MVA2011 IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN

Image colorization using discriminative textural features

Michal Kawulok, Jolanta Kawulok and Bogdan Smolka *
Faculty of Automatic Control, Electronics and Computer Science
Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
{mkawulok,jkawulok,bsmolka}@polsl.pl

Abstract

This paper presents a novel approach to scribble-based image colorization. In the work reported here we have explored how to exploit the textural information to improve this process. For every scribbled image we extract the most discriminative features using linear discriminant analysis (LDA). After that, the whole image is projected onto a discriminative textural feature space. Our main contribution lies in propagating the color in the feature space domain rather than using the luminance channel. The presented experimental validation confirms the importance of using textural information and show that our method significantly improves the obtained colorization results.

1 Introduction

Image colorization is a process of adding colors to a grayscale image. This is a sophisticated task which requires high-level arbitrary knowledge concerning the image content. Such information cannot be delivered by recent image understanding systems yet and therefore this process is currently intended to be human-assisted. Existing solutions make it possible to limit the necessary human actions to defining the initial color scribbles or giving examples of color images of similar content. Among many others, the image colorization applications include: enhancing visual attractiveness of monochrome photographs or videos whose color versions are not available, marking regions of interest in medical images, interior design or make-up simulators.

Image colorization attracts considerable attention from the academia world. The first method [1] was a luminance keying based on a function which maps every luminance level into color space. Obviously, the whole color space cannot be covered in this way without increasing manual input from the user. Color transfer method [12] colorizes a grayscale image based on a given reference color image. This method matches textural and luminance information and can be performed automatically, but gives better results with user assistance. Unsupervised image colorization by example [11] matches at first similar image feature points to predict their color. After that, the color is spread all over the image using probabilistic relaxation. There are also a number of methods which are focused on using the prior information delivered by a user in a form of color scribbles. Levin et al. [7] formulated an optimization problem based on an assumption that neighboring pixels of similar intensity should have similar color values under the limitation that the colors indicated in the scribbles remain the same. Yatziv and Sapiro [13] proposed a method for determining propagation paths in the image by minimizing geodesic distances from every scribble. Based on the distances from each scribble, pixel color is obtained by blending scribble chrominances. In other works, the color is propagated with probabilistic distance transform [6], using cellular automaton [5] or by random walks [4].

In the work reported here we have investigated how to exploit the textural information to improve the colorization result. Although the existing techniques work well for colorizing plain areas, they fail for rough, textured regions. Therefore, we transform the image pixels into a discriminant textural feature space, in which the color propagation is performed. This is the main contribution of our work, while in alternative approaches the costs are obtained based on the difference in pixel luminance [13]. The propagation itself, as well as the chrominance blending, is executed following the conventional techniques presented in Section 2. Our method for computing the textural features and color propagation using the obtained feature space is described in Section 3. The obtained colorization results are shown and discussed in Section 4, and the conclusions are presented in Section 5.

2 Color propagation and blending

In order to colorize a monochromatic image Y based on a set of n initial scribbles $\{S_i\},\ i = 1, ..., n$, first it is necessary to determine the propagation paths from each scribble to every pixel in the image. A path from a pixel x to another pixel y is defined as a discrete function $p(t) : [0, l] \to Z^2$, which maps a position t in the path to the pixel coordinate. The position is an integer ranging from 0 for the path beginning ($p(0) = x$) to l for its end ($p(l) = y$). Also, if $p(i) = a$ and $p(i + 1) = b$, then a and b are neighboring pixels.

The propagation paths from a scribble to every pixel are determined by minimizing a *total path cost*:

$$C(p) = \sum_{i=0}^{l-1} \rho \left\{ p(i), p(i+1) \right\}, \tag{1}$$

where ρ is a *local cost* between two neighboring pixels and l is the path length. An image is considered as a graph and the cost minimization is performed using Dijkstra algorithm [3]. The path route depends mainly on how the local costs are computed. Following Yatziv's approach [13], the local cost is obtained by projecting the luminance gradient onto a line tangent to the path direction. This means that the cost is proportional to the difference in luminance between the neighboring pixels.

*This work has been supported by the Polish Ministry of Science and Higher Education under R&D grant no. N N516 374736 from the Science Budget 2009-2011.

Chrominance of each pixel is determined based on the propagation paths from every scribble. Its value is computed as a weighted mean of scribbles' colors with the weights obtained as a function of the total path cost. The chrominance is calculated as a weighted mean of scribbles' colors with the weights obtained as a function of the total path cost. Usually two or three strongest components are taken into account which provides a good visual effect of smooth color transitions. The final color value $v(x)$ of a pixel x is obtained as $v(x) = \sum_i v_i w_i(x) / \sum_i w_i(x)$, where v_i is the chrominance of i-th scribble and $w_i(x)$ is its weight in pixel x. We use YC_rC_b color space and calculate color values separately for C_r and C_b channels. The weights are obtained as

$$w_i(x) = (C_i(x) + 1)^{-2}, \tag{2}$$

where $C_i(x)$ is the total path cost from i-th scribble to pixel x.

3 Textural features for image colorization

Regions of uniform texture quite often have similar chrominance. Following this fact, the texture may be an important source of information in image colorization. Unfortunately, this is neglected by many existing techniques, which assume that the chrominance boundaries are correlated with the brightness changes rather than with the texture. Following this assumption, the raw pixel values in luminance channel are used as the color propagation domain [2,13]. Although color transfer methods [8,12] exploit simple textural descriptors, they do not provide the distinctiveness helpful in improving the propagation. In our work we focus on finding a better color propagation domain which takes into account the textural features rather than the luminance exclusively.

Texture analysis is a complex task itself and many works have been reported, mainly in the aspect of texture-based image segmentation [9, 14]. The main difficulty lies in a lack of precise definition of the correct texture segmentation result – usually the expected outcome depends on a specific application. The considered case is not identical to the segmentation task. Here, the aim is to define a suitable domain for color propagation, and we have found textural features very helpful for this purpose. Among the existing colorization methods, textural features have been exploited for color transfer [8,12]. However, only simple texture descriptors are used there, which may be helpful in some cases, but does not guarantee the distinctiveness between the regions marked with different scribbles.

3.1 Discriminative textural features

The color propagation domain should induce low costs between pixels belonging to a single scribble. On the other hand, the cost should be high, when the path crosses a boundary between areas marked with different scribbles. It is therefore important to find such image properties that would be uniform within a single scribble and different between the scribbles. It is worth noting that the most relevant properties may be different from case to case. For some images the luminance itself may be sufficiently distinctive (e.g. in case of cartoons), while for others the variation in the gradient intensity may be relevant. In the work reported here we select the distinctive properties independently for every scribbled image using linear discriminant analysis (LDA). The analysis is performed over a set of simple image features extracted from pixels which belong to the scribbles. In this way we obtain the color propagation domain which is dynamically conformed to every specific case.

3.1.1 Linear discriminant analysis

Linear discriminant analysis [10] is a supervised statistical feature extraction method frequently used in machine learning. It finds a subspace defined by the most discriminative directions within a given training set of M-dimensional vectors classified into K classes. The analysis is performed first by computing two covariance matrices: within-class scatter matrix $S_W = \sum_{i=1}^{K} \sum_{\boldsymbol{u}_k \in K_i} (\boldsymbol{u}_k - \boldsymbol{\mu}_i)(\boldsymbol{u}_k - \boldsymbol{\mu}_i)^T$ and between-class scatter matrix $S_B = \sum_{i=1}^{K} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$, where $\boldsymbol{\mu}$ is a mean vector of the training set and $\boldsymbol{\mu}_i$ is a mean vector of the i-th class (termed K_i). Subsequently, the matrix $S = S_W^{-1} S_B$ is subjected to the eigen decomposition $S = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_M)$ is the matrix with the ordered eigenvalues along the diagonal and $\Phi = [\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_M]$ is the matrix with the correspondingly ordered eigenvectors as columns. The eigenvectors form the orthogonal basis of the feature space. Originally, the feature space has M dimensions, but only those associated with the highest eigenvalues have strong discriminative power, while the remaining can be rejected. In this way the dimensionality is reduced from M to m, where $m < M$.

After having built the m-dimensional feature space the feature vectors are obtained by projecting the original vectors $\boldsymbol{u}$ onto the feature space: $\boldsymbol{\nu} = \Phi^T \boldsymbol{u}$. The similarity between the feature vectors is computed based on their Euclidean distance in the feature space.

3.1.2 LDA for texture analysis

In order to determine the discriminative features, first we calculate *basic image features* from every pixel. They are composed of: a) luminance, b) gradient intensity, c) local binary pattern, d) mean value and e) standard deviation computed in many kernels of different size, f) the difference between maximum and minimum values in the kernels, and g) the pixel value in the median filtered image. The basic features (d) – (g) were obtained for 15 kernel sizes ranging from 3×3 to 31×31. Hence, every pixel x is described by an M-dimensional basic feature vector $\boldsymbol{u}_x$ ($M = 63$ in the presented case). The feature vectors of the scribble pixels are subsequently subject to LDA. Every scribble forms a separate class, so the analysis determines the most discriminative features between the scribbles for a given image. The feature vectors ($\boldsymbol{\nu}$) obtained using LDA are further termed *discriminative textural features* (DTF). The distance between any two feature vectors $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ in DTF space is computed as:

$$d = \sum_{i=1}^{m} (\nu_{1i} - \nu_{2i})^2. \tag{3}$$

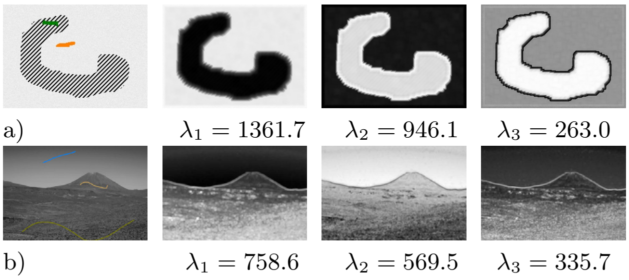

Figure 1. Scribbled images and their projections onto three leading LDA components.

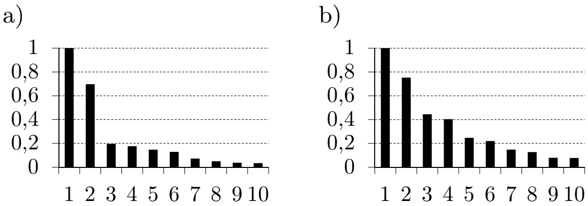

Figure 2. Eigenvalues of subsequent LDA components (relatively to the highest eigenvalue).

During our experiments we observed that for the majority of analyzed cases it is sufficient to reduce the dimensionality of DTF vectors to $m = 2$. Also, we limit the number of the input vectors in each class to 100 so as to reduce the LDA training time. If a scribble contains more pixels, 100 of them are randomly selected. We have not observed any noticeable difference in the outcome compared to using all the scribble pixels, while the training time is definitely shorter.

3.2 DTF-based color propagation domain

After training, a projection matrix Φ is obtained and every pixel in the image is projected onto m-dimensional DTF space. Examples of two scribbled images and their projection onto three leading LDA components representing the most discriminative textural features are presented in Fig. 1. The eigenvalues associated with these components are also given in the figure. It may be observed that these projections differentiate well between the areas marked with the scribbles. Also, 20 highest eigenvalues obtained for these two images are plotted in Fig. 2.

Fig. 3 shows an image with two scribbles (a) marked over forest and sky. The luminance of these pixels scaled from 0 to 100 is presented in (b) on the horizontal axis. It may be noted that although the forest pixels (F) are generally darker than the sky pixels (S), the luminance alone is not a discriminative feature here. The pixels projected onto 2D DTF subspace are shown in (c). Here, two classes are well separated.

For every scribble a mean DTF feature vector is obtained and its DTF-distance (3) to every pixel in the image is computed in DTF space. In this way a DTF-distance map $\boldsymbol{d}_i$ is obtained for every i-th scribble, which serves as a domain for determining the propagation paths. The local cost ρ from pixel x to y equals the y pixel value in the DTF distance map ($\rho(x, y) = \boldsymbol{d}_i(y)$). Examples of DTF-distance maps obtained for the landscape image in Fig. 1b are presented

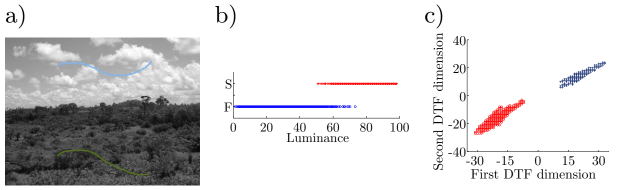

Figure 3. Scribble pixels (a) projected onto luminance (b) and 2D LDA (c) subspaces.

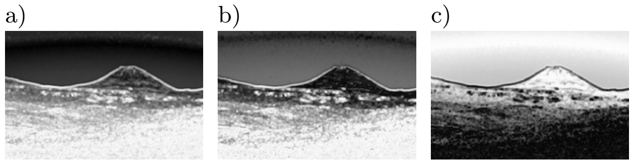

Figure 4. Examples of DTF-distance maps.

in Fig. 4. These are the distances from the scribbles marked over: (a) sky, (b) volcano, and (c) ground.

It is worth noting that potentially the distance maps could be used directly for chrominance blending without determining the propagation paths. In this case, to obtain an i-th weight for a pixel x, the distance in DTF space $\boldsymbol{d}_i(x)$ would be used instead of the total path cost $C_i(x)$ in (2). Such approach does not guarantee continuity of the regions and therefore has not been utilized in the investigated cases. However, it may be useful for some other applications, e.g. for color transfer or video colorization.

The propagation paths are determined so that they follow the texture similar to that covered by the source scribble as long as possible. This is contrary to the conventional Yatziv's approach [13], with which the path is determined to minimize the gradient integrated in the propagation direction. An example of a difference between these two alternative approaches is given in Fig. 5. It shows the propagation paths leading from a scribble to a selected pixel obtained using two methods (the original scribbled image is shown in Fig. 1a). The path determined using our method (a) does not leave the striped area, which allows to colorize the image correctly. The result obtained using a conventional method (b) shows that the texture information is not taken into account during the propagation.

In some cases the drawback of the presented method lies in the precision. At the boundaries of regions having different texture the pixels may be misclassified (it can be seen in Fig. 5a). Size of such uncertain area depends on the kernel dimensions used for obtaining the basic textural features. This results in observing small halos at the region boundaries. We have approached this problem by determining a map of equal

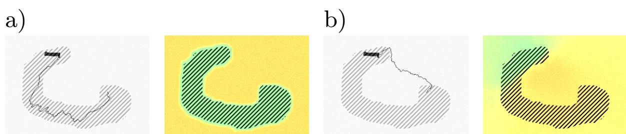

Figure 5. Propagation paths and colorized image obtained using our (a) and Yatziv's approach (b).

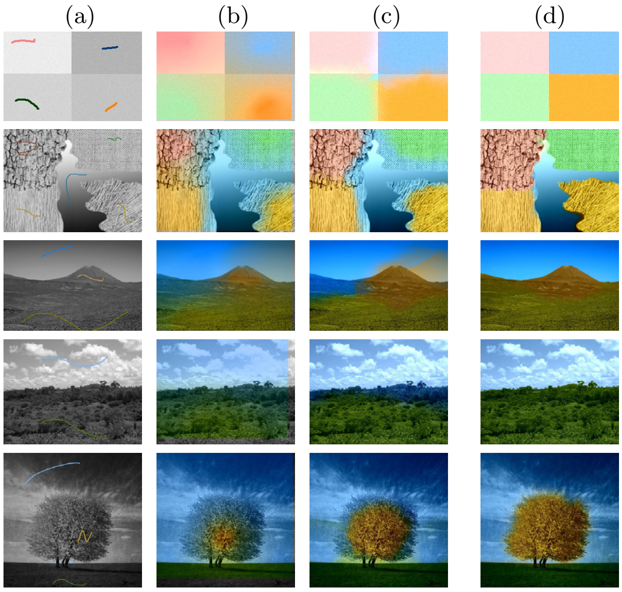

Figure 6. Input image with scribbles (a) and colorization results obtained with [7] (b), [13] (c) and our algorithm (d).

DTF distances. Then, in proximity of the equally distant pixels, the propagation domain is changed to the conventional one. This allows to reduce the aforementioned effect in most cases.

4 Experimental validation

We compared the proposed method with two well-established colorization techniques proposed by Levin [7] and Yatziv [13]. The first one is published in the form of MATLAB code and for the latter a Java applet is available to colorize a fixed set of images (for others we used our implementation). We evaluated the colorization on the basis of the visual result as this is a commonly adapted practice in image colorization.

Several examples of colorization result achieved using our method and alternative algorithms are given in Fig. 6. We validated our algorithm using three types of images: 1) artificial to verify the theoretical assumptions (1^{st} row from the top), 2) semi-artificial composed of various textures (2^{nd} row), and 3) photographs to assess applicability of our approach. Their sizes range from 280×180 to 768×704 pixels. It may be seen that our algorithm delivers the best visual effect in all of the presented cases, making it possible to colorize the textures with the largest precision. The halo effect mentioned in Section 3.2 can be observed around the tree for the image in the bottom row. Also, small imprecisions can be noted for the images in the top two rows, but they are definitely smaller than for the alternative methods. The photographs in 3^{rd} and 4^{th} row have been colorized perfectly using our method.

The time needed to complete the colorization process has not exceeded 20s for the tested cases. We have limited the number of pixels in every class to 100 and we use only two LDA dimensions. This assures that calculating the DTF space is not computationally expensive and takes a few percent of the time needed to colorize the image using conventional methods.

5 Conclusions and future work

In this paper we have presented a novel method for scribble-based image colorization, which uses the discriminant textural features domain for color propagation instead of the luminance channel. The experiments have shown that the proposed solution greatly improves the obtained results and facilitates human assistance in the colorization process.

Although the DTF domain works well for color propagation, its precision is limited at the region boundaries due to a large kernel size. This was discussed in Section 3.2 and presented in Figs. 5a and 6. Sometimes it results in unnatural halo effects which can be avoided in most cases using a simple technique. However, in future we are planning to develop a better solution to this problem. Moreover, we want to apply the proposed technique to color transfer and video colorization.

References

[1] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley Publishing, 1987.

[2] J.H. Heu, D.Y. Hyun, C.S. Kim, and S.U. Lee. Image and video colorization based on prioritized source propagation. In *IEEE ICIP*, pages 465–468, 2009.

[3] L. Ikonen and P.J. Toivanen. Distance and nearest neighbor transforms on gray-level surfaces. *Pattern Rec Lett*, 28(5):604–612, 2007.

[4] T.H. Kim, K.M. Lee, and S.U. Lee. Edge-preserving colorization using data-driven random walks with restart. In *IEEE ICIP*, pages 1661–1664, 2009.

[5] V. Konushin and V. Vezhnevets. Interactive image colorization and recoloring based on coupled map lattices. In *GraphiCon*, pages 231–234, 2006.

[6] P. Lagodzinski and B. Smolka. Digital image colorization based on probabilistic distance transform. In *ELMAR, 2008*, volume 2, pages 495–498, 2008.

[7] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *SIGGRAPH*, pages 689–694. ACM, 2004.

[8] U. Lipowezky. Grayscale aerial and space image colorization using texture classification. *Pattern Rec Letters*, 27(4):275–286, 2006.

[9] T. Ojala, M. Pietikinen, and T. Menp. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell*, 24(7):971–987, 2002.

[10] G.A.F. Seber. *Multivariate Observations*. Wiley, New York, 1984.

[11] D. Sykora, J. Burianek, and J. Zara. Unsupervised colorization of black-and-white cartoons. In *SIGGRAPH*, pages 121–127. ACM, 2004.

[12] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Trans Graph (TOG)*, 21(3):277–280, 2002.

[13] L. Yatziv and G. Sapiro. Fast image video colorization using chrominance blending. *IEEE Trans on Image Proc*, 15(5):1120–1129, 2006.

[14] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *Int J of Computer Vision*, 73:213–238, 2007.