

## ON IMAGE RECOLORING - COLOR GRADING

Diana-Maria Popa<sup>1\*</sup>

### ABSTRACT

*This paper analyzes an example-based color transfer algorithm. The process of recoloring an input image by making use of another image has many applications in computer vision. This is the case of digital restoration technique that tries to reapply color to paintings that have been deprecated by dust, smoke, passing of time. Throughout history, there have been many ideas proposed regarding color transfer computation.*

**KEYWORDS:** *Color Conversion, Color Mapping, Color Reduction*

### 1. INTRODUCTION

The focus of this paper is to analyze and implement an algorithm that recolors an input image based on a palette image given by the user [2] [3]. The technique of transferring the feel of a second image to a source image, by keeping the dynamics of the initial one, is known as color grading. Basically, the color of an input image is adjusted in order to match illumination and atmosphere in the palette given also as an input, and thus obtaining an output image out of the combination of the two. This technique represents one of the fundamental process in film grading. Most of the time this issue is fixed by professional video post-production experts manually. The color of an image can be so important that special techniques were developed in order to correct slight aberrations, using a predefined target [25]. A recent article published by ‘The Wall Street Journal’ in 2011 [20], lists the Video Post-Production Services in the ‘Top 10 Dying Industries’:



Figure 1. Example-based recoloring

---

<sup>1\*</sup> corresponding author, Engineer, Politehnica University of Bucharest, 060042 Bucharest, Romania, diana.popa@cti.pub.ro

“...The widespread adoption of digital media have adversely affected the industry’s range of services, from editing and animation to archiving and format transfer.”

This operation has also been referred throughout as example based re-coloring stylization [10]. The idea of example-based recoloring is better described by Figure 1. The first ‘Gladiator’ picture has to undergo a transformation so that its colors match the palette of the second Gladiator picture, regardless of what the first one may contain.

This paper concludes the work done in the license thesis presented at the faculty of Automatics and Computers from the “Politehnica” University of Bucharest by the author of the paper [28] and continues the work presented in [30].

## 2. RELATED WORK

### Transfer of color statistics

One of the most popular methods proposed was that of Reinhard [4] whose strategy was to choose a suitable color space and then to apply a simple operation. By suitable color space, Reinhard meant that they had to focus on an orthogonal color space without correlations between the axes, which means that there is very little dependence between the color information. One color space that meets all those requirements is the *Lab* space which firstly, was developed by Ruderman [5] based on data-driven human perception research that claims the human visual system responds the best at processing natural scenes and secondly, it is a space that minimizes correlation between channels. The Reinhard method is based on transferring the mean and standard deviation of data points in the *Lab* space between images along each of the three axes. The advantage of this technique is its simplicity, as it requires only two images to compute the desired result, its efficiency, as the computing time is less than a few minutes or even shorter. Third, it is semi-automatic, demanding very little user intervention. There are also disadvantages to this Reinhard algorithm as it is very dependent on how much the source image and the target image had in common in order to obtain acceptable results. All in all, the Reinhard method remains the first fast and robust color transfer solution for the image processing field and this is the reason why his work is very often referred to in other related research as ‘pioneering work’.



Figure 2. Color transfer between images (image taken from Reinhard [4])

To go deeper into the currently existent color transfer methods, [6] describes a method for Image Sequence Color Transfer (ICST algorithm) where the user is requested to provide an input image and three target images, an integer N, indicating the numbers of image sequence that are to be rendered. Based on all these inputs, the ICST method is able to render an image sequence that illustrates characteristics from all three target images provided.



Figure 3. The input image, three target images, and the resultant basis images in ICST algorithm (image taken from [6])

These two methods described above [5, 6] have a limitation: linear transformations. In all the practical situations where color transfer is needed, all the recoloring techniques require non-linear color mapping.

One way to perform non-linear mapping is histogram matching, sometimes called histogram specification or histogram normalization, which is a basic signal processing technique [7]. It basically refers to the transformation of one histogram into another by remapping the signal values so that it is used to adjust the statistical profile of a dynamic range. Histogram specification has a wide variety of fields in which it makes itself useful: computer vision, remote sensing, medical imaging, speech recognition, scientific visualization. It can also be used for fast image retrieval, for example searching in a large set of images for the ones that best resemble an input image. When it comes to image processing, histogram specification is primarily a means of obtaining image enhancement for visual inspection. Contrast enhancement makes content of images be easier to differentiate, thus more distinguishable.

A more complex type of mapping is treated in [8] where the color histogram equalization is performed via the deformation in color space of a mesh in order for it to fit the histogram of an image and then by applying some specific equation to map it to a uniform histogram.

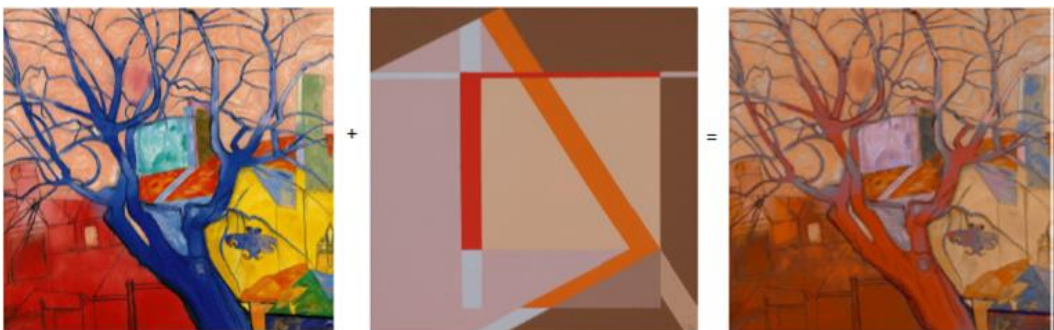


Figure 4. Color Style Transfer Technique (image taken from Neumann [9])

Another color histogram matching, which is also simpler from an implementation point of view and also fast in a computational way is described in [9]. The novelty in this case is introduced by the use of basic perceptual approaches which are: hue, lightness and saturation instead of approaching the classical way of using opponent color channels.

### **Solutions to content variations**

The main problem of color transfer based techniques is the fact that content between input image and palette image may vary quite a lot and thus they do not work well together. For example, if the input image has more sky in it than grass and the palette image has more grass than sky, then the computation of color transfer statistics is expected to fail. One solution to this problem is to select different samples (swatches) of grass and sky from the two images and compute the statistics separately and then render them back together in *Lab* space [4]. However, this is not a very practical solution as it would be very time costly to perform this type of image segmentation in the case of a big images and almost impossible to apply this type of technique when it comes to sequences of images.

A more interesting method was described in [10] where the human color perception is taken into consideration. Each pixel value is classified into one of eleven basic color categories. These categories are derived from psychophysical experiments and are universal to every human being. By using these categories, the mapping of color is more prone to look undistorted and to be free of unnatural results.

Another way to avoid the problem of content variations when it comes to transferring the color between images is to make use of the spatial characteristics. This technique is widely used when transferring color from one colored image to a greyscale one [11]. Given that the greyscale image is represented by a 1D distribution, only the luminance channels can be matched and taking into consideration that a single luminance value could represent multiple regions from an image, another criteria that is left to be used in order to guide the matching process is the statistics within the pixel's neighborhood.



Figure 5. Colors are transferred based on neighborhood statistics (image taken from [11])

As for the method of color transfer that is presented in this paper, the technique is based on an exact color transfer technique and the refinement part that deals with content variations is based on a re-graining process that will be further exposed. Pitié et al. [3] developed a method that succeeded in automating the process of color transfer by using color distribution transfer which represents the basis of the algorithm, and implicitly of the framework that will be presented in this paper.

### 3. RECOLORING ALGORITHMS – THEORETICAL PORTRAYAL

#### Automated color grading using color distribution transfer

##### *Definition and concepts*

Probability Density (Distribution) Function – PDF

By definition, a random variable  $X$  has the property of being **continuous** if the values that it takes can form a “continuum”, meaning that the possible values that  $X$  can take are represented by a single interval on the number line ( $\forall A, B \mid A < B, X$  can be any number between  $A$  and  $B$ ) or that the possible values are formed by a union of disjoint intervals. Moreover,  $P(X=x) = 0$  for any number  $x$  that is a possible value (solution) of  $X$ . The function  $F(x) = P(X \leq x)$  is called the **cumulative distribution function** (CDF).

One of the methods to obtain the PDF of some continuous values (that is also the one used in the algorithm proposed in this paper) is by plotting the histogram. Basically, that means recording the temperature using time slices. The smaller the slices, the finer the approximation will be, but the quantity of data needed will be greater and the finer the curve that will be obtain based on the data gathered.

Definition: Let  $X$  be a continuous random variable. Then a **probability distribution** or a **probability density function** (PDF) of  $X$  is a function  $f(x)$  such that for any two numbers  $a$  and  $b$  with  $a < b$ ,

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{1}$$

In more informal terms, the probability that  $X$  has some value in the interval  $[a,b]$  is represented by the area under the graph and above the mentioned interval. The term of **density curve** is often used to refer the graph of  $f(x)$ . We can also note from the equation (1) that the relation (2) is also valid, where  $f$  is the PDF and  $F$  is the CDF.

$$f(x) = F'(x) \tag{2}$$

Basically the histogram is a continuous set of rectangles that have the width the size of the step and the length is represented by the amount of values from the input data set that fit into that specific bin (characterized by an inferior limit and a superior limit).

#### Color Distribution Transfer

##### *Mathematical View*

In the context of the results that we seek to obtain, the color distribution transfer is based on the transmission of the image statistics (presented above) of the target image (palette) onto the source image, thus obtaining a new image that combines the characteristics of the two (preserves the content of the initial picture and mangles it with the feel of the target).

Following the example of the temperature given above in order to describe the PDF, we can consider the input images as being two sets of random tuples. We say tuples and not

variables, because these tuples are the representation of each pixel of the image which we will consider, for transparent use, to be N-dimensional.

Let's denote these two sets as  $\{s_i\}_{i \leq S}$  and  $\{t_i\}_{i \leq T}$  where S and T are the cardinals (number of total pixels) of the two images (i.e. source image and target image), respectively. One tuple of a RGB color image is 3D dimensional and could very well be represented by:  $s_i = (R_i, G_i, B_i)$ .

**Problem statement:** Find a differentiable bijective mapping function m that takes as input the tuples of the input image and outputs a new set of tuples whose statistics match the statistics of the target tuples.

It is important to notice the problem above is expressed for the continuous case. In the case of image datasets, which are represented by discrete values the solution to this problem is given by making use of the histograms of the two set of tuples. The continuous probability density function (see 3.2.1) that is sought to be used will be replaced by histograms and these will be the one being transferred between the images. This problem was referred in the specialty books as the **Distribution Transfer Problem** [3].

One other aspect to be noted down is that, in contrast to the greyscale algorithm presented in the previous paper, in the case of color transfer it suffices to use the RGB color space in the manipulation of the input data sets. There is no use in transforming it to *Lab, Lαβ, Luv* or any other color space because this algorithm tries the transfer the statistics of every color channel simultaneously. In the case of greyscale conversion, only the luminance channel was being preserved through the manipulation.

***The One-Dimensional case***

The first step to solving the above mentioned problem is to first consider the 1D case, which means to analyze the case where the tuples that form the set of the images are of dimension 1 (the pixel are represented by one channel value).

Considering all the above definitions, we can now transpose the problem into mathematical equations:

Matching the target PDF g with the source PDF f

$$f(s)ds = g(t)dt \tag{3}$$

Finding a mapping function m that maps source to target

$$m(s) = t \tag{4}$$

Integrating the (3) equation gives us:

$$\int^s f(s)ds = \int^t g(t)dt \tag{5}$$

Making use of equation (4) results in:

$$\int^s f(s)ds = \int^{m(s)} g(t)dt \tag{6}$$

Going back to the definition of the cumulative distribution function, and denoting the CDF notations for  $f$  as  $F$  and for  $g$  as  $G$ , we obtain the mapping function which represents the solution to the initial problem statement.

$$G(t) = F(s) \Rightarrow G(m(s)) = F(s) \Rightarrow m(s) = G^{-1}(F(s)), \forall s \in \mathfrak{R} \tag{7}$$

**The N-Dimensional case**

In order to upgrade the solution to a higher dimensional situation, Pitié et al. in [3] invoked the example of the Radon Transform. This Transform, which has been called in the specialty books as the shadow transform (or the X-Ray transform), maps an N-Dimensional function onto one-dimensional lines or axes [21]. In order to make it clearer why the authors used the Radon Transform as a solution to the problem, some concepts and definition have to be introduced. In order to do that, Figure 6 will be used in order to better describe some basic notions that lie at the core of understanding the Radon Transform: line integrals and projections.

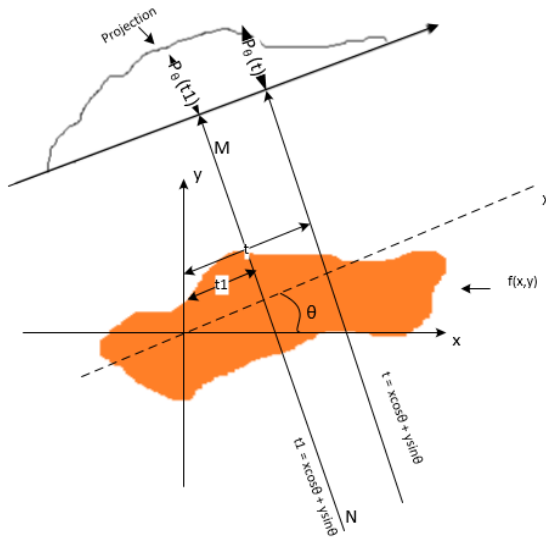


Figure 6. A color sample  $f(x, y)$  and its projection  $P_{\theta}(\theta, t)$  for an angle of  $\theta$

Let's take the example of a color sample from an image which is represented in Figure 6 by a two dimensional function  $f(x, y)$ . The line integrals are denoted by the parameters:  $\theta$  (the angle of inclination from the main axis) and  $t$  which represents the distance from the origin. The notation  $P_{\theta}(\theta, t)$  represents the Radon Transform and, as it can be seen from Figure 6, it is used to calculate the projection of the multi-dimensional function onto some axis at angle  $\theta$ .



This is what inspired Pitié et al. in [3]. The authors based their theory on the fact that after projecting the N-Dimensional PDF function, the result will be a sequence of one-dimensional projection results. Manipulating these series of results for a number of iterations, the one-dimensional projection results of the source PDF, denoted with  $f$  throughout the paper, will finally match those of the target PDF  $g$ . In fact, in specialty books this one-dimensional projection of a N-Dimensional function onto some axis has been referenced as marginal as it can be seen in Figure 7 which is very suggestive for what the logic of the algorithm is. The algorithm has been referenced in specialty books [3] as the Iterative Distribution Transfer (IDT).

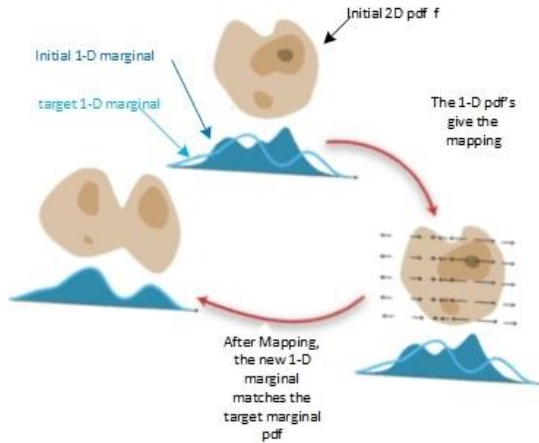


Figure 7. Iterative Distribution Transfer (IDT) based on one-dimensional PDF transfer (image taken from [3])

Let's consider now all that has been said in theory above and analyze how the present algorithm makes use of that. The equivalent of the 2-Dimensional function from above will be the two 3-Dimensional PDFs functions:  $f$  and  $g$ . In order to project them onto some axis, [3] proposes to use an orthogonal Rotation matrix. This rotation matrix is denoted by  $R = [e_1, \dots, e_N]$ , where  $N=3$  and  $e_i$  are the rotated axis from the theoretical portrayal from above. From mathematics, the projection of same tuple  $s = [s_1, \dots, s_n]^T$  (the input sample) is obtained by multiplying some rotation axis with the sample.

$$e^T s = \sum_i e_i s_i \tag{8}$$

After projecting the two PDFs, next step is finding that one mapping by which all one-dimensional marginal projections of the source PDF and all those of the target PDF match. This will be done exactly like in the one-dimensional case, but for convention, we will consider the projection of the PDF  $f$  to be  $f_e$  and that of the PDF  $g$  to be  $g_e$ . By association, the CDF  $F$  is  $F_e$  and  $G$  and  $G_e$ . Thus, equation (8) becomes:



$$m_e(s) = G_e^{-1}(F_e(s)), \forall s \in \mathfrak{R} \quad (9)$$

After finding the mapping function  $m_e$  in the same way as for one-dimensional case, which is through discrete look-up tables, there remains only one step, to add to the initial set of samples (tuples) the displacement along the axis, which is suggested by final step in Figure 7.

$$s = s + (m_e(e^T s) - e^T s)e \quad (10)$$

### ***Implementation***

The implementation of the algorithm described in the next chapter was based on the theoretical portrayal of the Iterative Distributed Transfer proposed by the authors in [3].

### ***Generation of rotation matrixes***

The algorithm that lies at the core of the framework starts by generating the rotation matrixes that were to be used in order to get the projections of both the PDFs,  $f$  and  $g$ . The way they were generated was influenced by the analysis of convergence that is given in [3]. In this chapter, it is mentioned that in order to obtain convergence of the algorithm, which in simpler words mean that at the end of the computation the marginal of the input PDF matches the marginal of the target PDF, it is sufficient to use random orthogonal bases when considering the rotation matrixes. Thus, it was started with a matrix that was known to be orthogonal. In order to obtain new orthogonal rotation bases Gram-Schmidt algorithm was applied on some initially randomly generated matrix and then multiplied these rotation matrix with the initial one. Taking into account the definition of an orthogonal matrix, the result will also remain an orthogonal matrix. As per the algorithm presented above, the number of iterations to be applied on the two one-dimensional PDFs is equal to the number of orthogonal matrixes generated, as in every iterations we look to obtain the projections on every orthonormal axes. An arbitrary number of 20 iterations was used as per the analysis of convergence described in [3]. The user, however, can modify it and observe the difference in the output results as will be show in latter chapters of this paper.

### ***PDF matching***

The next step of the algorithm was to actually make use of the orthogonal matrixes created above and match the input PDF with the target PDF. In the process of creating a probability density function (PDF) for an image, the first step is to create a histogram for every color channel. This was the second most intensively computational part of the program. The histograms were computed by firstly finding the minimum value and maximum value of the input data sets of both images. In order to make the partition of this interval, an arbitrary number is chosen in the framework related to this paper. The above mentioned interval is then divided in equal slices and also the bins are labeled with their inferior limit and superior limit. This way it is possible to place every value from the initial input set in its corresponding bin. While doing this each bin's counter increases accordingly. Resulting from this manipulation is an array of structures (that basically represented the bins) with the length an arbitrary number that contained the number of

values from the data sets in an ascending order. A bin of the histogram revolves around some color value and stores the relative frequency for it.

As per the theoretical portrayal of the IDT algorithm described above, the next step was represented by the one-dimensional PDF transfer. This was done for every RGB channel, as the algorithm expects to receive as inputs two color (RGB) images. In order to get smooth results, the single PDF transfer was done by using linear interpolation. The logic of the function that performs linear interpolation basically solves equation (8) by making use of the two histograms of the source and palette images which are received as inputs. The linear interpolation used in the algorithm implementation plays the role of a discrete look-up table technique. As per the equation that was earlier mentioned, the linear interpolation function is called twice: the first time is called for the  $F(s)$  component of the equation. What happens is that the probability of a color value is searched based on two adjacent bins in the histogram (linear interpolation). Upon retrieving the results, the function in question is called for  $G^{-1}(F(s))$ . In the last case, the technique it is actually called reverse look-up table technique, as this time the probability of a color value is obtain based on the quintile (the inverse function of G is applied). It is important to make the observation that without linear interpolation the resultant image would only contain as many colors as the number of the bins used, by default the algorithm uses 300, but it can be varied by the user.

### **Results and discussion**

The following analysis is based on the two images below. The first one is taken from [22] and the second one from [23].



Figure 8. Source image (left) and target image (right) for color based transfer algorithm analysis

By default the number of iterations is 20 and the bin count is 300. The execution time will also be analyzed in each scenario. The first thing that is worth observing is the influence of the iterations parameter. Figure 9a represents the output when the executable received an input parameter of 5, Figure 9b is run with a parameter of 15 iterations and Figure 9c is run with a parameter of 20 iterations. It is worth describing how much of a visual difference can be noticed between the first two pictures and how little the last two vary in

resemblance. This proves that the algorithm converged earlier than 15 iterations and that continuing to increase the number of iterations would have no further visual impact.

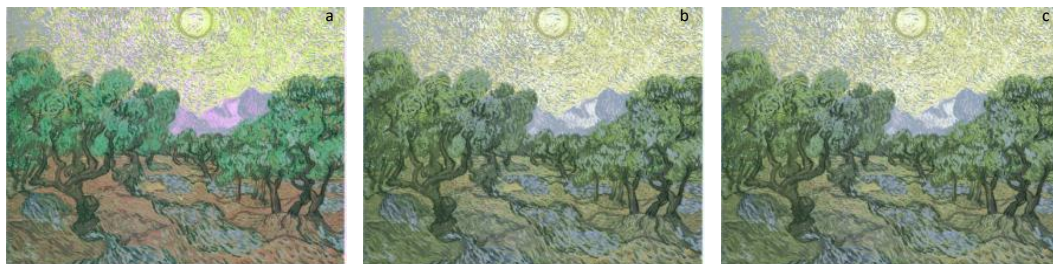


Figure 9. Results for color based transfer algorithm

It can be said that the results proved to be satisfactory but the overall performance of the algorithm could use some improvement. Table 1 below shows how the time varied with the number of iterations.

Table 1. Performance of example-based color transfer

Iterations	2	5	10	15	20
Time	34.86s	83.86s	159.55s	240.75	318.33s

#### 4. CONCLUSIONS AND FUTURE WORK

Image recoloring plays an important role in digital media technology as it finds itself in a continuous and fruitful growth, as can be seen for example in the increasing popularity of digital cameras.

Throughout this paper it has been proven that a simple application that can mimic in a minimal way the behavior of such an image editing tool can be created even with very little open source and third-party support . The framework developed through this paper has several advantages.

Firstly, the framework offers some level of manipulation of the output images. The color transfer executable implemented in this paper gives the user the possibility to vary its iteration parameter. This parameter has a direct impact on how much of the feel, atmosphere of the palette is being transferred to the original picture.

As far as improvement to performance results, the algorithm for color transfer between images could use from finding a solution that deals with content variation. This means reducing the graining appearance that can appear when the dynamic of the color in the source image and target image differs a lot. Another issue that could be further improved and analyzed would be the way the rotation matrixes are being generated in order to obtain an optimal sequence. This plays a major role in the quality of the resultant matrix because basically the finer the choice of axis is the bigger the probability of the mapping function to find the best match.

## 5. REFERENCES

- [1] Gooch, J. Tumblin, B. Gooch and S. Olsen. *Color2Gray: Saliency Preserving Color Removal*. In *ACM Transactions on Graphics*, Volume 24, Issue 3, 2005, pp. 634-369.
- [2] F. Pitie, A. Kokaram and R. Dahyot. *Towards Automated Colour Grading*. In *IEEE European Conference on Visual Media Production*, November 2005.
- [3] F. Pitie, A. Kokaram and R. Dahyot. *Automated Colour grading using colour distribution transfer*. In *Journal of Computer Vision and Image Understanding*, February 2007.
- [4] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley. *Color Transfer between Images*. In *IEEE Computer Graphics Applications*, Volume 21, Issue 5, 2001, pp. 34-41.
- [5] D. Ruderman, T. Cronin, C. Chiao. *Statistics of cone responses to natural images: implications for visual coding*. In *Journal of the Optical Society of America*, Volume 15, Issue 8, 1998, pp. 2036-2045.
- [6] C.M. Wang, Y.H.Huang. *A novel color transfer algorithm for image sequences*. In *Journal of Information Science and Engineering*, Volume 20, Issue 6, 2004, pp. 1039-1056.
- [7] M. Grundland, N.A. Dodgson. *Color histogram specification by histogram warping*. In *Proceedings of the SPIE*, Volume 5667, 2004, pp.610-624.
- [8] E. Pichon, M. Niethammer and G. Sapiro. *Color histogram equalization through mesh deformation*. In *IEEE International Conference on Image Processing*, Volume 2, 2003, pp. 117-120.
- [9] L. Neumann, A. Neumann. *Color style transfer techniques using hue, lightness and saturation histogram matching*. In *Proceedings of Computational Aesthetics in Graphics, Visualization and Imaging*, 2005, pp. 111-122.
- [10] Y. Chang, S. Saito, K. Uchikawa, M. Nakajima. *Example-based color stylization of images*. *ACM Transactions on Applied Perception*, Vol. 2, Issue 3, 2005, pp. 322-345.
- [11] T. Welsh, M. Ashikhmin, K. Mueller, M. Nakajima. *Transferring color to greyscale images*. *Proceedings of ACM SIGGRAPH*, San Antonio, 2002, pp. 227-280.
- [12] R. Bala, R. Eschbach. *Spatial color-to-grayscale transform preserving chrominance edge information*. In *Color Imaging Conference*, 2004, pp. 82-86.
- [13] L. Neumann, M. Čadík and A. Nemcsics. *An efficient perception-base adaptive color to gray transformation*. In *Proc. Computational Aesthetics*, 2007, pp. 73-80.
- [14] K. Smith, P. E. Landes, J. Thollot and K. Myszkowski. *Apparent Greyscale: A simple and fast conversion to perceptually accurate images and video*. In *Computer Graphics Forum (Proc. Eurographics 2008)*, Volume 27, Issue 2, 2008, pp. 193-200.
- [15] Y. Nayatani. *Simple estimation methods for the Helmholtz-Kohlrausch effect*. In *Color Research and Application*, Volume 22, Issue 6, 1997, pp. 385-401.

- [16] K. Yongjin, J. Cheolhun, J. Demouth and S. Lee. *Robust Color-to-Gray via Nonlinear Global Mapping*. In ACM Trans. Graph, Volume 28, Issue 5, 2009, pp. 1-4
- [17] H. Lekowitz. *Color theory and Modeling for Computer Graphics, Visualization, and Multimedia applications*. In Kluwer Academic Publishers, 1997, pp.55-58.
- [18] E. Reinhard and T. Pouli. *Colour spaces for colour transfer*. In Proceedings of the Third International Conference on Computational Color Imaging, Volume 6626, 2011, pp. 1-15.
- [19] M. Grundland, N. Dodgson. *Decolorize: fast, contrast enhancing, color to grayscale conversion*. In Pattern Recognition, Volume 40, Issue 11, 2007, pp. 2891-2896.
- [20] <http://blogs.wsj.com/economics/2011/03/28/top-10-dying-industries/>. “*Top 10 Dying Industries - Real Time Economics – WSJ*”, Accessed on: 27 Jun 2014.
- [21] <http://www.mathworks.com/help/images/radon-transform.html>, “*Radon Transform – Matlab & Simulink*”, Accessed on: 30 Jun 2014.
- [22] Van Gogh, Vincent. Olive Trees. 1889. The Minneapolis Institute of Arts, Minneapolis. <http://www.vangoghgallery.com/catalog/Painting/360/Olive-Trees-with-Yellow-Sky-and-Sun.html>. Accessed on 1 Jul 2014.
- [23] Van Gogh, Vincent. The Olive Trees. 1889. The Museum of Modern Art, New York. <http://www.vangoghgallery.com/catalog/Painting/359/Olive-Trees-with-the-Alpilles-in-the-Background.html>. Accessed on 1 Jul 2014.
- [24] Andrei Tigora, Costin-Anton Boiangiu, “*Image Color Reduction Using Iterative Refinement*”, International Journal of Mathematical Models and Methods in Applied Sciences, Volume 8, 2014, pp. 203-207
- [25] Costin-Anton Boiangiu, Alexandru Victor Ștefănescu, “*Target Validation and Image Color Calibration*”, International Journal of Circuits, Systems and Signal Processing, Volume 8, 2014, pp. 195-202
- [26] Costin-Anton Boiangiu, Alexandra Olteanu, Alexandru Victor Ștefănescu, Daniel Rosner, Alexandru Ionut Egner (2010). „*Local Thresholding Image Binarization using Variable-Window Standard Deviation Response*” (2010), Annals of DAAAM for 2010, Proceedings of the 21st International DAAAM Symposium, 20-23 October 2010, Zadar, Croatia, pp. 133-134
- [27] Costin-Anton Boiangiu, Andrei Iulian Dvornic. “*Bitonal Image Creation for Automatic Content Conversion*”. Proceedings of the 9th WSEAS International Conference on Automation and Information, WSEAS Press, pp. 454 - 459, Bucharest, Romania, June 24-26, 2008
- [28] Diana-Maria Popa, “*Recolorarea Imaginilor*”/”*Image Recoloring*”, License Thesis, Unpublished Work, Bucharest, Romania, 2014

- [29] Costin-Anton Boianiu, Ion Bucur, Andrei Tigora - „*The Image Binarization Problem Revisited: Perspectives and Approaches*”, The Proceedings of Journal ISOM Vol. 6 No. 2 / December 2012, pp. 419-427
- [30] Diana-Maria Popa - “*On Image Recoloring – Part 1: Correct Grayscale Conversion*”, The Proceedings of Journal ISOM, Vol. 10 No. 1 / May 2016 (Journal of Information Systems, Operations Management), pp 222-234.