

Isoline Based Image Colorization

Adam Popowicz
Institute of Electronics
 Silesian University of Technology
 Gliwice, Poland
 apopowicz@polsl.pl

Bogdan Smolka
Institute of Automatic Control
 Silesian University of Technology
 Gliwice, Poland
 bogdan.smolka@polsl.pl

Abstract—Colorization is a process of adding colors to black and white images and videos. Each scalar pixel value is replaced by three values which describe, depending on color mode, e.g. the luminance and the chrominance or red-green-blue components. By introducing colors, the resulting image benefit greatly. Not only that it is more attractive for a viewer, but also the perception of hardly noticeable features increases, which is crucial e.g. in medical image analysis.

In our paper we present a novel algorithm for grayscale image colorization. The idea is based on the concept of isolines on geographical maps. For each color indicated by a user with scribbles, we create distance maps which present the measure of distance between a pixel and the colored region. The resulting color is calculated as a weighted average of all colors indicated by the user in a form of scribbles. The method is computationally efficient and can be adjusted by several parameters to obtain optimal colorization results. We present several multimedia and biomedical images, colorized using our approach.

Keywords—image processing; colorization; segmentation

I. INTRODUCTION

Colorization is an automatic or interactive process of assigning colors to the pixels of a grayscale image. It was for the first time introduced by W. Markle as a method of enhancing still images and video sequences [1]. With a use of modern technologies, such a task is nowadays usually performed by a computer system. However, full automation is only possible if any reference (similar) image is available. If not, then a user has to define colors of each part of a picture. It is usually achieved by introducing color scribbles into the grayscale image. Such a process, called the scribbling, usually has strong impact on final colorizing results.

In colorization process, the intensity of each pixel is replaced by three components representing a color. Depending on calculation methodology, several models are available, for example: RGB Red, Green and Blue, YCrCb, YUV luminance and two chrominance channels, HSV Hue, Saturation and Value [2]. For colorization purpose, the second representation is mostly suitable, because the luminance channel is directly obtained from a grayscale image. Two remaining channels are to be calculated by a particular algorithm. Although the luminance-chrominance

representation can be easily adopted for colorization, all models can be utilized, but they need time consuming transformations for proper brightness preservation.

The article is organized as follows. In Section 2 we present a review of the interactive colorization algorithms. In Section 3 we introduce our concept and its implementation. In the next Section we show exemplary images colorized with our method and a brief comparison with two other methods. We also present the segmentation ability of our framework. In the last Section we summarize the results.

II. STATE-OF-THE-ART COLORIZATION METHODS

In general, colorization methods can be divided in two categories: fully automatic algorithms and interactive, which need human interaction. Methods of the first type usually try to map brightness level on particular color or use reference color image and compare it with a grayscale one [3]–[5]. The second type of methods needs color hints assigned by a user. Because our method utilizes a user-defined scribbles, we present a brief review of similar, interactive algorithms, which can be compared with our approach.

In [6], [7] authors present a method of colorization based on the minimization of color difference between adjacent pixels of similar intensity. Starting with scribbles indicated by an operator, their algorithm minimizes the difference between the color in particular pixel and the weighted average of colors of neighboring pixels. The minimization formulated this way can be solved with standard optimization approaches like the Least Squares method. The presented idea can be easily extended to time domain for video colorization and therefore it is one of the most popular colorization technique and was exploited for video purposes [8], as well as for biomedical image processing [9].

In [10] a method of seed pixels propagation is presented. The color is transferred from a seed (scribbled by a user) to its neighbors basing on local Markov property of the image. The algorithm propagates seed colors from indicated pixel to the neighbor by minimizing color difference of particular pixel's pair in RGB space.

An approach of shortest path calculations is described in [11]. The authors make use of Dijkstra algorithm to specify the smoothness of a digital path between two image points.

They integrate the absolute value of luminance gradients encountered on the shortest path from scribbled pixels to the others and finally they calculate resulting chrominance as a weighted average of the costs of the best paths from each scribbled color. The algorithm is simple and effective what was confirmed on several test images.

In [12] the authors present the utilization of morphological distance transformation for proper colorization. They suggest a modification of the standard distance transformation by introducing additional cost for a transition between two neighboring pixels which is proportional to the intensity changes. Using such a method they were able to provide another effective method allowing for high quality colorization results of still images.

The distance transformations approach is used also in [13]. In this method the authors utilize their algorithm before applying the fuzzy clustering of the pixels. With initial local hints introduced by a user using color scribbles, they obtain the initial clusters' centers and calculate the fuzzy membership based on the estimation of Euclidean distance provided by the transformations.

In [14] the authors use the method of watershed segmentation [15] so that pixels are divided into unique segments. After the segmentation, they perform a merge operation by flooding a large number of regions, till their number reaches the number of color markers initially introduced by a user on the grayscale image.

In [16]–[18] the concept of texture-dependent colorization is presented. Unlike other methods, which are based mostly on luminance of grayscale image, the authors analyze the texture features and perform the scribble-based colorization adaptively. They utilize two types of distance metrics. The first - suitable for smooth regions and the second, which prefers highly textured parts of the image. The type of a metric is selected depending on scribbled part of the image.

III. ISOLINES-BASED COLORIZATION

Our method of colorization utilizes the concept of isolines on the geographical maps, (see Fig. 1). Like in the isolines one indicates the locations with e.g. the same height; in our solution the measure of deviation from a scribble is showed. For a given pixel, (so-called seed pixel, the one within a scribble), we create a distance map based on the intensity difference between the pixels and the seed. The distance for each pixel represents the maximum intensity deviation encountered on the shortest path starting from the seed pixel and ending in current point

$$D(p_i, p_j) = \min_{\Phi} \{ \max |p_k - p_i|, k \in \Phi \}, \quad (1)$$

where p_i and p_j are pixel intensities, Φ is a set of adjacent pixels forming a path between pixels at positions i and j .

An example of distance calculations for a simple one-dimensional path is shown in Fig. 2. In this example only

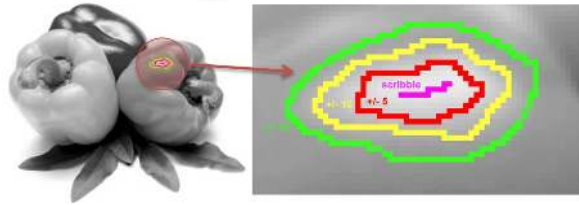


Figure 1: The visualization of isolines within a grayscale image

one path from seed pixel is possible. However, for an image one can point several possible paths between two pixels. In our method the distance is minimal which means we choose only the shortest path in terms of the brightness difference. In our version of the algorithm we allow only steps between adjacent pixels. Exemplary two-dimensional *minimal path* calculations are presented in Fig. 3.

Pixel brightness	10	9	10	12	8	7	9
Isoline distance	x	1	1	2	2	3	3

Figure 2: Distance calculations. The first pixel is selected as a seed (shaded).



Figure 3: Two-dimensional case of isoline calculation. Blue arrows - best path between the seed (yellow) and a pixel (green). Calculated isoline distance is indicated by red values.

For grayscale images with only one seed pixel, the minimization of the paths and calculation of our distance map could be performed with e.g. Dijkstra algorithm. However, such a method would require much computational time, as there are many seed pixels due the structure of the scribbles. Thus we propose a novel efficient algorithm for creating such distance maps utilizing the propagation of intensity difference and the seed brightness encountered on the path.

The algorithm scans an image twice: 1) from left to right, up to bottom, and 2) from right to left, bottom to top. In

each scan in a single step we analyze the seed intensity and the distance value of adjacent pixels (depending on the scan). We compute the intensity difference between current pixel and the seed intensity from neighboring pixels and finally, depending on its value, we assign minimum possible distance and propagate corresponding seed brightness. Thus we operate on three matrices: the gray-scale image intensity matrix, the distance matrix and the seed intensity matrix. According to the assumptions of our method, the distance value for a given pixel cannot be lower than the distance from preceding pixel because the intensity difference on the path can only rise. The resulting distance maps for several scribbles within the image are presented in Fig. 4.

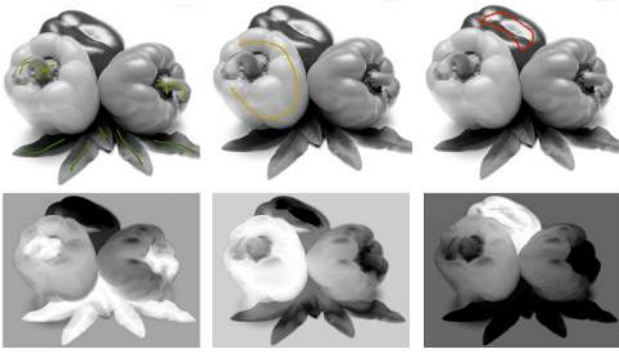


Figure 4: Distance maps calculated for exemplary scribbles.

To obtain the resulting color of a pixel we calculate the weighted average of each color from the seeds. The weights are the values from corresponding distance maps. We make use of exponential weighting function. By operating on YCrCb color representation we take the Y component from the grayscale image. The chrominance is the following weighted sum:

$$C = \frac{\sum_{i=1}^N C_i \exp\left\{\frac{-D_i}{\sigma_i}\right\}}{\sum_{i=1}^N C_i}, \quad (2)$$

where C is the chrominance of a given pixel, C_i is the chrominance of i -th color, D_i is the calculated distance from i -th color, σ_i is the parameter responsible for the level of color blending.

During the very first experiments we observed, that in some images the colors propagate far away from the corresponding scribbles and the resulting colorized image seem to be blurred (see Fig. 5). It was due to the fact, that no step cost was introduced and, especially for seed pixels of intensity in the middle of a grayscale range, there is a tendency to influence the color composition at a long distance from the seed. Thus we decided to add an extra parameter δ - limiting the possibility of such situations. The δ is a simple cost function adding the penalty for each transition on a digital

path between the pixels and the Eq. (1) is modified:

$$D(p_i, p_j) = \min_{\Phi} \{ \max_{k \in \Phi} |p_k - p_i| + \delta \cdot n, k \in \Phi \}, \quad (3)$$

where n is a length of digital path Φ . In this way we obtained visually far better results. The effect of applying additional cost is visible in Fig. 5.

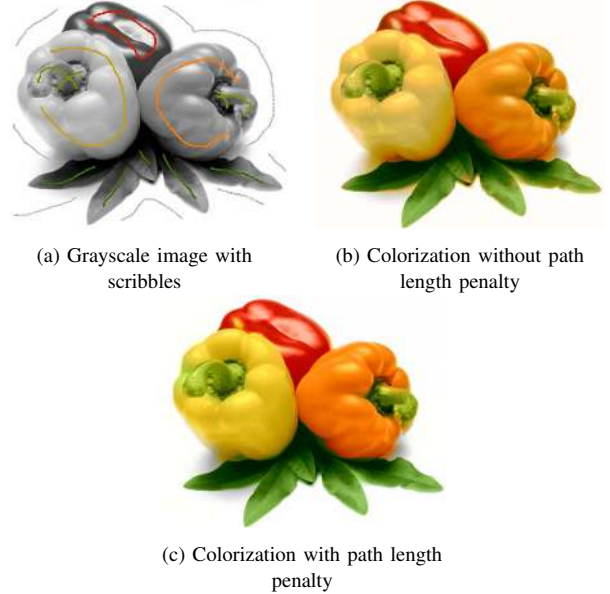


Figure 5: Results of colorization after introducing a path length penalty.

On the next page we present a pseudocode of our algorithm.

IV. EXEMPLARY RESULTS

The results of our method are presented in Fig. 6. We used three sample images as a benchmark test for our algorithm. As can be seen, it is extremely hard to distinguish between the original color version and the colorized one.

We compared the results of our colorization with the outcomes of other algorithms presented in [11] and [12]. With the same scribbles configuration (presented in left column of Fig. 6) we obtain very similar results. However, after applying a zoom it is easily visible, that white color seems to spill from the eyes in the images obtained by two algorithms taken for comparison. It could be a result of a specific scribble configuration or parameter selection, but it, for sure, confirms that our algorithm can be a competitive solution.

As it was mentioned at the beginning of this paper, the colorization algorithm can be used as a segmentation method for biomedical images. Due to the collaboration within ongoing grant on *Automated Assessment of Joint Synovitis Activity from Medical Ultrasound and Power Doppler Examinations using Image Processing and Machine Learning*

Definitions:

X, Y - width and height of image
 $I(x_i, y_i)$ - brightness of a pixel at (x_i, y_i)
 $\Omega = (x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)$ - set of all pixels' positions within scribbles
 δ - step cost penalty

Initialization:

```

if  $(x_i, y_i) \in \Omega$  then
     $DistMap(x_i, y_i) = 0;$ 
     $SeedMap(x_i, y_i) = 0;$ 
else
     $DistMap(x_i, y_i) = \infty;$ 
     $SeedMap(x_i, y_i) = \infty;$ 
end

```

First scan (left to right, top to bottom):

```

for  $I = 2 : Y - 1$  do
    for  $j = 2 : X - 1$  do
         $FilterPos = \{(i - 1, j), (i, j - 1)\};$ 
         $NewDistMap =$ 
         $\delta + \min\{DistMap(FilterPos) +$ 
         $(SeedMap(FilterPos) >$ 
         $I(i, j)) \cdot (SeedMap(FilterPos) - I(i, j))\};$ 
         $NewSeedMap = SeedMap(FilterPos')$ 
        where  $FilterPos'$  is the best position from
        above minimization;
        if  $NewDistMap < DistMap(i, j)$  then
             $DistMap(i, j) = NewDistMap;$ 
             $SeedMap(i, j) = NewSeedMap;$ 
        end
    end
end

```

Second scan (right to left, bottom to top):

```

for  $I = Y - 1 : 2$  do
    for  $j = X - 1 : 2$  do
         $FilterPos = \{(i + 1, j), (i, j + 1)\};$ 
         $NewDistMap =$ 
         $\delta + \min\{DistMap(FilterPos) +$ 
         $(SeedMap(FilterPos) >$ 
         $I(i, j)) \cdot (SeedMap(FilterPos) - I(i, j))\};$ 
         $NewSeedMap = SeedMap(FilterPos')$ 
        where  $FilterPos'$  is the best position from
        above minimization;
        if  $NewDistMap < DistMap(i, j)$  then
             $DistMap(i, j) = NewDistMap;$ 
             $SeedMap(i, j) = NewSeedMap;$ 
        end
    end
end

```

Algorithm 1: The pseudo code for isoline algorithm.

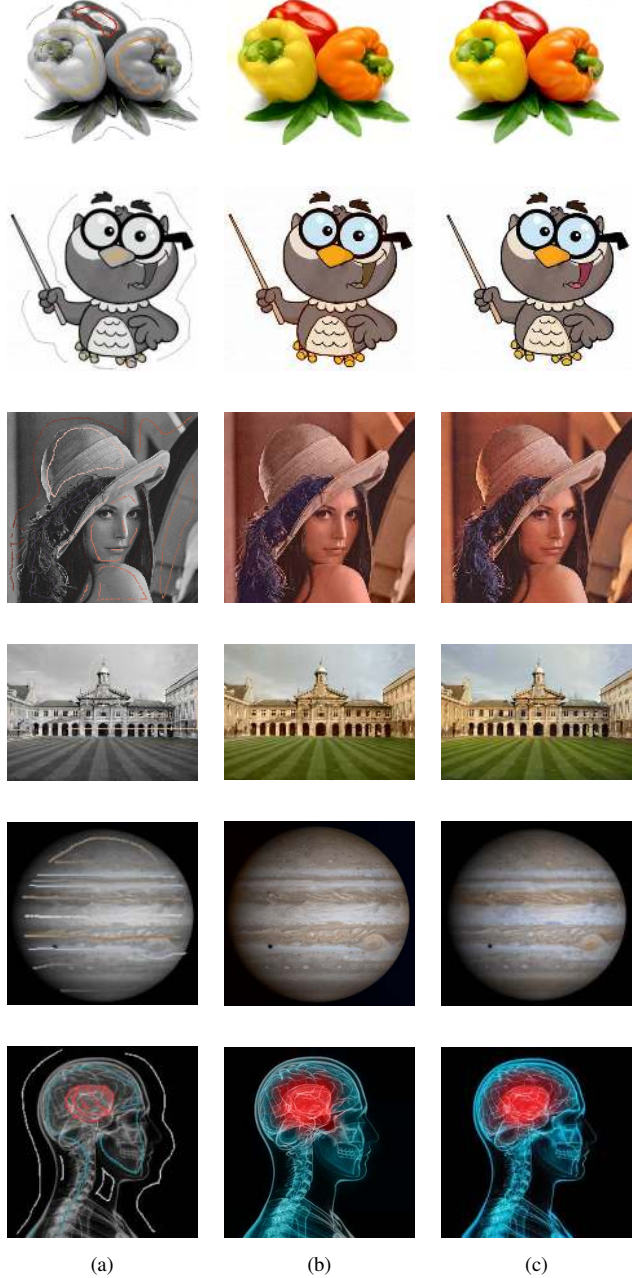


Figure 6: Comparison of colored and original images. (a) - a grayscale image with the scribbles, (b) - the result of colorization using our algorithm, (c) - original color image

Methods *MEDUSA* carried by the Polish-Norwegian Research Programme, we used an exemplar USG image with visible joint synovitis.

The contour of the image part, representing the affected tissue, was introduced by an experienced physician and is visible in Fig. 8(a). With a single or double seed points within this area and the white scribble outside the region of interest (see Fig. 8(b)), we were able to provide very

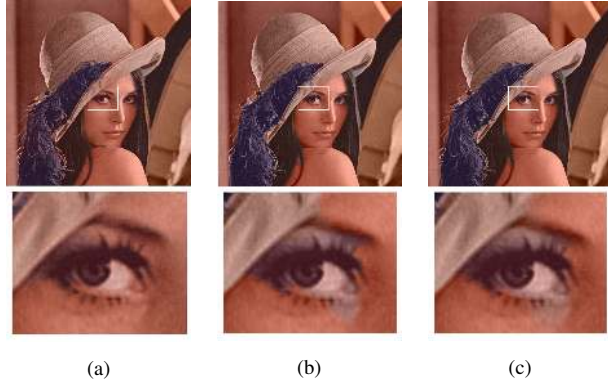


Figure 7: Comparison of colorized images: (a) - isoline algorithm, (b) - chrominance blending algorithm [11], (c) - modified distance transformation algorithm [12].

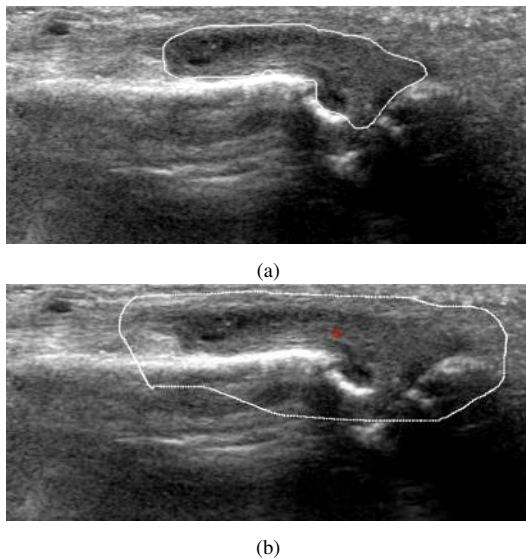


Figure 8: USG synovitis image with a disease region indicated (a), and scribbles configuration (b).

interesting results using our algorithm. In Fig. 9 we show a range of possible results for varying step cost δ .

The synovitis region is hardly visible and only experienced doctors dealing with this kind of disease can indicate the affected parts precisely. We show that our algorithm, with properly chosen value of δ marks the region of interest with good accuracy. This kind of utilization of the method presented in this paper will be extensively investigated in the nearest future.

V. SUMMARY

In the the paper we present a novel method of grayscale image colorization. We utilize an interactive algorithm where user indicates color scribbles within an image. We utilized the isoline concept of geographical maps where similar points are given the value corresponding to the same property like height. In our proposition we assign the pixels the

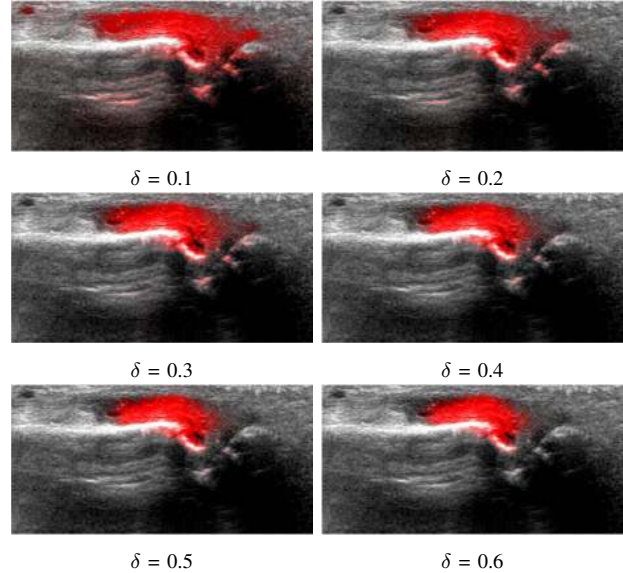


Figure 9: Results of USG image colorization for several step cost values δ .

value representing the maximum deviation of intensity on a path from a scribble to given pixel.

We implemented a double scan algorithm and seed transferring concept for proper distance calculation. The algorithm is very fast and not complicated. We hope, the pseudo-code presented in the article, will help with implementation of our method.

The results of colorization of multimedia images prove the accuracy of the presented idea. The colorized images are indistinguishable from their original color versions. We also used the proposed algorithm for exemplary USG image, where it produced very encouraging result in term of proper joint synovitis segmentation. The investigation of possibility of applying our method for such a purpose is the key element in our future research.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Norwegian Financial Mechanism 2009-2014 under Project Contract No. Pol-Nor/204256/16/2013.

REFERENCES

- [1] W. Markle, "The development and application of colorization," *Society of Motion Picture and Television Engineers*, vol. 93, no. 7, pp. 632–635, 1984.
- [2] J. Russ, *The Image Processing Handbook, Sixth Edition*. CRC Press, Taylor & Francis Group, 2011.
- [3] W.-Q. Yan and M. Kankanhalli, "Colorizing infrared home videos," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, July 2003, pp. 1–97–100 vol.1.
- [4] Z. Zhen, G. Yan, and M. Lizhuang, "An automatic image and video colorization algorithm based on pattern continuity," in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, July 2012, pp. 531–536.

- [5] Y. Zhao, L. Wang, W. Jin, and S. Shi, "Colorizing biomedical images based on color transfer," in *Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on*, May 2007, pp. 820–823.
- [6] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, Aug. 2004.
- [7] —, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 689–694.
- [8] V. Rao Veeravasarapu and J. Sivaswamy, "Fast and fully automated video colorization," in *Signal Processing and Communications (SPCOM), 2012 International Conference on*, July 2012, pp. 1–5.
- [9] A. Shah, M. Gandhi, and K. Shah, "Medical image colorization using optimization technique," *International Journal of Scientific and Research Publications*, vol. 3, no. 3, 2013.
- [10] T. Horichi and S. Hirano, "Colorization algorithm for grayscale image by propagating seed pixels," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1, Sept 2003, pp. I–457–60 vol.1.
- [11] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1120–1129, May 2006.
- [12] P. Lagodzinski and B. Smolka, "Application of the extended distance transformation in digital image colorization," *Multimedia Tools and Applications*, pp. 1–27, 2012.
- [13] Z. Zhang, H. Cui, H. Lu, R. Chen, and Y. Yan, "A colorization method based on fuzzy clustering and distance transformation," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, Oct 2009, pp. 1–5.
- [14] V. Jacob and S. Gupta, "Colorization of grayscale images and videos using a semiautomatic approach," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, Nov 2009, pp. 1653–1656.
- [15] V. Osma-Ruiz, J. I. Godino-Llorente, N. Sáenz-Lechón, and P. Gómez-Vilda, "An improved watershed algorithm based on efficient computation of shortest paths," *Pattern Recogn.*, vol. 40, no. 3, pp. 1078–1090, Mar. 2007.
- [16] M. Kawulok and B. Smolka, "Competitive image colorization," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 405–408.
- [17] M. Kawulok, J. Kawulok, and B. Smolka, "Discriminative Textural Features for Image and Video Colorization," *IEICE Transactions on Information and Systems*, vol. 95, pp. 1722–1730, 2012.
- [18] M. Kawulok and B. Smolka, "Texture-adaptive image colorization framework," *EURASIP J. Adv. Sig. Proc.*, vol. 2011, p. 99, 2011.