

REVERSIBLE COLOR TO GRAYSCALE CONVERSION

Cristian Ungureanu¹

Andrei Tigora²

Ion Bucur³

Mihai Zaharescu⁴

ABSTRACT

This article presents an algorithm for conversion of color images into grayscale ones, together with a codification schema which allows the reconstruction of color images. Two versions of the image recoloring are described.

KEYWORDS: image conversion, image reconstruction, grayscale, color

1. INTRODUCTION

Sight is the most important of human senses, accounting for an important percentage of the brain activity at any given time; so there is no wonder that designing image processing algorithms is one of the toughest problems in computer science nowadays. And despite the fact that hardware devices are capable of capturing color images of the real world with great detail and accuracy, in order to assign meaning to the image [4], processing techniques usually deal with simplified versions of the image in the either converted to grayscale or even binarized. Moreover, due to manufacture costs and complexity constraints, some equipment [6][7] is still only capable of rendering single channel content (printing presses, some monitoring systems, etc.) or to process by the nature of the algorithm only single channel of data [5]. Also, there is the possibility of applying grayscale conversion to an image simply for an artistic effect.

Therefore, grayscale conversion, despite the severe information it imposes on the image, is perhaps one of the most frequent image processing operations that is applied in modern systems.

In some cases though, it is desirable to recover the color image from a medium that is not capable of displaying color components, and this is where the color recovery mechanism intervene.

The paper begins with a short presentation of previous work in the domain of grayscale conversion. Sections 3 and 4 are dedicated to a theoretical description of the two versions of the conversion algorithm as presented in the original paper of Queiroz and Braun [1], whereas section 5 presents an improved solution. Section 5 contains results and evaluation

¹ Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, cristian.ungureanu@cti.pub.ro

² Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, andrei.tigora@cti.pub.ro

³ Associate Professor PhD Eng., "Politehnica" University of Bucharest, 060042 Bucharest, Romania, ion.bucur@cs.pub.ro

⁴ Engineer, "Politehnica" University of Bucharest, Bucharest, Romania, mihai.zaharescu@cti.pub.ro

of the performance of the algorithm as compared to the original. The last section is reserved for conclusions and future work.

2. BACKGROUND

Conversion from color to grayscale is a lossy image procedure, as there is usually no way of having a bijective mapping between three channel values and single channel ones (of course there is also the case of images with low number of colors but these are an exception). However, this is hardly an issue as most of the times the end user is aware of the tradeoff. Simple weighted averages are good enough for most conversions, although on some occasions, a need might arise to have more complex mappings in order to outline certain aspects of the image as described in the works of Cadik [8].

In some instances, it might be required to be able to recover the color image from a grayscale only medium. One solution would be to have each of the three (red, green and blue) channels displayed/represented individually using the unique color (black most frequently) and then combine the three images into a color representation. This is a valid idea, but it is hardly an efficient one, given that the technical equipment is already limited, it would be quite a waste to use three times more display support to reproduce something that will exhibit some obvious information loss. The solution of Queiroz and Braun [1] was to produce a single grayscale image that would have all the necessary information encoded within. And it is this solution that this paper attempts to improve.

3. THE BASIC ALGORITHM VERSION

The algorithm is based on the discrete wavelet transformation (DWT), being applied on Y level of luminosity of YCbCr color encoded images.

First, particular regions of the image are replaced with Cb and Cr versions of the scaled image. Then, by applying inverse discrete wavelet transformation, a grayscale image is obtained, which contains information about colors. This procedure is exemplified in Figure 1. So, even though the end user perceives a grayscale image, hidden within it, there is enough information to restore the color image by simply following the previously described procedure in reverse order, as shown in Figure 3.

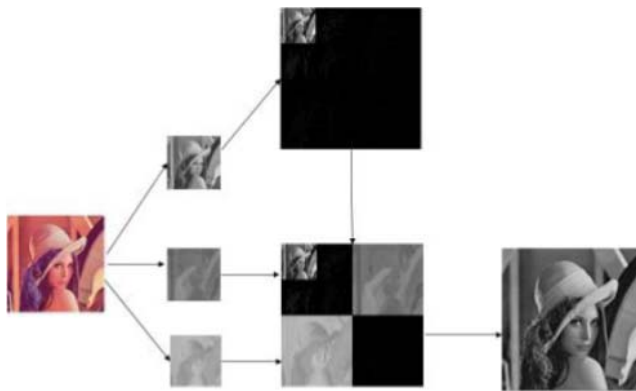


Figure 1. Color to grayscale steps applied on Lena image

Here is a detailed, step by step description of the algorithm:

1. Color image is transformed from RGB format to YCbCr (usually the case, as most images are RGB encoded)
2. Y level is transformed using discrete wavelet transform (DWT), producing an image divided into 4 sections.
3. The Cb and Cr components are scaled at 25% the initial image size (half the initial width, and half the initial height).
4. The bottom left quarter is replaced with Cr and top right quarter is replaced with Cb (see Figure 2).
5. The inverse discrete wavelet transformation (IDWT) is applied on the resulting image producing a grayscale image.

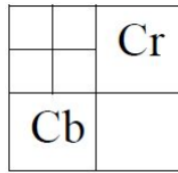


Figure 2. Image contents division

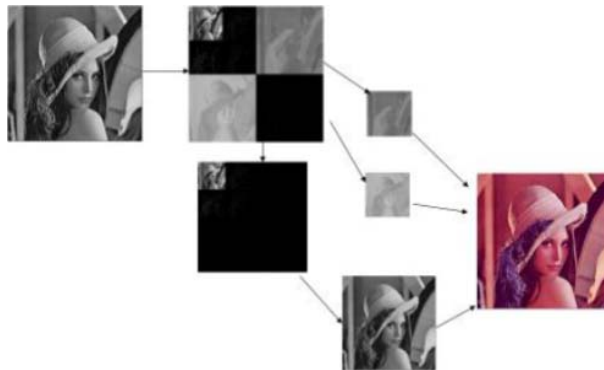


Figure 3. Grayscale to color steps applied on Lena image

Retrieving the color image is done as follows:

1. A DWT transformation is applied on the image, producing an image divided into 7 sectors (see Figure 4).
2. The bottom left quarter contains the scaled down Cr component, whereas top right quarter the Cb component; these can be re-dimensioned to the initial values using interpolation.
3. Pixels values in the two sections where Cr and Cb were stored are set to zeros (black tone in above image). On the resulting image an IDWT transformation is applied, producing a Y image.
4. By putting together Y, Cb and Cr levels will obtain YCbCr image, which can then be optionally be transformed into an RGB image.

		Cr=0
Cb=0		

Figure 4. Reverse image contents division

There are some drawbacks to using this algorithm:

- if the image is turned to grayscale, printed, and then scanned, it will most likely be impossible to recover the color image, due to various distortions that appear in the process: printer printing algorithm, scanner resolution, calibration problems, etc
- Given the size changes of the various components during algorithm execution and the approximations used for the interpolation mechanism the final image will be blurred;
- As a consequence of DWT transformation, there are “film” effects

4. IMPROVED VERSION OF THE ALGORITHM

Given the drawbacks of the previous approach, the authors decided to make some changes. The main change that was introduced was the separation of Cb and Cr levels into Cb+ and Cb- , Cr+ and Cr- , the X+ image containing all positive pixels of X image and 0 instead of negative pixels. The X- image contains all negative pixels of image X and 0 instead of positive pixels. The Cb+, Cb-, Cr+, Cr- , properly scaled, replace distinct sectors of resulting image by wavelet transformation on Y.

The steps for color to grayscale conversion are:

1. The RGB image is transformed into YCbCr, followed by an extraction of the Y, Cb and Cr components.
2. The DWT transformation is applied on the Y component, resulting 7 fragmented sub-images.
3. Cb and Cr are scaled at 25% of initial dimension and Cb+ and Cb- images are extracted (positive/negative values of pixels copied from Cb, the others equaled with 0), as well as Cr+ and Cr- (the same procedure).
4. The quarter from right bottom of left top sub-image is replaced with Cb- , the right-top sub-image with Cr+, the sub-image from left-bottom with Cb+ and sub-image from right-bottom with Cr- (see Figure 5).
5. IDWT transformation is applied to the resulting image, obtaining the desired grayscale image.

		Cr+
	Cb-	
Cb+		Cr-

Figure 5. Component distribution in the image for the improved algorithm

The reverse procedure, recovering the color image, is detailed below:

1. Read the grayscale image obtained above.
2. Determine size of the image.
3. If it is necessary, identify the corners and apply a transformation to cancel the warping effect.
4. Reduce the image to the right resolution.
5. Apply a DWT transformation on the image, obtaining an image with 7 components, amongst which the Cb-, Cb+, Cr- and Cr+, components are identified (see Figure 6).
6. Using interpolation, Cb- is scaled back to the original size.
7. The Cb and Cr components are computed as: $Cb = |Cb+| - |Cb-|$ and $Cr = |Cr+| - |Cr-|$
8. Through interpolation, Cb and Cr are scaled back to the original sizes.
9. The Cb-, Cb+, Cr- and Cr+ pixels are set to 0 (black), then, the IDWT transformation is applied to the image in order to obtain the Y component.
10. The three components are put together to form the YCbCr image, from which the RGB image is obtained.

		Cr+=0
	Cb-=0	
Cb+=0		Cr-=0

Figure 6. Reverse image contents division for improved algorithm

The second algorithm fixes some of the problem of the first one, but introduces others, such as an unclear image due to larger surface replacement from the DWT image.

5. PROPOSED ALGORITHM

The proposed algorithm is based on the from the second version, but changes the sub-images which are replaced, which leads to a better recovering of color image information.

Other changes brought by the latest version are:

- For a better delimitation of shades, the value conversion from $[0, \text{max value}]$ range to $[0, 255]$ range
- Linear adjustments of values from image in order to enclose the colors values frame.
- Enlargement and reduction of grayscale image obtained after conversion in order to simulate image printing/scanning and loses of possible information.
- Blur application on enlarged image, before reduction, in order to prevent possible errors regarding calibration, alignment, etc.

The steps for color to grayscale conversion algorithm are:

1. Transform the RGB image into YCbCr, and extract the three components: Y, Cb and Cr.
2. Apply the DWT transformation on Y image, creating an image fragmented in 7 sub-images (see Figure 7).
3. Scale Cb and Cr at 25% of the original size and extract Cb+ and Cb- images (positive/negative pixels copied from Cb, the others fixed to 0), then do the same for Cr+ and Cr-.
4. In the left-top sub-image, replace the quarter from right-top with Cr+, the quarter from right-bottom with Cb- and the one from left-bottom with Cb+. Then, replace the right-bottom sub-image with Cr-.
5. Apply the IDWT transformation to the resulting image to obtain the grayscale image.

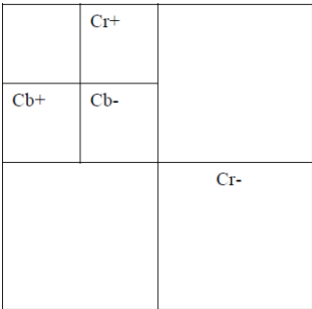


Figure 7. Component distribution in the image for the proposed algorithm

The grayscale to color reversion algorithm steps are:

1. Read the grayscale image previously obtained.
2. Determine the image size.
3. If necessary, identify the corners should be identified and have a transformation applied in order to cancel the warping effect.
4. The image is reduced to the right resolution.

5. A DWT transformation is applied on the image, obtain a 7-component image, in which $Cb-$, $Cb+$, $Cr-$ and $Cr+$ are identified as in Figure 8 presented below.
6. Using interpolation, $Cb-$, $Cb+$ and $Cr+$ are scaled to 4 times their current size.
7. The Cb and Cr components are computed as follows: $Cb = |Cb+| - |Cb-|$ and $Cr = |Cr+| - |Cr-|$.
8. Using interpolation, Cb and Cr are scaled to image size.
9. All pixels belonging to $Cb-$, $Cb+$, $Cr-$ and $Cr+$ are set to 0 (black), then an IDWT transformation is applied on the image to obtain Y component.
10. As all three components (Y , Cb and Cr) are known, the RGB image can be obtained.

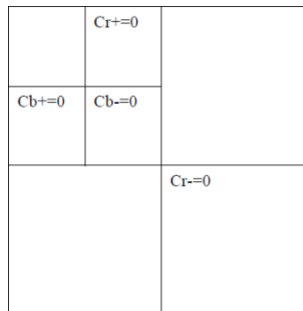


Figure 8. Reverse image contents division for the proposed algorithm

6. RESULTS



Figure 9: The grayscale image using the second and the proposed version of algorithm

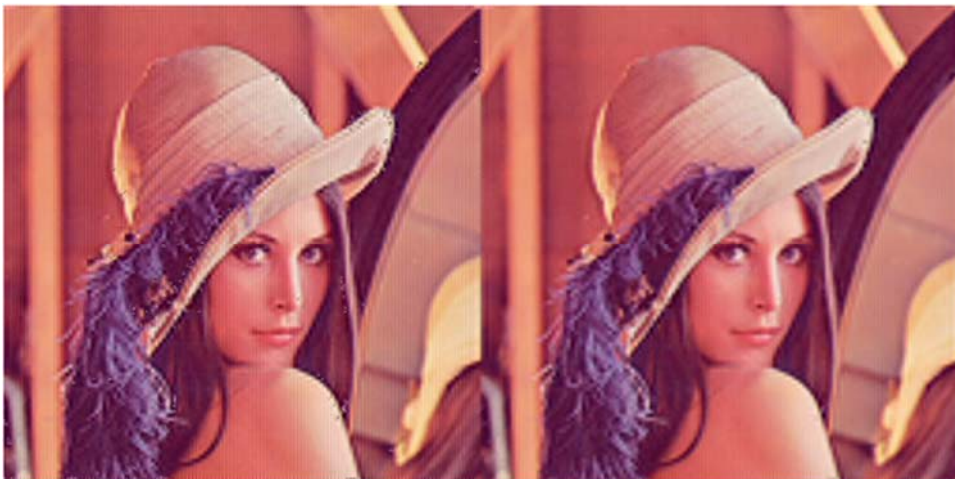


Figure 10: Recolored image using the second and the proposed version of algorithm

The algorithms were implemented in C++, using OpenCV [2] and functions library for Discrete Wavelet Transform [3] on an Intel Core2Duo machine running Windows7.

As can be seen in Figure 9, the proposed algorithm produces a better looking image than the second version of the original paper. There is a distinct vertical grid aspect identifiable in the first picture that is not so strongly visible in the second one. In relation to the recovered color versions, the proposed algorithm appears to have a certain blurring effect, as indicated in the second picture of Figure 10. However, this eliminates some of the less pleasing visual artifacts that are clearly identifiable in the original algorithm in around the surface borders (different colored dots that appear out of nowhere).

7. CONCLUSIONS

The proposed change to the second version of the algorithm, adds a significant improvement to the original. Through this minor modification, a smaller surface from the DWT transformed image is replaced, leading to less irreversible losses from the transformation. Also, the recovered image is more accurate, as there are less incorrect pixels. Even if the final image is less clear than the one obtained with the first algorithm, the latest version of it is usable in real situations (printer, scanner, fax, etc.)

As a conclusion, the proposed algorithm offers the color to grayscale image conversion possibility in a reversible way, obtaining final images almost identical to the original ones at detail level.

8. REFERENCES

- [1] Ricardo L. de Queiroz, Karen M. Braun, "Color to Gray and Back: Color Embedding Into Textured Gray Images", IEEE Transactions on Image Processing, 15(6), pp.1464–1470, 2006
- [2] The OpenCV Library (<http://opencv.org>)
- [3] Discrete Wavelet Transform - (<http://code.google.com/p/wavelet1d/>)

- [4] Andreea-Mihaela Pintilie, Costin-Anton Boiangiu, "Study Of Neurobiological Images Based On Ontologies Used In Super-Resolution Algorithms", The Proceedings of Journal ISOM Vol. 7 No. 2, pp. 300-308, December 2013
- [5] Adrian Enache, Costin-Anton Boiangiu, "Adaptive video streaming using residue hypercubes," in Proceedings of the 12th International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS '13), Budapest, Hungary, December 10-12, 2013, pp. 173-179
- [6] Costin-Anton Boiangiu, Alexandru Victor Ștefănescu, "Target validation and image calibration in scanning systems", in Proceedings of the 1st International Conference on Image Processing and Pattern Recognition (IPPR '13), Budapest, Hungary, December 10-12, 2013, pp. 72-78
- [7] Costin-Anton Boiangiu, Alexandru Victor Stefanescu, Daniel Rosner, Alexandra Olteanu, Anca Morar (2010). „Automatic Slanted Edge Target Validation in Large Scale Digitization Projects” (2010), Annals of DAAAM for 2010, Proceedings of the 21st International DAAAM Symposium, 20-23 October 2010, Zadar, Croatia, pp. 131-132, 2010.
- [8] M. Čadík, Perceptual Evaluation of Color-to-Grayscale Image Conversions, Computer Graphics Forum, Vol. 27, 2007, pp. 1745-1754