Algorithm for Adaptive Color KLT of Images based on Histogram Matching of the Color Components

Peter N. Ivanov[1], Roumen K. Kountchev[2] and R. Mironov[3]

Abstract – **In this paper is proposed a new algorithm for Adaptive Color Karhunen-Loeve Transform (ACKLT) of RGB components, based on pre- and post-processing, depending of the their power distributions. The pre-processing of the components is made by the histogram of the most powerful component which is used as reference image for the transformation of the other components by the histogram-matching. On the already transformed components is applied ACKLT. Through the proposed algorithm for color transformation we increase the power in the first component at the expense of the others hence we can compress more efficiently the color components. The results demonstrated by the experiments made with the modeling of the algorithm are confirming the advantages.**

Keywords – **Color systems, Adaptive color Karhunen-Loeve transform, Representation of color images components, Histogram matching, Power color components distribution, Enhanced Adaptive Color Transform (EACT).**

I. Introduction

The transformation of the primary color space *RGB* is a very important part of the image processing. In general the color components in an *RGB* color system are very much correlated. The most important feature of the new EACT (Enhanced Adaptive Color Transform) [1] is the decorrelation of the color components. The proposed In this paper algorithm gives predictability of the power distribution of the color components in the new format. Hence the applications of the method are many: lossy and lossless image compression, color image segmentation, image recognition, and others.

There are two types of color spaces - deterministic and statistical. The deterministic transforms such as *YCrCb*, *YUV*, *YIQ*, *Lab*, *CMYK* [3, 4, 7, 8] are calculated by using fixed coefficients – hence they require less computations in order to make the color transform but the disadvantages are that they are not adapted to the image specific that is being transformed. Also they are unpredictable in terms of knowing which transformed component is the most powerful or in other words which component carries most information. The proposed algorithm has excellent predictability in terms of power distribution i.e. the first component is the most powerful then comes the second then the third etc. Also the EACT is adapted to the statistical properties of every image or group of images that is being transformed hence the transform is more accurate and from that follows the better quality of the restored image in comparison with the deterministic transforms [2] or the power distribution that we obtain by using the ACT algorithm without the pre-processing techniques [3, 7]. The disadvantage is that the algorithm requires a large number of computations for the pre- and post-processing of the color components.

The purpose of this paper is to improve the already created algorithm Adaptive Color Transform (ACT), by using the method for histogram matching as pre-processing technique on selected color components. The goal is to achieve better power distribution of the components which is with very good predictability, something that can be further improved.

This paper is organized in the following manner: in part two we give the algorithm description, in part three are the experimental results in the forms of analysis, graphics and tables. The final forth part gives the conclusions.

II. Algorithm Description

The proposed algorithm designed as an improvement of the already created algorithm ACT [1, 2]. It is also important to mention that this algorithm is a complete analytical solution to the problem of the color KLT, presented in [2]. The algorithm is simplified to reduce the necessary computations of the color transform.

Transforming an image into the new color format is made by the following steps, which are the forward part of the EACT.

Step 1: Histogram analysis and selection of the base component.

In order to select the base component which we will use in the following step as a reference image for the histogram matching we must perform histogram analysis of the power contained by all the components. For that purpose we use the following formulas:

$$Pow(R)=\sum_{i=0}^{S}R_i^2,\ Pow(G)=\sum_{i=0}^{S}G_i^2,\ Pow(B)=\sum_{i=0}^{S}B_i^2$$

Where R_i, G_i, B_i are the respective color components in the *RGB* image, "*i*" is the current pixel index and *S* is the total number of pixels in the image. For base component we select the component *R*, *G* or *B* for which the power function has greatest value. For that component we don't perform histogram matching, because it is used as a reference image in the next step.

Step 2: Histogram matching and output array expansion.

Let us consider the two histograms $h_{in}(k)$ and $h_{out}(r)$ where the first one is the component that have to be transformed and the second is the histogram of the reference

[1]Peter Ivanov is PhD student in the Faculty of Telecommunications, Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, e-mail: peter.n.ivanov@gmail.com

[2,3]Roumen Kountchev and R. Mironov are professors in the Faculty of Telecommunications, Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, e-mail: rkountch@tu-sofia.bg, rpm@tu-sofia.bg.

component we selected. Respectively $l = r = 0,1,...,L-1$ where they are the discrete levels in the two histograms.

In accordance with the method "histogram matching" of a histogram in an image based on the intensity of every pixel with a discrete level "*k*" in first is applied the following non-linear transform:

$$g_1(k) = (L-1)\sum_{l=0}^{k} h_{in}(l) \quad \text{for} \quad k = 0,1,..,L-1,$$

where L is the number of discrete levels in the image. The function $g_1(k)$ is the first non-linear transform where $h_{in}(k) = n(k)/S$ is the intensity histogram, $n(k)$ is the number of pixels in the image with the same value (intensity) "*k*", and "*S*" is the total number of pixels in the image. This action is known "histogram equalization" or "histogram normalization". On the function $g_1(k)$ we must apply another non-linear transform in order to reshape the original histogram. The relation between the level "*r*" of the transformed intensity and the original level "*k*" is defined by the expression:

$$r = g_2^{-1}[g_1(k)] = g_2^{-1}[(L-1)\sum_{l=0}^{k} h_{in}(l)] \quad \text{for } k = 0,1,..,L-1,$$

The function $g_2(r)$ is calculated in the same manner as $g_1(k)$.

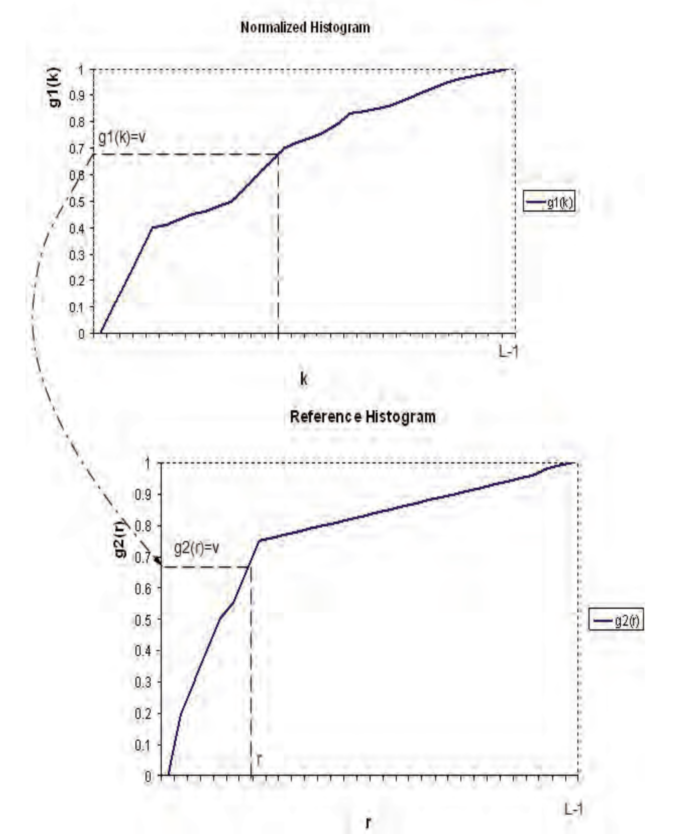


Fig 1. Histogram Matching

We must add that the two functions $g_1(k)$ and $g_2(k)$ must comply with the condition: $0 \le g_1(k), g_2(r) \le 1$ or that the two functions must be monotonically increasing and their values are always between 0 and 1 (because they are normalized).

Before actually applying the method described above, we must add a little customization. In the general case an image in the format RGB is with 24 bpp (bits per pixel) size or 8 bpp per component, so the discrete levels in every component are 256. The use of 256 discrete values in the histogram matching will generate a lot of errors in the restoration of the image. To avoid that, we must increase the number of discrete values in the transformed components. It is a good choice to have 10 bits or 1024 different values for the components that are to be transformed. So, to comply with that condition, we must expand the histograms by linear interpolation of the elements. To do that we must look at the equation $256 + 3 \times 256 = 1024$, hence to every two components in the original histogram we must add another three components. The best choice is to add them in between the known two. Let us consider two neighbor elements in the already normalized histogram $g_1(k)$ or $g_2(k)$, let $g_1(i) = a$ and $g_1(i+1) = e$ where "*i*" is a random number in the interval $0 \le i \le L-1$ and $a \le e$ (which is a given because the function is monotonically increasing). So, let the expanded histogram be denoted as $g_{1\exp}(j)$ where $0 \le j \le 4L-1$ and then let us consider a small section of the histogram, for example the section of elements a,b,c,d,e where "*a*" and "*e*" are already know so the others must be calculated by using the equations: $c = \frac{a+e}{2}, b = \frac{c+a}{2}, d = \frac{c+e}{2}$ and the result is in Fig 2.

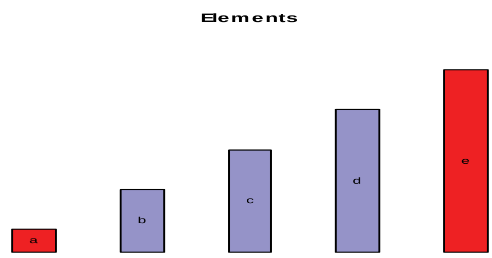


Fig 2. Histogram Expansion

In the end we have a function $g_{1\exp}(j)$ with the same shape as $g_1(r)$ but with more values. The same operation must be applied to the base component histogram in order to have two arrays with the same size for the process of histogram matching.

After completing the histogram matching we will have all the three components of the *RGB* image in the same shape. It is important to note that the base component which is selected as the most powerful component is not transformed by the method. It is passed to the EACT coder as it is. Only the other two components are transformed and expanded. In addition to that we must ensure that the reverse functions can be applied – so we must write into the header the original histograms of the two transformed components in order to be able to construct the inverse functions [3, 7].

Step 3: Calculation of the primary color vectors $\vec{C}_s = [R_s, G_s, B_s]^t$ for each pixel $s = 1,2,..,S$ from the already pre-processed image, where $S = M \times N$ pixels and M and N are respectively the height and the width of the image [1, 2].

Step 4: Calculation of the image covariance matrix $[K_C] = \frac{1}{S}\sum_{s=1}^{S} \vec{C}_s \vec{C}_s^{\,t} - \vec{\mu}\vec{\mu}^t$, where $\vec{\mu}_s = \frac{1}{S}\sum_{s=1}^{S} \vec{C}_s$. It is important to mention that the matrix is symmetric across the main diagonal hence the eigenvalues are always real numbers [1, 2].

Step 5: Calculating the coefficients of the characteristic equation of the covariance matrix $[K_C]$ [1, 2].

Step 6: Calculation of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the characteristic equation defined in the previous step. Given that the matrix $[K_C]$ is a symmetric matrix the eigenvalues can be defined by the "Cardano" relations or the so called trigonometric solution, where we have the condition [1, 2]:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$$

From where we have the predictability of the power distribution. The first component has more power or carries more information than the second etc.

Step 7: Calculation of the eigenvectors $\vec{\Phi}_1, \vec{\Phi}_2, \vec{\Phi}_3$ of the covariance matrix $[K_C]$ and from them forming the transformation matrix $[\Phi]$ [1]:

$$[\Phi] = \begin{bmatrix} \vec{\Phi}_1^t \\ \vec{\Phi}_2^t \\ \vec{\Phi}_3^t \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix}$$

Step 8: Performing the color transform $\vec{L}_s = [\Phi](\vec{C}_s - \vec{\mu})$ using the already generated transformation matrix $[\Phi]$ to obtain the transformed color vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^t$ where s is the current pixel in the image [1, 2].

The restoration to the primary format *RGB* is made by two steps. The first one is by applying the inverse ACT the method of which is described in [2]. The second step is applying inverse histogram matching in order to restore the original color values. For this we use the already stored in the file header original histograms of the transformed components and we use them as reference images in the same way that we used the base component in the forward transformation part of the algorithm. It is important to mention that we don't actually need to store the whole two components. We need to store at least one of them and the other can be represented as a difference array. Also, we can apply any form of lossless compression on them in order to reduce their size.

The improvement of the algorithm presented in this paper shows great results in terms of power distribution in the color components, quality of the restored images and excellent predictability of the power distribution in the transformed matrices. More detailed results are shown in the next section which is the experimental results.

III. Experimental Results

As experimental dataset was used three different sets of images of different size – the "Kodak" image set plus the image "Lena" and "Barbara" 26 images in total, of size 512×768 or 768×512, Lena - 512×512 and Barbara - 640×512 pixels. The "cgraph" image set which comprises ten computer generated images of size 1024×768. The "natural" image set which is comprises 10 images of HD size, 1920×1024 pixels. All the images are in the format .bmp (primary format for the RGB color system) with 24 bits describing each pixel (bpp).

For quality measurement was used the Peak Signal to Noise Ratio (PSNR) - equation below. The PSNR gives the objective representation of the restored image quality [3].

$$MSE = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(X_{i,j} - \hat{X}_{i,j})^2$$

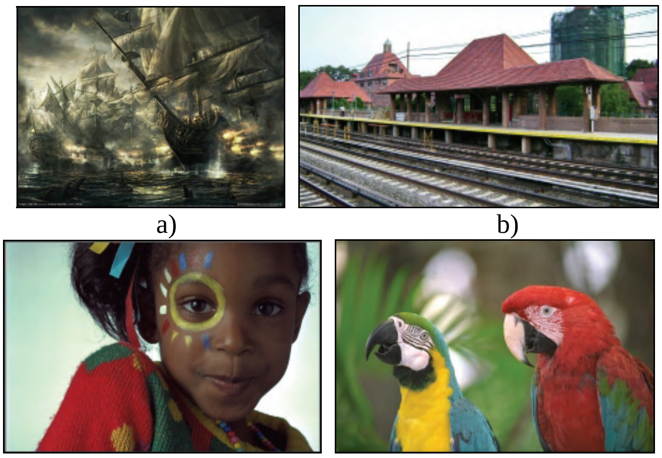
Where the MSE is the Mean Square Error of one component of the *RGB* image. The value $X_{i,j}$ is the original value from the original image and $\hat{X}_{i,j}$ is the restored value in the point *(i, j)* of the image for the given component *R*, *G* or *B*. Therefore there are three different *MSE*s which represent the error in the image. So to find the total error we must add them $(MSE_R + MSE_G + MSE_B)$ [3,7]:

$$PSNR = 10\log_{10}\frac{255^2 \times 3}{MSE_R + MSE_G + MSE_B} \quad \text{[dB]}.$$

For measurement of the power distribution we use the following formula:

$$Pow(L_k) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(L_{i,j})^2 \text{ for } k = 1,2,3$$ and where M and N are respectively the width and the height of the image and $L_{i,j}$ is the respective value from the k-th matrix.

In Fig. 3 are shown some images from the image toolboxes.



a) b) c) d)

Fig. 3 Images from the image toolboxes: a) cgraph1, b) natural7, c) kodim15, d) kodim23.

TABLE 1.

EACT MEAN PSNR VALUES FOR ALL THE TOOLBOXES

Toolbox	EACT Mean PSNR, dB
Kodak + Lena +Barbara	50.79
Cgraph	51.24
Natural	50.17

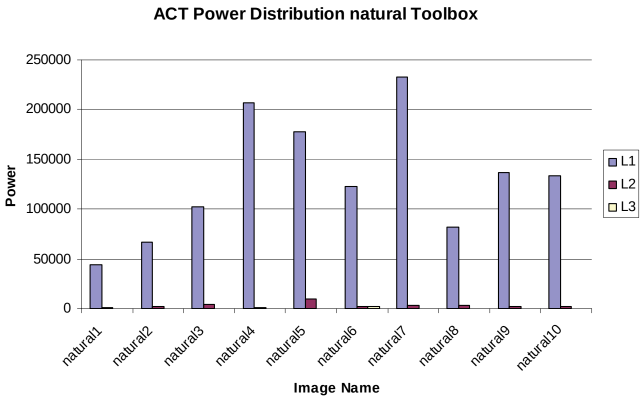


Fig 4. Power Distribution “Natural Toolbox” EACT

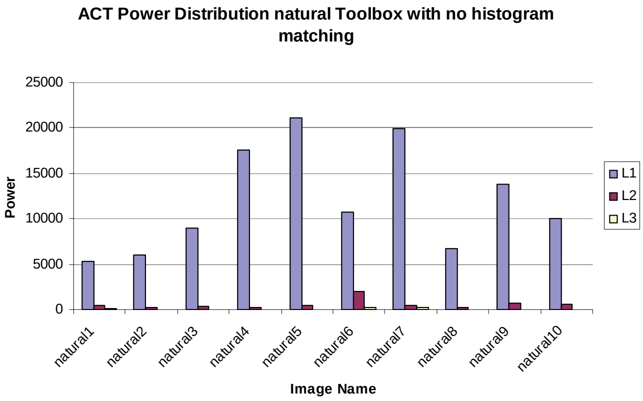


Fig 5. Power Distribution “Natural Toolbox” ACT without histogram matching

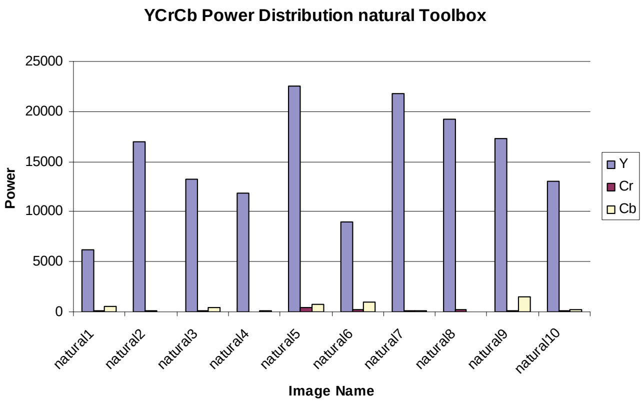


Fig 6. Power Distribution “Natural Toolbox” YCrCb

From the chart in Fig. 4 we can see that power distribution in the transformed components strongly favors the $[L_1]$ matrix, and then comes the $[L_2]$ and the $[L_3]$ matrix is not visible in the charts. From here we can tell that the power distribution that we obtain from the improved algorithm presented in this paper is with excellent predictability in terms of the power distribution as we mentioned above. We can also make a comparison with the power distribution of the ACT Algorithm presented in [1, 2] and the results prove the effectiveness of the new algorithm, Fig 5 and the YCrCb power distribution mentioned above in section two, Fig 6. Also the power ratio between the transformed components is very high. We must add that the quality of the restored images is very good as we see from Table 1.

IV. CONCLUSION

In the proposed algorithm the transformation is adapted for each image that is being transformed therefore the transform matrix generated by the algorithm is adapted to the statistical information of the image hence the transform is more accurate. Also we must add that the demonstrated in the previous part power distribution in the matrices allows the algorithm to be used very effectively for image compression and/or processing. The results of the algorithm in terms of image quality and processing time can be further improved by finding an efficient way to decide when the histogram-matching will be an advantage. Something that can be described in future papers.

ACKNOWLEDGEMENT

This paper was supported by the National Fund for Scientific Research of the Bulgarian Ministry of Education and Science, Contract DNTS 02/19.

REFERENCES

[1] R. Kountchev, R. Kountcheva. "New Method for Adaptive Karhunen-Loeve Color Transform". Proc. of the 9th Intern. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS’09) IEEE, Nis, Serbia, Oct. 07-09, 2009, pp. 209-216.

[2] P. Ivanov, R. Kountchev, "Comparative analysis of Adaptive Color KLT and YCrCb for representation of color images", XLV Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST’10), Ohrid, Macedonia, 23-26 June 2010, SP I, pp. 29-32.

[3] D. Carevic, T. Caelli, “Region based coding of color images using K-L transform, Graphical Models and Image Processing”- Vol. 59, No.1, 1997, pp. 27-38.

[4] W. Pratt, “Digital Image Processing”, Wiley Interscience, New York, 2007.

[5] M. Fleury, A. C. Downton and A. F. Clark, “Karhunen - Loeve Transform - Image Processing”, University of Essex, Wivenhoe Park, UK 1997.

[6] R. Dony. "Karhunen-Loève Transform". Book Chapter in The Transform and Data Compression Handbook, Ed. K. Rao and P. Yip, Boca Raton, CRC Press, 2001.

[7] R. Gonzalez, R. Woods, “Digital Image Compression” Prentice Hall, Second Edition, 2002.

[8] S. Sangwine, R. Horne, "The colour image processing handbook", Chapman & Hall, 1998.