Descriptor-based Image Colorization and Regularization

Dmitry Kuzovkin, Christel Chamaret, Tania Pouli

Descriptor-based Image Colorization and Regularization

Dmitry Kuzovkin[1,2], Christel Chamaret[1], and Tania Pouli[1]

[1]Technicolor Research and Innovation, Rennes, France
[2]Jean Monnet University, Saint-Étienne, France
kuzovkin.dmitry@gmail.com
{christel.chamaret,tania.pouli}@technicolor.com

Abstract. We propose a new, fully automatic method for example-based image colorization and a robust color artifact regularization solution. To determine correspondences between the two images, we supplement the PatchMatch algorithm with rich statistical image descriptors. Based on detected matches, our method transfers colors from the reference to the target grayscale image. In addition, we propose a general regularization scheme that can smooth artifacts typical to color manipulation algorithms. Our regularization approach propagates the major colors in image regions, as determined through superpixel-based segmentation of the original image. We evaluate the effectiveness of our colorization for a varied set of images and demonstrate our regularization scheme for both colorization and color transfer applications.

Keywords: Colorization, Regularization, Color Transfer

1 Introduction

Colorization — the process of adding color to grayscale content — is crucial for giving a new lease of life to legacy movies or photographs. Historically, colorization has been a time-consuming, manual task, however in recent years, automated solutions have emerged for colorizing content. Existing automatic colorization solutions rely on a user-provided reference image [16], a palette specifying the colors the image should obtain [11] or set of strokes [10] drawn directly on the image to define colors.

Our work falls within the first category, known as *example-based colorization*, and relies on a reference color image with semantically similar content to the target grayscale image. To transfer color information from the reference to the target, our method finds correspondences using a modified version of the PatchMatch algorithm [2]. Although typically this algorithm considers only color information in the image, this often does not provide enough distinguishing information when the two images are not depicting the same scene. To that end, we supplement PatchMatch with a series of statistical descriptors that provide a rich representation of the structure in the two images.

In addition to our colorization method, we propose a novel regularization scheme to remove artifacts typical to such color manipulations. In processes such as colorization or color transfer, small discontinuities in the luminance (e.g. due to compression artifacts) may lead to corresponding anomalies in chromaticities. To remove such artifacts, regularization methods aim to smoothly propagate colors within coherent regions. Our solution relies on superpixel segmentation and subsequent merging to robustly divide the image into coherent areas. A stroke is automatically created for each region and propagated depending on adjacent luminance information [10].

Experiments conducted on various types of images demonstrate that our method can provide visually appealing colorization results in different scenarios, competitive with the state of the art, at a lower computational cost. Additionally, we demonstrate the usefulness of our regularization approach in the context of color modification methods. Our work offers the following contributions:

- A novel, fully automatic colorization approach
- A combination of image descriptors that allows using a wider set of images as references in colorization
- A regularization scheme for correcting artifacts typically seen after color processing such as colorization or color transfer

2 Related work

Given a reference color image, example-based methods aim to find correspondences with the grayscale target in order to define how colors should be distributed. Earlier methods rely only on luminance information to determine correspondences [16], which can often fail when corresponding areas (e.g. sky) do not have corresponding luminance. Additionally, small irregularities in the luminance can lead to spatial consistency issues.

To address some of these limitations, the higher-level context of each pixel needs to be considered. Irony *et al.* [8] use a supervised classification scheme based on image features and can provide plausible colorization results, however it requires manual segmentation into corresponding regions, making it ill-suited for automatic colorization. To obtain better correspondences, Charpiat *et al.* [4] further exploit the idea of feature descriptors by using SURF descriptors incorporated into a probability estimation model. Probabilistic information is also used by Bugeau *et al.* [3], where a variational energy minimization framework is employed to simultaneously find candidate colors and regularize results.

The closest to our approach is the work of Gupta *et al.* [7], which relies on a cascade feature matching scheme to find correspondences between reference and target images. The spatial consistency of matching is improved by a voting step, which is based on mean-shift segmentation together with k-means clustering. Each segment is assigned with a color, and obtained colors are used to produce micro-scribbles in the center of each segment. These are propagated across the entire image using the algorithm by Levin et al. [10]. In contrast to their approach, we use a smaller and more general set of descriptors, allowing us to colorize using a less restrictive selection of reference images. At the same time, we take advantage of the robustness of the PatchMatch algorithm to obtain correspondences, which leads to more accurate assignment of colors as will be shown in Section 5. Finally, we ensure a better propagation of colors in our regularization scheme by creating large skeleton-based strokes that span coherent areas in the image. The following sections will describe our colorization solution and regularization scheme in detail.

3 Colorization

The general scheme of the proposed method is shown in Figure 1. The input is represented by two images: a target grayscale image I_t to be colorized, and a reference color image I_r. The images are smoothed using a Gaussian filter before analysis to remove small artifacts, such as noise or painterly texture, that might influence the matching step (w = 5x5, $\sigma = 1.0$). Then, our method finds correspondences between the intensity channels of the two images (Y_t and Y_r respectively), using their descriptor representations. The obtained correspondences are used to map chroma information from the reference to the target image, producing an initial colorization result. Finally, a regularization step (Section 4) is applied to suppress color artifacts that might be present in the initial result.

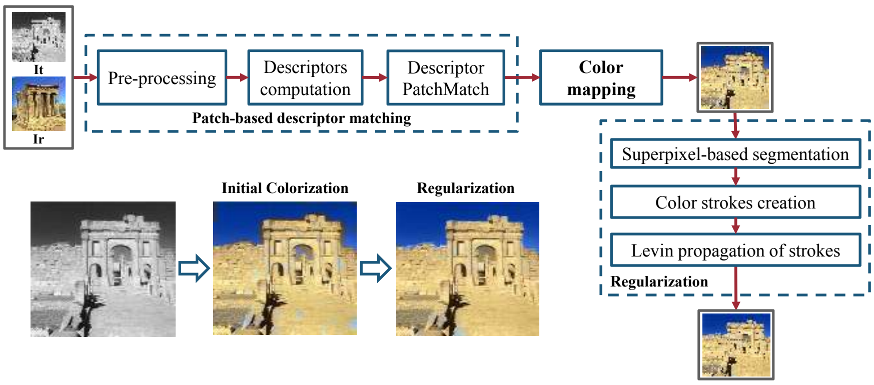

Fig. 1: Overview of the proposed colorization framework

3.1 Descriptor Computation

Intensity information by itself is not sufficient to provide reliable matches between similar but not identical objects in images, as it has been demonstrated since the pioneer work by Welsh *et al.* [16]. Even for identical objects, illumination changes or other factors might significantly affect their appearance. Image descriptors can enrich the available intensity information and represent object structures and textures in a more robust way. Thus, the influence of changes in object appearance is reduced, and more accurate matching can be achieved.

Our approach computes 38-dimensional descriptors D_t and D_r for the target and reference images respectively by concatenating grayscale intensity, texture gradients and histogram of oriented gradients descriptors. The descriptor vector for a pixel p is given by:

$$D(p) = D_{HoG}(p) \cup D_{tg}(p) \cup Y(p), \tag{1}$$

where $D_{HoG}(p)$ is the 31-D histogram of oriented gradients descriptor, $D_{tg}(p)$ is 6-D texture gradients descriptor and $Y(p)$ is the intensity for that pixel.

Grayscale Intensity. Although intensity information is not invariant to global changes in illumination between the two images, it can still provide useful information to guide matching. Therefore, in our algorithm the intensity $Y(p)$ of each pixel is used as an auxiliary descriptor.

Texture Gradients. Texture information around each pixel can help describe the local structure in the scene. To make use of such information we include texture gradients in our descriptor, relying on the approach of Martin *et al.* [13]. To compute this descriptor, responses for several filters are first collected for each pixel and categorized according to a set of representative precomputed responses known as textons [12]. Textons identify the presence of image structures, such as bars, corners or various levels of contrast. Once textons are computed, a disc is considered around each pixel and for each half of the disc, a texton histogram is computed. The distance between the histograms of each half, computed for 6 different orientations, forms the 6-D vector $D_{tg}(p)$.

Histogram of Oriented Gradients. The final descriptor used in our method is the histogram of oriented gradients (HoG), denoted by D_{HoG}, which is aimed at representing local object appearance and shape by the distribution of intensity gradients and edge directions [5][1]. To compute the HoG for a given pixel p, first, horizontal and vertical gradients are computed at that location. To compute the histogram for each pixel, a cell (4x4 pixels in our case) is considered around it, and the gradient of each pixel within the cell is allocated to one of 9 orientation bins. These responses are then locally normalized and processed into contrast-sensitive, contrast-insensitive and texture features, which are combined into a 31-dimensional HoG descriptor.

3.2 Descriptor PatchMatch

Once the descriptors D_r and D_t are computed for the reference and target images, correspondences can be determined between them. We rely on the PatchMatch algorithm [2] for this task but apply it on the 38-D space spanned by our descriptor. PatchMatch computes an approximate nearest neighbor field (NNF), which provides dense, global correspondences between image patches. The PatchMatch algorithm works over image patches, but the final correspondence field is

[1] We use the implementation from Yamaguchi: `http://www3.cs.stonybrook.edu/~kyamagu/software/misc/dense_hog.m`.

created at the pixel level. Given a pixel p_t in the target image, NNF provides us the matching pixel p_r in the reference image:

$$NNF(D_t \longleftrightarrow D_r) : p_t \rightarrow p_r \tag{2}$$

Descriptor-based matching is able to provide matches not only between same objects, but also between generally similar objects. In our approach, PatchMatch is used with a patch size of 9x9 pixels and 5 iterations, following the recommendations in [2], as after 4-5 iterations the NNF typically converges.

3.3 Color Mapping

Using the correspondences determined through PatchMatch, colors can be mapped from the reference to the target image. This is performed in the $YCbCr$ space and information from the Cb and Cr channels of the reference I_r are used to populate the corresponding channels of the target I_t, while intensity information in I_t is left unchanged. Given the correspondences defined by Eq. (2), chromatic information for a pixel p_t is given by:

$$C\{b|r\}_t(p_t) = C\{b|r\}_r(p_r) \tag{3}$$

After reconstructing the Cb and Cr channels of I_t, the image is converted back to RGB, producing an initial colorization result I_c. Although the global assignment of colors in I_c is correct in most cases at this stage of our algorithm, small local artifacts can appear, due to inaccurate correspondences. To correct such artifacts, a regularization step takes place, described in the next section.

(a) (b) (c) (d) (e)

Fig. 2: Regularization of the colorization result. (a) Initial colorization without regularization. (b) SLIC superpixels. (c) Superpixel-based segmentation. (d) Resulting color strokes. (e) Final regularized colorization after stroke propagation.

4 Regularization of Color Artifacts

To regularize color artifacts, first, the target image I_t is segmented using a superpixel-based approach. Then, strokes that span each segment are created and assigned with colors representing the color distribution of the corresponding segments from the initial colorization result I_c. The color strokes are marked on the I_t and propagated to the rest of the pixels using Levin's approach [10]. The results of main regularization steps are shown in Figure 2.

4.1 Superpixel-based Segmentation

The first step of our regularization scheme segments I_c into coherent areas based on intensity information. Using the SLIC method [1], the segmentation starts from the computation of a set of n superpixels $S = [s_1, ..., s_n]$ which span the image, where adjacent similar pixels form coherent superpixel units.

To avoid oversegmentation of large homogeneous regions, the obtained superpixels are further processed, producing a coarser set of segments S_{seg}. To compute each segment $s_{seg,i}$, adjacent superpixels with similar statistics are merged together, forming larger segments. Merging is based on the DBSCAN clustering algorithm [6], where for each superpixel s_i a set of adjacent superpixels N_i is considered. If a distance ϵ between s_i and an adjacent superpixel s_j does not exceed threshold T, s_j is taken as a new point of a cluster. Any other superpixel reachable from s_j is taken as part of the cluster as well. For grayscale images, ϵ is a distance based on simple statistics expressed by the mean and variance of the intensity within considered superpixels.

$$s_{seg,i} = s_i \bigcup_{j \in N_i} s_j \mid \epsilon < T, \quad \text{where} \quad \epsilon = \sqrt{(\mu_{s_j} - \mu_{s_i})^2 + (\sigma_{s_j} - \sigma_{s_i})^2}, \tag{4}$$

For the choice of parameters in this step, we use the values that provide a reasonable trade-off between over- and undersegmentation, given the lack of color information. The initial set of superpixels is computed using $n = 250$, and the threshold T in the clustering step is set to 2.75. Higher value of n would provide more detailed segmentation causing preservation of undesirable artifacts, whereas higher value of T might lead to excessive merging and diffusion of colors.

4.2 Color Stroke Creation and Propagation

Each segment obtained from the previous step is used to create a continuous stroke, which spans that segment and contains the dominant color of that area. The created strokes serve as input to the stroke-based colorization approach of Levin *et al.* [10], where colors of strokes placed on the grayscale image are smoothly propagated to the rest of the image. Since the stroke colors are taken from our initial colorization result I_c, this process smooths colors within coherent areas, removing wrong local assignments that might have created artifacts before.

The color propagation method of Levin *et al.* [10] relies on the premise that nearby pixels with similar intensities are likely to have the same color. Previous approaches that automatically create strokes for this process opt for micro-strokes placed in the center of each area [7]. In practice, we have found that larger strokes that span the area to be colorized can lead to better coverage once propagated. Based on that, to create strokes we compute the binary skeleton of each segment using the thinning morphological operation [9].

To determine the color of a stroke, the Cb and Cr values of each pixel within a segment are quantized into 48 clusters using k-means clustering, and the major chromatic component of a segment is assigned to the stroke. This clustering step is necessary to determine a single representative color for each region. Figure 2(d) shows the color strokes for an image as colorized skeleton shapes.

5 Colorization Evaluation

We compare results obtained from our method with the results of previous example-based colorization methods. Figure 3 depicts the final results of our combined colorization and regularization approach as well as results from the methods of Gupta *et al.* [7], Charpiat *et al.* [4] and Welsh *et al.* [16]. Input images are taken from [7] and represent typical scenes with semantically similar content, but demonstrating different characteristics of intensity, textures and overall appearance.

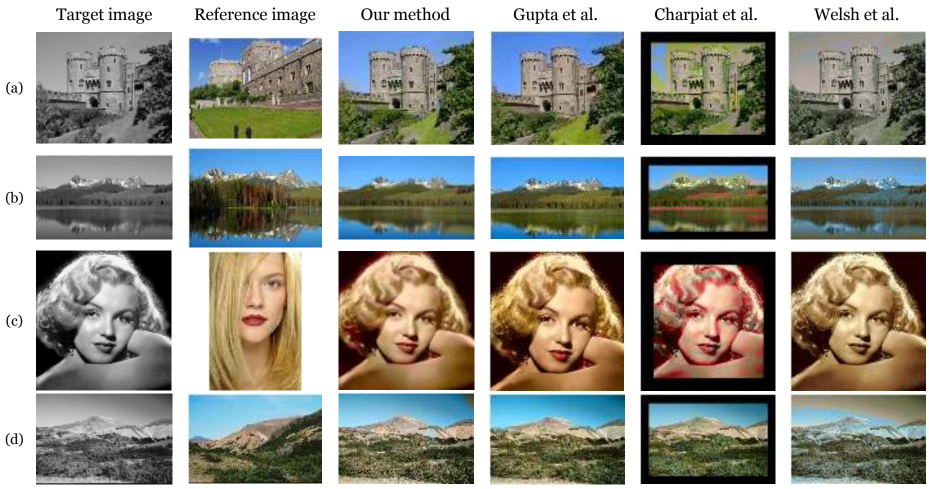

Fig. 3: Comparison with existing example-based colorization methods. Represented methods: Gupta *et al.* [7], Charpiat *et al.* [4], Welsh *et al.* [16]

Due to complex image content, the method by Welsh *et al.* [16] fails in almost all cases since it is based only on direct intensity matching. Charpiat's method [4], which is based on SURF descriptors and multimodal probability distribution estimation, is more robust to differences in intensity. It can produce successful matches between identical or very similar objects in input images, as it can be seen on landscape images in rows (b) and (d). However, this method can fail when changes of visual features are more significant, as shown in examples (a) and (c). In contrast to the previously discussed methods, our approach and the approach by Gupta *et al.* [7] lead to comparable results with fewer artifacts and a more satisfactory colorization.

Additional comparisons with Gupta *et al.* [7] are shown in Figure 4 to assess the effectiveness of our descriptors. Despite the similarities in the input images, the descriptors used in [7] cannot accurately determine correspondences for ambiguous areas, such as the clouds seen in row (a), where our method preserves the white of the clouds. Similarly, our method is robust to global changes as shown in row (b). In this case the input images vary significantly in global luminance and contrast, even though they depict the same structures. Our texture and gradient based descriptors successfully determine correspondences in this case. Similar observations can be made about the last example (c), where the same objects are depicted but with changes in pose.

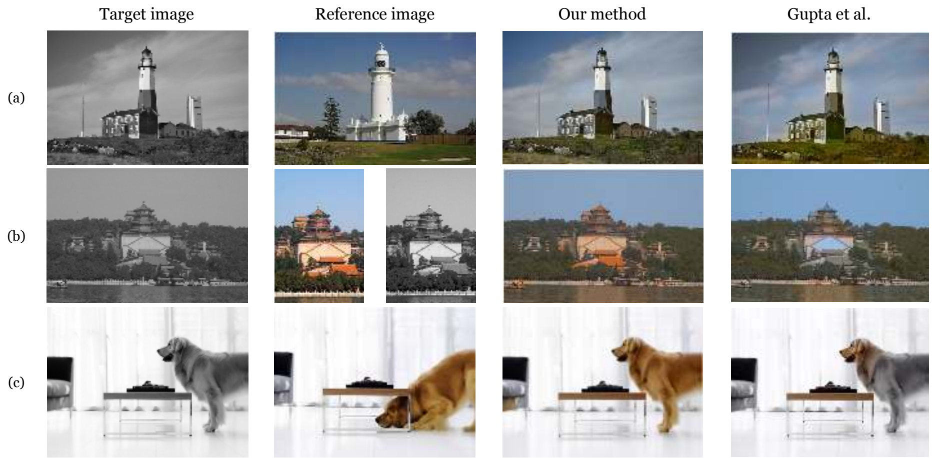

Fig. 4: Additional comparisons with method by Gupta *et al.* [7]. The intensity of the reference image is also shown in example (b). Our method preserves the white of the clouds (a) and can successfully colorize images despite global luminance and contrast changes (b) or changes in pose (c).

Our method achieves qualitative improvements and more robust performance over the state of the art for a large selection of images, while offering a 2 to 4-fold computational improvement against the method by Gupta et al. [7].

6 Color Artifact Removal by Regularization

Application of the proposed regularization method is not limited to colorization. Our method can also be applied for artifact suppression in a more general scenario of color manipulations, e.g. color transfer. In Figure 5, an image (a) is recolored according to the reference image (b), using the color transfer method by Pitié *et al.* [14]. In this case, the result (c) shows distinctive color artifacts, which can be regularized with our approach requiring only minimal adjustments. To regularize artifacts in this case, the original color image is considered for the superpixel segmentation process, while the recolored result serves as input for the computation of strokes. As the original image here contains color information, all three channels are used for the superpixel segmentation and clustering, leading to more accurate results. Given the additional color information, we found that a larger number of superpixels could be used in this case: we set $n = 750$, allowing the segmentation to follow image edges more closely. At the same time, a higher threshold was used for DBSCAN ($T = 5$ in this case), allowing for smaller clusters to form, therefore better respecting image detail.

After the segmentation, color strokes are automatically created and propagated as described in Section 4. Note that the colors of strokes are extracted from the result of color transfer, but the strokes are marked onto the luminance channel of the original artifact-free image, to allow for smoother propagation. Figure 5 shows a comparison of our regularized result as well as those of competing methods [14,15]. In contrast to other solutions, we only regularize in the chromatic domain, leaving luminance information unchanged.

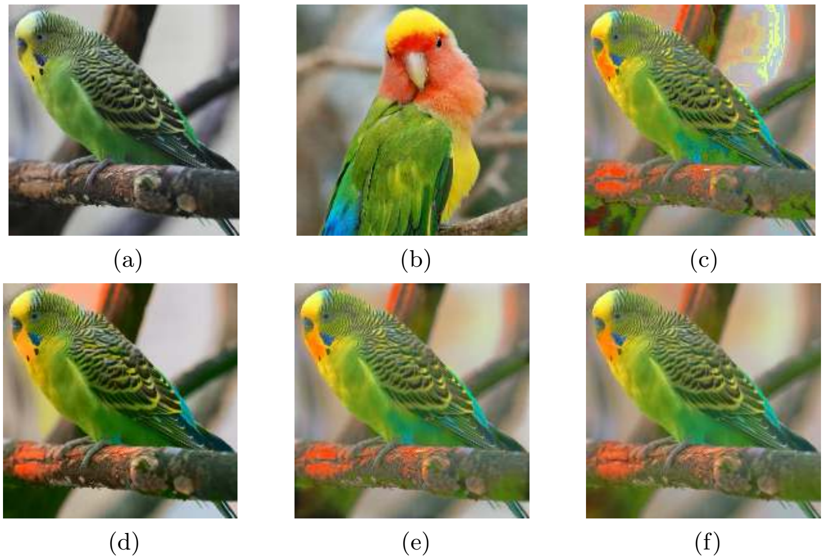

Fig. 5: Regularization of color transfer artifacts. (a) Original image. (b) Reference image. (c) Color transfer by Pitié *et al.* [14]. Regularization using (d) our method, (e) regraining method [14] and (f) TMR filter [15].

7 Conclusions

We presented a new, automatic method for colorizing grayscale images. Our method combines advantages of example-based and stroke-based techniques, requiring minimal levels of user interaction. Using descriptor representations of the input images, the matching step can provide meaningful correspondences between images, even under global contrast and illumination changes or differences in scene structure. In addition, we proposed a robust regularization scheme for reducing artifacts due to color manipulations, which we demonstrated in the context of colorization and color transfer. We applied our technique to several image pairs with varied content and compared our results to the state of the art, showing that the proposed colorization method is competitive with recent methods, providing robust colorization at a lower computational time.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(11), 2274–2282 (2012)
2. Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A.: The generalized patchmatch correspondence algorithm. In: ECCV 2010, pp. 29–43. Springer (2010)
3. Bugeau, A., Ta, V.T., Papadakis, N.: Variational exemplar-based image colorization. IEEE Transactions on Image Processing 33(0), 1–9 (2013)
4. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: Computer Vision–ECCV 2008, pp. 126–139. Springer (2008)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
7. Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: Proceedings of the 20^{th} ACM international conference on Multimedia. pp. 369–378. ACM (2012)
8. Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: Proceedings of the Sixteenth Eurographics conference on Rendering Techniques. pp. 201–210. Eurographics Association (2005)
9. Kong, T., Rosenfeld, A.: Topological Algorithms for Digital Image Processing. Machine Intelligence and Pattern Recognition, Elsevier Science (1996)
10. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. pp. 689–694. SIGGRAPH '04, ACM, New York, NY, USA (2004)
11. Lin, S., Ritchie, D., Fisher, M., Hanrahan, P.: Probabilistic color-by-numbers: suggesting pattern colorizations using factor graphs. ACM Trans. Graph. 32(4), 37 (2013)
12. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. International journal of computer vision 43(1), 7–27 (2001)
13. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(5), 530–549 (2004)
14. Pitié, F., Kokaram, A.C., Dahyot, R.: Automated colour grading using colour distribution transfer. Computer Vision and Image Understanding 107(1), 123–137 (2007)
15. Rabin, J., Delon, J., Gousseau, Y.: Removing artefacts from color and contrast modifications. Image Processing, IEEE Transactions on 20(11), 3073–3085 (2011)
16. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: ACM Transactions on Graphics (TOG). vol. 21, pp. 277–280. ACM (2002)