Discriminative Textural Features for Image and Video Colorization

Michal KAWULOK, Nonmember, Jolanta KAWULOK, Student Member, and Bogdan SMOLKA, Nonmember

SUMMARY Image colorization is a semi-automatic process of adding colors to monochrome images and videos. Using existing methods, required human assistance can be limited to annotating the image with color scribbles or selecting a reference image, from which the colors are transferred to a source image or video sequence. In the work reported here we have explored how to exploit the textural information to improve this process. For every scribbled image we determine the discriminative textural feature domain. After that, the whole image is projected onto the feature space, which makes it possible to estimate textural similarity between any two pixels. For single image colorization based on a set of color scribbles, our contribution lies in using the proposed feature space domain rather than the luminance channel. In case of color transfer used for colorization of video sequences, the feature space is generated based on a reference image, and textural similarity is used to match the pixels between the reference and source images. We have conducted extensive experimental validation which confirmed the importance of using textural information and demonstrated that our method significantly improves colorization result.

key words: image colorization, video colorization, color transfer, textural features

1. Introduction

Development of monochrome photographs and films was an important step of our civilization towards documenting visual information. However, as soon as color image acquisition became feasible, it appeared desirable to create color versions of existing monochrome material. The process of adding colors to grayscale images and videos is termed colorization. First of all, image colorization is applied to enhance visual attractiveness of monochrome photographs or videos whose color versions are not available, but it can also be used for such applications as marking regions of interest in medical images, image matting, or interior design and make-up simulators.

At the beginnings, for example in the case of colorizing Apollo mission footage in 1960s, the whole process was fully manual and therefore required huge amount of work.

Along with the development of digital imaging, image colorization is being automated in order to simplify this tedious task. Image colorization attracts considerable attention from computer vision society and new methods aimed at reducing required human interaction are still being developed.

Nowadays, image colorization is still considered a sophisticated problem 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. For scribble-based routines the colors are propagated from the marked image regions using distance transform in the luminance domain. This works well for colorizing plain areas, but usually fails for rough, textured regions. Color transfer methods match the most similar pixels between two given images, and the chrominance is transferred from the reference image. The matching is performed based on pixel luminance or statistics computed in the pixel’s neighborhood.

In the work reported here we present a new method which exploits the textural information to improve the colorization result. First, based on a scribbled image, we obtain a discriminative textural feature space, adapted to every individual case. Then, the whole image is transformed into the feature space to obtain a suitable propagation domain. The most relevant points in our contribution are:

1. Color propagation from the scribbles is performed using a discriminative domain which includes both textural and luminance features.
2. The propagation domain is dynamically adapted to every individual image by determining those features which provide best discrimination between the scribbled regions.

Furthermore, we also used the textural feature space to match the pixels between two different images and transfer colors between them. We have successfully used this framework to colorize video sequences.

The paper is organized as follows. Overview of existing techniques with particular attention given to texture-based methods is outlined in Sect. 2. Chrominance blending and color propagation is explained in Sect. 3. Our contribution is presented in Sect. 4: procedure for extracting...
discriminative textural features is outlined in Sect. 4.1, proposed method for computing the textural features and color propagation using the obtained feature space is described in Sect. 4.2, and color transfer using the textural feature space is presented in Sect. 4.3. The obtained colorization results are shown and discussed in Sect. 5. All the figures presented in this paper can be downloaded in full quality from http://sun.aei.polsl.pl/~mkawulok/df_tfig.

2. Related Work

The first method for automating image colorization was the luminance keying [12], proposed by Gonzalez and Woods in 1987. It is 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.

Unsupervised image colorization by example [13] matches at first similar image feature points to determine their color. After that, the color is spread all over the image using probabilistic relaxation. Furthermore, Horiuchi proposed a method for texture colorization [14] which defines pixel similarity based on their Euclidean distance and difference in luminance values. Hence, even if two neighboring pixels differ much in luminance, which is often observed for textured regions, their similarity will be high due to low Euclidean distance. This approach works better for colorizing textures than the earlier methods, but it does not perform any extraction of textural features.

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. formulated an optimization problem [1] 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 proposed a method [2] 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 [3], using cellular automaton [4] or by random walks [5]. In our earlier works we proposed to use two competitive propagation domains suitable for plain and rough regions [6].

Color transfer makes it possible to add colors to a monochrome image or to modify image colors based on a given reference image. The first color transfer algorithm [10] was proposed by Reinhard et al. and it was later improved [7] by Welsh et al. The latter method matches simple textural and luminance information and can be performed automatically, but gives better results with user assistance. First, a set of 200 pixels is randomly selected from the reference image. Similarity between every pixel in the source image and each reference pixel is obtained as an average difference in their luminance and standard deviation in $5 \times 5$ neighborhood. Finally, chrominance is transferred from the best matching reference pixel. This global image matching procedure can be improved by defining regions of interest (originally called swatches) in the reference and source images. Corresponding swatches are supposed to have similar contents, and initially the colors are only transferred within each pair of swatches. After that, the color is transferred from the colorized swatches to the remaining region of the source image following the global matching procedure. It is worth noting that the swatches must be annotated both in the source and reference image which makes this feature practically unapplicable for colorizing video sequences.

Color transfer methods were further investigated and improved. Lipowezky used texture prototype matching [8] to improve Welsh’s framework. Xiang et al. investigated a possibility of using multiple reference images [9] for color transfer. Yao et al. proposed to detect and match feature points in two images based on local binary patterns [11]. However, these methods were primarily focused on improving the matching rules rather than extracting textural features.

2.1 Texture Analysis

Regions of uniform texture quite often have similar chrominance and therefore advanced textural analysis proved to be beneficial for image colorization. Irony et al. proposed a color transfer method [15] based on textural features extracted using discrete cosine transform (DCT). Here, the image is transformed into a 49-dimensional DCT feature space, which is further reduced to 10 dimensions using linear discriminant analysis (LDA). The textural features are only used to determine location of transferred micro-scribbles which are subsequently propagated using Levin’s method [1].

Qu et al. exploited textural features for colorizing manga images [16] from user-added scribbles. Here, the features are extracted using Gabor wavelets and the propagation is performed using level sets. The proposed textural feature space is highly effective for the patterns typically used in manga images, but it is not suitable for colorizing non-textured regions. An interesting approach proposed by Luan et al. is to take into account both intensity continuity and texture similarity [17]. These two components form an energy map which is optimized to generate the colorized image.

There are also works which use textural features to automate the image colorization procedure. Charpiat et al. investigated how to determine color from texture based on machine learning schemes [18]. Liu et al. proposed a color transfer framework [19] which automatically retrieves reference images from the Internet. This method was extended by Chia et al. [20], who exploited textural features extracted using Gabor filters for image selection and color transfer.

3. Color Propagation and Blending

In order to colorize a monochromatic image $Y$ based on a set
of \(n\) initial scribbles \(s_i\), \(i = 1, \ldots, n\), 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 \(p(i) : [0, l] \rightarrow \mathbb{Z}^2\), which maps a position \(i\) 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:

\[
C(p) = \sum_{i=0}^{l-1} \rho \{ p(i), p(i+1) \} ,
\]

where \(\rho\) 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 [21]. The path route depends mainly on how the local costs are computed. Following Yatziv’s approach [2], 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.

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

\[
v(x) = \frac{\sum_i v_i w_i(x)}{\sum_i w_i(x)} ,
\]

where \(v_i\) is the chrominance of \(i\)-th scribble and \(w_i(x)\) is its weight in pixel \(x\). We use \(YC_aC_b\) color space and calculate color values separately for \(C_a\) and \(C_b\) channels. The weights are obtained as

\[
w_i(x) = (C_i(x) + 1)^{-2} ,
\]

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

4. Proposed Texture-Based Image Colorization

Textural features are an important source of information exploited by many image colorization methods outlined in Sect. 2.1. However, to fully benefit from the textural features, it must be provided that they discriminate well between the regions of different colors. This can be hardly achieved using a fixed feature space, such as in the Qu’s method for manga colorization [16]. An adaptive approach presented in the Irony’s color transfer [15] deals with this problem by using LDA, but as this method is based on local frequency analysis, it is required that the regions differ in the frequency domain. The obtained feature space is used only to match the most similar pixels, while the propagation is performed in the luminance domain using the Irony’s method [1].

In the work reported here we also use LDA to find a discriminative feature space, but contrary to the Irony’s method we take into account local image statistics obtained in different kernels. This makes our method applicable both for the images with distinctive textures, as well as for the images without prominent presence of textures, in which the chrominance is best correlated with the luminance. The obtained feature space provides suitable propagation domain adapted to an individual image, and it can also be used for color transfer and video colorization.

4.1 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 features may vary from case to case. For some images the luminance itself may be sufficiently distinctive, while for others the variation in the gradient intensity may be relevant. We select the distinctive properties independently for every scribbled image using 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 textural feature space which is dynamically conformed to every specific case.

4.1.1 Linear Discriminant Analysis

Linear discriminant analysis [22] 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 \(S_w = \sum_{i=1}^K \sum_{u \in K} (u - \mu)(u - \mu)^T\) and between-class scatter matrix \(S_B = \sum_{i=1}^K (\mu_i - \mu)(\mu_i - \mu)^T\), where \(\mu\) is a mean vector of the training set and \(\mu_i\) 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 \(S = PL\Phi L^T\), where \(L = \text{diag}(\Lambda_1, \ldots, \Lambda_M)\) is the matrix with the ordered eigenvalues along the matrix diagonal and \(\Phi = [v_1] \ldots [v_M]\) is the matrix with the corresponding 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 \( \mathbf{u} \) onto the feature space: \( \mathbf{v} = \Phi^T \mathbf{u} \). The similarity between the feature vectors is computed based on their Euclidean distance in the feature space.

### 4.1.2 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 5 kernel sizes ranging from \( 3 \times 3 \) to \( 11 \times 11 \). Hence, every pixel \( x \) is described by an \( M \)-dimensional basic feature vector \( \mathbf{u}_x \) (\( M = 23 \) 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 \( \mathbf{v} \) obtained using LDA are further termed discriminative textural features (DTF). The distance between any two feature vectors \( \mathbf{v}_1 \) and \( \mathbf{v}_2 \) in DTF space is computed as:

\[
d = \sum_{i=1}^{m} (v_{1i} - v_{2i})^2.
\]

Before training 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.

### 4.2 DTF-Based Color Propagation Domain for Single Image Colorization

After training, a projection matrix \( \Phi \) is obtained and every pixel in the image is projected onto \( m \)-dimensional DTF space. Examples of two scribbled images and their projections onto three leading LDA components, representing the most discriminative textural features, are presented in Fig. 1 (a)–(c). 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, 10 highest eigenvalues obtained for these images are plotted in Fig. 1 (d) (their values are given in relation to the sum of all eigenvalues). It may be observed from these charts that the discriminating power is cumulated in the first two leading components. Our experiments confirmed this observation and hence we reduce the dimensionality of DTF vectors to \( m = 2 \). Using more dimensions does not improve the result, while making the computations more time-consuming. It is worth to note that the Irony’s method [15] which also uses LDA requires 10-dimensional feature space.

![Fig. 1](landscape.png) ![Fig. 1](cheetah.png)

**Fig. 1** Scribbled images, their projections onto three leading LDA components (a)–(c) and 10 highest eigenvalues (d).

![Fig. 2](forest.png) ![Fig. 2](tree.png)

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

Figure 2 shows images with annotated scribbles (a). The luminance of these pixels scaled from 0 to 100 is presented in (b) on the horizontal axis. For the Forest image (top row) it may be noticed that although the forest pixels (F) are generally darker than the sky pixels (S), the luminance alone is not a discriminative feature here. In case of the Tree image (bottom row) the scribbles cannot be differentiated by the luminance at all — sky (S) and grass (G) pixels have very similar values, while tree (T) pixels represent a large variance in the luminance domain. The pixels projected onto 2D DTF subspace are shown in (c). Here, the classes are better separated for both images. Although some overlapping can be noticed in case of the tree image for (G) and (S) scribbles, the discrimination is sufficient to obtain a satisfactory colorization result, which is presented later in Sect. 5.

For every scribble, a mean DTF feature vector is obtained and its DTF-distance (4) to every pixel in the image is computed in the DTF space. In this way a DTF-distance
map \( d_i \) is obtained for every \( i \)-th scribble, which serves as a domain for determining the propagation paths. The local cost \( \rho \) from pixel \( x \) to \( y \) equals the \( y \) pixel’s value in the DTF distance map:

\[
\rho(x, y) = d_i(y). \tag{5}
\]

Examples of DTF-distance maps obtained for the Landscape and Cheetah images in Fig. 1 are presented in Fig. 3. Flowchart of the colorization process using DTF color propagation domain is demonstrated in Fig. 4.

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 \( d_i(x) \) would be used instead of the total path cost \( C_i(x) \) in (3). However, such an approach does not guarantee continuity of the regions and therefore has not been applied in the investigated cases.

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 [2], with which the paths are expected to minimize the gradient integrated along the propagation direction. An example of a difference between these two alternative approaches is given in Fig. 5 for an artificial monochrome image with annotated scribbles (a). Propagation paths leading from a scribble to a selected pixel are obtained using two methods. The path determined using our method (b) does not leave the striped area, which allows to colorize the image correctly (c). The result obtained using a conventional method (d) shows that the texture information is not taken into account during the propagation, which results in unexpected outcome (e).

In some cases the drawback of the presented method lies in the precision. The pixels may be misclassified near the boundaries of regions having different texture. This is because the local costs \( \rho \) are obtained in kernels of size \( 11 \times 11 \), and if the kernel spans over two different textures, the propagation boundary may be shifted compared to a real boundary of a texture region. This results in observing small “halos” at the region boundaries. We have approached this problem by determining a map of equal DTF distances, which indicates the boundaries of regions classified to individual scribbles. Then, in proximity of the boundaries, the propagation domain is changed to the conventional one, proposed by Yatziv [2]. Technically, when the propagation path reaches a pixel that is close enough to a boundary (i.e. less than the kernel size), the local costs are computed as in [2]. This allows to eliminate the aforementioned effect, because the Yatziv’s local cost is based on the gradient direction and magnitude obtained in \( 3 \times 3 \) kernels. An example of reducing the “halo” effect is presented in Fig. 6.

4.3 DTF-Based Color Transfer

Discriminative textural features can also be used for transferring colors from a given reference image to a source image or to multiple source images. Here, the feature space is generated based on the reference image with annotated scribbles. It may be a color image, in which a user marks the most distinctive regions supposed to be the transfer source, but it also may be a monochrome image (e.g. a selected frame from a video sequence) annotated with color scribbles. The procedure is as follows:

1. Generate DTF space for a scribbled reference image \( I_{ref} \) as specified in Sect. 4.1.2. This produces LDA projection matrix \( \Phi_{ref} \).
2. Transform every source image \( I_{src} \) into DTF-features image using the LDA projection matrix \( \Phi_{ref} \), obtained in Step (1).
3. Randomly select up to 50 reference pixels \((r_{ij})\) from each scribble \( s_i^{(ref)} \) in the image \( I_{ref} \).
4. For every source pixel \( x \in I_{src} \) compute distance in the DTF space to every reference pixel and associate
it with the closest reference pixel \( r_{ij} \). Furthermore, find the closest reference scribble \( re_{ref}(x) = \hat{s}^{(trans)}_k \), based on an average distance to all reference pixels that belong to each individual scribble: \( d(x, \hat{s}_k) = \bar{d}(x, r_{ij}) \). If the closest reference pixel belongs to the closest scribble \( (i = k) \), the source pixel is classified to the \( i \)-th scribble class and matched with the reference pixel \( r_{ij} \). Otherwise, the pixel remains unclassified and unmatched.

5. Transfer scribbles to the source image. Determine the threshold based on the distribution of closest distances obtained for all of the source pixels \( d_{th} \) (e.g., the upper quartile). Then, if a source pixel \( x \) has been classified in Step (4) to an \( i \)-th class and the closest distance is smaller than the threshold: \( d(x, re_{ref}(x)) < d_{th} \), add the pixel to the new transferred scribble \( \hat{s}^{(trans)}_i \).

6. Transfer color to all the pixels which belong to the transferred scribbles, based on their match determined in Step (4).

7. Colorize the source image from the transferred scribbles following the procedure described in Sect. 4.2.

This procedure selects the most similar pixels in the source image with respect to the textural features and transfers scribbles from the reference image to the source image. An example of transferred scribbles is presented in Fig. 7. A scribbled reference image (Cheetah from Fig. 1) is analyzed and DTF space created. Then, two scribbles from the reference image are transferred to the source image: (a) — ‘grass’ scribble and (b) — ‘cheetah’ scribble. Also, the colors are transferred within the region covered by the transferred scribbles and the result is shown in (c). After that, the image is colorized from the transferred scribbles as specified earlier in Sect. 4.2. A flowchart of the color transfer procedure is outlined in Fig. 8. The LDA projection matrix is obtained from the reference image as presented on the flowchart in Fig. 4.

5. Experimental Validation

5.1 Scribble-Based Colorization

We compared the proposed method with two well-established colorization techniques proposed by Levin[1] and Yatziv[2]. 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).

Several examples of colorization result achieved using our method and alternative algorithms are given in Fig. 9. We validated our algorithm using three types of images: 1) artificial to verify the theoretical assumptions (1st column), 2) semi-artificial composed of various textures (2nd column), and 3) photographs to assess applicability of our approach. Their sizes range from \( 280 \times 180 \) to \( 768 \times 704 \) 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 highest precision. Small imprecisions can be noted for the artificial images, but they are definitely smaller than for the alternative methods. Most of the photographs have been colorized perfectly using our method, and the halo effect mentioned in Sect. 4.2 has been eliminated.

The time needed to complete the colorization process has not exceeded 20 s 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 color propagation methods.

5.2 Color Transfer and Video Colorization

Color transfer was tested both for still images and video sequences, and compared with the conventional Welsh’s method [7]. An example of color transfer between two still images is presented in Fig. 10. A reference image (a) is annotated with two scribbles (marked as red), from which the colors were transferred to two similar images using Welsh’s (b, d) and proposed (c, e) method. For fair comparison, the reference pixels for the Welsh’s method were chosen randomly from the scribbled regions. It can be observed that although small imperfections occur using our method, the result is much better in terms of visual attractiveness and naturalness.

In order to objectively measure the quantitative results as well, we also transferred colors to the reference image itself and verified how similar the transfer result is to the original color image. We measured the similarity using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [23, 24], and Normalized Color Distance (NCD) [25]. Five examples are shown in Fig. 11. The top
Fig. 9  
Input image with scribbles and colorization results obtained with Levin’s [1], Yatziv’s [2] and our algorithm.

Fig. 10  
Reference image with annotated red scribbles (a) and transfer results obtained using Welsh’s [7] (b, d) and our (c, e) method.

Fig. 11  
Colors transferred back to the reference images (top row) using Welsh’s method (middle row) and our algorithm (bottom row).

row presents original color images with annotated scribbles, the results obtained using Welsh’s algorithm are in the middle row and our results are in the bottom row. The similarity scores are given in Table 1, in which we also show the gain achieved over the conventional Welsh’s method. It can be concluded that our algorithm delivers better results both in terms of quantitative and visual assessment.

We also used color transfer procedure for colorizing video sequences. An example is presented in Fig. 12†. A single frame from the sequence was selected as a reference image (left), from which the colors were transferred. The upper row demonstrates the result achieved using our algorithm, and the bottom row shows the result generated with the Welsh’s algorithm. It can be seen from the figure that our result is more natural and visually attractive. This has been also confirmed by quantitative results presented in Table 2, obtained based on original ground-truth data. The table presents an average value obtained for all the frames in the sequence, as well as the standard deviation. Also, the

†The video sequence was downloaded from http://trace.eas.asu.edu/yuv.
easier and faster, and reduces required human assistance. This makes the colorization process robust to outliers in the colorized images, and performs well even for sparse scribble annotation. Experiments have confirmed that our solution greatly improves the colorization process, and performs well for cases not only exploiting their textural characteristics, but also having similar chrominance. The method performs well for individual case and using them to match the pixels supposed to have similar chrominance. The method performs well for images rich in multi-colored details.

6. Conclusions and Future Work

In this paper we have presented a novel method for scribble-based image and video colorization. Our main contribution lies in extracting discriminative textural features for every individual case and using them to match the pixels supposed to have similar chrominance. The method performs well for processing large regions, regardless of whether they can be segmented using luminance or textural properties. The experiments have confirmed that our solution greatly improves the colorization process, and performs well even for sparse scribble annotation. This makes the colorization process easier and faster, and reduces required human assistance.

During our future work we are planning to explore how to extend the presented framework so that it matches the pixels not only exploiting their textural characteristics, but also using specific local features. This will make it also applicable to images rich in multi-colored details.

Table 1

| Image   | PSNR     | SSIM     | NCD     |
|---------|----------|----------|---------|
| Cheetah | 28.15    | 0.802    | 0.064   |
|         | ±0.011   | ±0.053   | ±0.053  |
| Flower A| 22.04    | 0.71     | 0.083   |
|         | ±1.8     | ±0.851   | ±0.073  |
|         | ±0.009   | ±0.142   | ±0.009  |
| Flower B| 22.43    | 0.832    | 0.068   |
|         | ±2.1     | ±0.91    | ±0.062  |
|         | ±0.007   | ±0.079   | ±0.007  |
| Hedgehog| 22.51    | 0.724    | 0.091   |
|         | ±2.0     | ±0.965   | ±0.064  |
|         | ±0.027   | ±0.241   | ±0.027  |
| Waterfall| 26.08   | 0.86     | 0.05    |
|         | ±1.4     | ±0.95    | ±0.031  |
|         | ±0.019   | ±0.09    | ±0.019  |

Table 2

| Image   | PSNR     | SSIM     | NCD     |
|---------|----------|----------|---------|
| Welsh   | 21.87 ± 0.1| 0.626 ± 0.004| 0.129 ± 0.002|
|         | ±1.23 ± 1.29| ±0.021 ± 0.023| ±0.023 ± 0.022|
|       | Gain      |          |         |
| DTF     | 23.11 ± 1.29| 0.737 ± 0.021| 0.12 ± 0.023|
|         | ±1.27     | ±0.011    | ±0.009 ± 0.022|
|         | Gain      |          |         |

Table 2 Similarity scores between the ground-truth and colorized images from a video sequence in Fig. 12.

References

[1] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” SIGGRAPH, pp.689–694, 2004.
[2] L. Yatziv and G. Sapiro, “Fast image and video colorization using chrominance blending,” IEEE Trans. Image Process., vol.15, no.5, pp.1120–1129, 2006.
[3] P. Lagodzinski and B. Smolka, “Digital image colorization based on probabilistic distance transform,” ELMAR, pp.495–498, 2008.
[4] V. Konushin and V. Vezhnevets, “Interactive image colorization and recoloring based on coupled map lattices,” GraphiCon, pp.231–234, 2006.
[5] T. Kim, K. Lee, and S. Lee, “Edge-preserving colorization using data-driven random walks with restart,” IEEE ICIP, pp.1661–1664, 2009.
[6] M. Kawulok and B. Smolka, “Competitive image colorization,” IEEE ICIP, pp.405–408, 2010.
[7] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” ACM Trans. Graph. (TOG), vol.21, no.3, pp.277–280, 2002.
[8] U. Lipowezky, “Grayscale aerial and space image colorization using texture classification,” Pattern Recognit. Lett., vol.27, no.4, pp.275–286, 2006.
[9] Y. Xiang, B. Zou, and H. Li, “Selective color transfer with multi-source images,” Pattern Recognit. Lett., vol.30, no.7, pp.685–689, 2009.
[10] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” IEEE Comput. Graph. Appl., vol.21, no.5, pp.34–41, 2001.
[11] C. Yao, X. Yang, L. Chen, and J. Wang, “Color transfer via local binary patterns mapping,” Proc. 17th IEEE Int. Image Processing (ICIP) Conf., pp.1825–1828, 2010.
[12] R. Gonzalez and R. Woods, Digital Image Processing, Addison Wesley Publishing, 1987.
[13] D. Sykora, J. Burianek, and J. Zara, “Unsupervised colorization of black-and-white cartoons,” SIGGRAPH, pp.121–124, 2007.
[14] T. Horiiuchi and H. Koteru, “Colorization for monochrome image with texture,” Proc. 13th Color Imaging Conference, pp.201–210, 2005.
[15] R. Ironty, D. Cohen-Or, and D. Lischinski, “Colorization by Example,” Proc. Eurographics Symposium on Rendering, pp.201–210, 2005.
[16] Y. Qu, T.T. Wong, and P.A. Heng, “Manga colorization,” ACM Trans. Graph. (SIGGRAPH 2006 issue), vol.25, no.3, pp.1214–1220, July 2006.
[17] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.Q. Xu, and H.Y. Shum, “Natural image colorization,” Proc. Eurographics Symposium on Rendering, ed. J. Kautz and S. Pattanaik, pp.309–320, June 2007.
[18] G. Charpiat, M. Hofmann, and B. Schölkopf, “Automatic image colorization via multimodal predictions,” European Conference on Computer Vision, ed. D. Forsyth, P. Torr, and A. Zisserman, Marseille, France, Sept. 2008.
[19] X. Liu, L. Wan, Y. Qu, T.T. Wong, S. Lin, C.S. Leung, and P.A. Heng, “Intrinsic colorization,” ACM Trans. Graph., vol.27,
Michal Kawulok graduated and received the M.Sc. degree and the Ph.D. degree from Silesian University of Technology, Gliwice, Poland in 2003 and 2007, respectively. Since 2007 he is an Assistant Professor at the Institute of Informatics, Silesian University of Technology. Apart from image colorization, his main research interests include face recognition, skin segmentation, and gesture recognition.

Jolanta Kawulok graduated and received the Eng. and M.Sc. degrees in 2009 and 2010, respectively, from the Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland. Currently she is a Ph.D. student in the Institute of Informatics at Silesian University of Technology. Her main research interests include applications of data mining algorithms to images, genome sequences and biomedical data.

Bogdan Smolka received the Diploma degree in physics from the Silesian University, Katowice, Poland, in 1986 and the Ph.D. degree in automatic control from the Department of Automatic Control, Silesian University of Technology, Gliwice, Poland, in 1998. From 1986 to 1989 he was a Teaching Assistant at the Department of Biophysics, Silesian Medical University, Katowice. From 1992 to 1994, he was a Teaching Assistant at the Technical University of Gesslingen, Germany. Since 1994, he has been with the Silesian University of Technology. In 1998, he was appointed as an Associate Professor at the Department of Automatic Control. In 2007 he was promoted to Professor of Silesian University of Technology. He has published over 200 papers in refereed journals and conference proceedings on digital signal and image processing. His current research interests include low-level color image processing, human-computer interaction and visual aspects of image quality.