.Sy. *Ink texture analysis for writer identification*. 8th Int. Work. on Frontiers in Handwriting Recognition, Ontario, Canada, 2002.

[4] R.Haralick, K.Shanmugam and I.Dinstein. *Textural Features for Image Classification*. IEEE Trans. on Systems, Man and Cybernetics, November, 1973.

[5] S.Haykin. Neural Networks: A Comprehensive Foundation. Prentice-Hall, 1998.

[6] J. Hertz, A. Krough and R. G. Palmer. *Introduction to the theory of Neural Computation*. Santa Fe Inst. Lecture Notes in Science of Complexity, Addison-Wesley Publ. Co.,1991.

[7] J. N. Kapur, Measures of Information and their Applications, John Wiley and Sons, 1994.

[8] C.A.B.Mello. *Synthesis of Images of Historical Documents for Web Visualization*. IEEE International Multi-Media Modeling Conference, Brisbane, Australia, 2004.

[9] J.R.Parker. Algorithms for Image Processing and Computer Vision. John Wiley and Sons, 1997.

[10] M. Riedmiller and H. Braun. *RPROP A Fast Adaptive Learning Algorithm*. Proc. of the Int. Symp. on Comp. and Information Sciences, Turkey, 1992.

[11] Z.Wang, A.C.Bovik and L.Lu, *Why is image quality assessment so difficult?*. IEEE ICASSP, Florida, USA, 2002.