ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/262353941

Image colorization using similar images

Conference Paper · October 2012

DOI: 10.1145/2393347.2393402

CITATIONS
54

READS
243

5 authors, including:

Raj Kumar Gupta
Nanyang Technological University
19 PUBLICATIONS **272** CITATIONS

SEE PROFILE

Ee Sin Ng
Institute for Infocomm Research
8 PUBLICATIONS **76** CITATIONS

SEE PROFILE

Zhiyong Huang
National University of Singapore
96 PUBLICATIONS **631** CITATIONS

SEE PROFILE

All content following this page was uploaded by Raj Kumar Gupta on 27 April 2017.

The user has requested enhancement of the downloaded file.

Image Colorization Using Similar Images

Raj Kumar Gupta[1*] Alex Yong-Sang Chia[2*] Deepu Rajan[1] Ee Sin Ng[2] and Huang Zhiyong[2]

[1]School of Computer Engineering, Nanyang Technological University
50, Nanyang Avenue, Singapore – 639798

[2]Institute for Infocomm Research (I[2]R)
1 Fusionopolis Way, #21-01 Connexis, Singapore – 138632
{rajk0005, asdrajan}@ntu.edu.sg, {ysachia, nges, zyhuang}@i2r.a-star.edu.sg

ABSTRACT

We present a new example-based method to colorize a gray image. As input, the user needs only to supply a reference color image which is semantically similar to the target image. We extract features from these images at the resolution of superpixels, and exploit these features to guide the colorization process. Our use of a superpixel representation speeds up the colorization process. More importantly, it also empowers the colorizations to exhibit a much higher extent of spatial consistency in the colorization as compared to that using independent pixels. We adopt a fast cascade feature matching scheme to automatically find correspondences between superpixels of the reference and target images. Each correspondence is assigned a confidence based on the feature matching costs computed at different steps in the cascade, and high confidence correspondences are used to assign an initial set of chromatic values to the target superpixels. To further enforce the spatial coherence of these initial color assignments, we develop an image space voting framework which draws evidence from neighboring superpixels to identify and to correct invalid color assignments. Experimental results and user study on a broad range of images demonstrate that our method with a fixed set of parameters yields better colorization results as compared to existing methods.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications;

General Terms

Algorithms

*denotes equal contributions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MM'12, October 29–November 2, 2012, Nara, Japan.
Copyright 2012 ACM 978-1-4503-1089-5/12/10 ...$15.00.

Keywords

Automatic colorization, cascade feature matching

1. INTRODUCTION

The goal of image colorization is to add colors to a gray image such that the colorized image is perceptually meaningful and visually appealing. A key challenge of this problem is that it is under constrained since there are potentially many colors that can be assigned to the gray pixels of an input image (e.g. leaves may be colored in green, yellow and brown). Hence, there is no one correct solution to the colorization problem and human intervention often plays an important role in the colorization process.

In general, colorization methods can be broadly divided into two main classes: interactive colorization methods and automatic colorization methods. Interactive colorization techniques [9, 7, 18, 15, 13] require a user to manually mark color scribbles on the target image. Colors from these scribbles are then smoothly propagated across the entire image based on an optimization framework. A key weakness of such methods is that they demand extensive efforts from a user. Additionally, for such methods, colorization quality is strongly dependent on the user supplied color scribbles, and it is often difficult for a novice user to provide sufficiently good color scribbles to achieve desirable colorization results. Automatic colorization methods [17, 6, 2, 12] take a different approach to image colorization. Specifically, rather than obtaining chromatic values from the user, these methods take a reference color image as input, and transfer colors from the reference image to the target image. While these methods can reduce the extent of user effort, in many cases, these methods often require careful tuning of a large number of parameters to yield satisfactory results.

In this paper, we propose a new automatic colorization method which exploits multiple image features to transfer the color information from reference color image to input gray image. Specifically, other than intensity and standard deviation features which are used by other colorization methods e.g. [17, 8], we incorporate the highly discriminative SURF and Gabor features in our method. The power of discriminative image features to reliably find correspondence between images has been demonstrated by Liu *et al.* [11] where they exploit SIFT features to find correspondences between images for scene alignment. The use of information rich Gabor and SURF features empowers our method to reliably find correspondences between the reference and target images for color transfer. Here, we choose Gabor feature for its effective representation of texture and popularity in the computer vision domain, while SURF feature is chosen for its excellent discriminative ability and compactness. We find correspondences at the resolutions of superpixels. This affords our method to exhibit a stronger level of spatial coherency than that possible with independent pixels. To support fast color transfer, we used a fast cascade feature matching scheme to quickly find correspondences between reference and target superpixels. At the first cascade step, we identify for each target superpixel a set of reference superpixels which are most similar to the target based on a particular feature type. This set of reference superpixels are screened at subsequent steps of the cascade using different feature types to sieve out matching reference superpixels. An initial set of color values are then assigned to target superpixels based on its matching reference superpixels. We note that these matchings are found based solely on image features and thus could be unreliable at image regions where features cannot be reliably extracted [16] (e.g. at regions corresponding to object boundaries). To improve the colorization results, we further enforce spatial consistency in the colorization by exploiting an image space voting framework which draws evidence from neighboring superpixels to quickly identify and correct invalid matchings. This leads to improved colorization results.

We evaluate our method on a diverse range of images comprising portrait, painting, landscape as well as on images containing deformable and rigid foreground objects. Experimental results demonstrate that our method, while straightforward, is sufficiently powerful to yield perceptually meaningful and visually appealing colorizations on these complex images. Additionally, comparison against existing state-of-the-art methods also demonstrates our method to be more effective at colorization, even with a fixed set of parameters.

2. RELATED WORK

Adding realistic colors to a gray image can improve the photorealism of the image, and has attracted much attention in the research community. Levin *et al.* [9] proposed a simple yet effective colorization algorithm that requires a user to provide color scribbles at various image regions. These color scribbles are then propagated automatically to the entire image by a least-squares optimization method. Huang *et al.* [7] improved on this method to reduce color blending at image edges. Yatziv and Sapiro [18] used multiple scribbles to colorize a pixel, where the combination weights are computed by a distance measure between the pixel and the scribble. While these methods have been shown to achieve good colorization results, a main shortcoming of these previous approaches is that they require a large number of color scribbles on the gray image as input. To reduce the number of color scribbles, Qu *et al.* [15] and Luan *et al.* [13] propagated scribbles that are marked on an image patch to other patches that have similar texture features. While their methods reduce the number of required scribbles, like all other interactive colorization methods, they demand substantial artistic skills of the user to mark appropriate colors on image patches to yield desirable colorizations.

Rather than obtaining colors directly from a user, Welsh *et al.* [17] obtained colors from a user-supplied reference color image. They extracted small image patches at each pixel of the target image and matched these patches to those of the reference image. These matches are then used to directly transfer colors from the reference image to the target image. Their method requires the user to manually mark corresponding regions between the reference and target to yield satisfactory color transfer, and is considerably less intuitive than our method which only needs a user-supplied reference image. Additionally, their method completely ignores spatial information of the pixels and hence their colorizations often yield very weak spatial consistency.

To address the spatial coherency problem, Irony *et al.* [8] proposed a method which used a segmented reference color image as an additional input. Here, they exploit color and texture information from segmented regions of the reference image to automatically segment the target gray image into a set of locally homogeneous image patches. Colors from the segmented regions of the reference image are then transferred to the image patches of the target image as color scribbles. Colors are propagated across the entire image by [9]. Their method demands segmentation masks as input, and is unsuitable for colorizing complex images e.g. foreground objects with fine scale structures. Charpiat *et al.* [2] assign colors to a gray image by minimizing an energy function using the graph cut approach. While their algorithm does not requires any user intervention during the colorization process, colorization results by their method are heavily influenced by the choice of many parameters.

Liu *et al.* [12] proposed an example-based colorization technique which is robust to illumination variations in the target and reference images. Their method downloads multiple reference images from the internet based on user-supplied keywords, and computes precise per-pixel registration between the reference and target images for color transfer. A key weakness of their method is that it is limited to colorizing foreground objects for which an exact replica of the target gray foreground object can be found. Consequently, their method is limited to rigid objects (e.g. landmarks) and is unsuitable for colorizing images which contain deformable objects such as those shown in this paper.

Recently, Chia *et al.* [3] developed a method which also exploits internet images for colorization. Relevant internet images are found with a novel image filtering framework, and colors are transferred from the reference to target with the belief-propagation framework. Very good colorization results are achieved by their method, though their method requires manual segmentation of major foreground objects of the target gray image.

In all previous works that we have discussed above in this section, Irony *et al.* [8] method is more similar to the proposed algorithm which also uses nearest-neighbor search and image space voting. However, our algorithm differs from their work in the following aspects:

- Irony's method requires manually segmented reference color image as an additional input. This segmented image is used to construct a feature space and a corresponding classifier, and hence affects the colorization result significantly. To minimize user intervention, the use of automatic image segmentation, which often generates small image segments in case of dense textured image regions, leads to poor colorization.

- To enforce the spatial consistency, Irony performs image space voting in feature space followed by a global

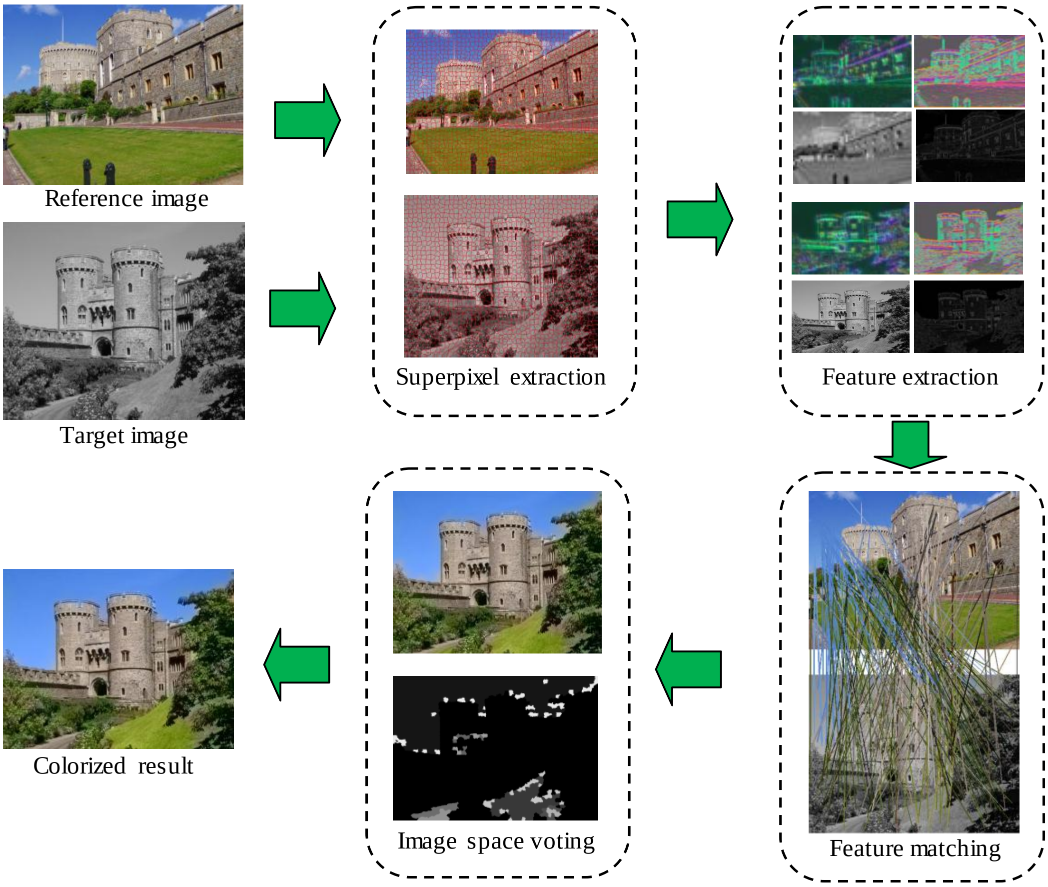


Figure 1: Overview of our colorization method. We work at the level of superpixels to extract different types of image features from the reference and target images, and find correspondences between features by a fast cascade feature matching scheme. These correspondences provide cues to the initial set of color assignments, which is refined by an image voting step to yield the final colorized result.

optimization. Their voting scheme identifies similar image regions in gray image with the same texture features as specified by the input reference color image segmentation mask. In our algorithm, the use of superpixels helps us to achieve spatial consistency in small image regions. After computing the initial color values for each of these superpixels, we group these superpixels and perform the image space voting in color space to update their previously assigned color values. Our primary focus is to enforce uniform color values to all connected pixels that are the part of same image segment. It enables us to achieve higher spatial consistency and better colorization results.

- Irony used the Discrete Cosine Transform (DCT) coefficients of a fix block size as a texture descriptor. While DCT coefficients are not sensitive to translations and rotations, they are more sensitive to scale changes. Here, we use a rich set of image features, including features computed at different scales, to find appropriate matches in reference image to transfer the color information.

3. COLORIZATION METHOD

The proposed algorithm colorizes gray input images by a user-supplied reference color image. We do not restrict the reference image to contain identical object instances as the input image. Instead, our only constraint is that the reference images should be semantically similar to the input image i.e. the reference image needs only to contain similar scene object types as the input image (e.g. castle). An overview of our method is given in Figure 1. As shown, our method comprises four key stages: (a) superpixel extraction (b) feature extraction, (c) feature matching, and (d) image space voting. We describe each of these steps below.

3.1 Superpixel extraction

We extract features from the color reference and target gray images at the resolution of superpixels, and transfer colors between superpixels to yield image colorization. An advantage of using a superpixel based representation is that it speeds up the colorization method. More importantly, it also affords our method with an ability to maintain stronger spatial coherency in the colorization as compared to that using individual pixels. To compute the superpixels, we used a geometric-flow based algorithm proposed by [10]. The algorithm computes the compact superpixels with uniform size and shape and preserves original image edges.

For all experiments presented in this paper, the input *time step* value and the *maximum number of iterations* are taken as 0.5 and 500, respectively. These values are the default parameter values provided by the authors along with their source code[1]. Depending on the image size, the input *number of superpixels* are chosen to keep an average superpixel size of around 40 pixels.

3.2 Feature extraction

For each superpixel in input gray image and reference color image, we compute 172−dimensional $(2+2+40+128)$ feature vector based on their intensity, standard deviation, gabor features and SURF descriptors. To compute this feature vector for a superpixel, we compute a 172−dimensional feature vector at each image pixel and then compute the mean value of all feature vectors that belong to the pixels within a superpixel to represent that superpixel. We compute this feature vector as follows:

Intensity features A two-dimensional feature vector is computed for each superpixel based on the intensity values. The first dimension is the average intensity values of all pixels within the superpixel S,

$$f_1(i) = \frac{1}{n} \sum_{(x,y)\in i} I_{(x,y)} \tag{1}$$

where $I_{(x,y)}$ is the intensity of pixel (x, y) and n is the total number of pixels within the superpixel S_i. The second dimension is computed as the average intensity values of the neighboring superpixels of S_i,

$$f_2(i) = \frac{1}{N} \sum_{j\in\eta} f_1(j), \tag{2}$$

where η represents the neighboring superpixels of S_i and N is the number of neighboring superpixels.

Standard deviation features Similar to intensity, we also compute a two-dimensional feature based on the standard deviation values in small pixels neighborhoods around each image pixel. For all experiments in this paper, we used a 5×5 square window to compute the standard deviation value at each image pixel. The standard deviation feature for the superpixel is then computed in the same way as that computed for the intensity feature.

Gabor features We apply Gabor filters [14] to an image with eight orientations varying in increments of $\pi/8$ from 0 to $7\pi/8$, and with five exponential scales $\exp(i \times \pi)$, $i = 0, 1, 2, 3, 4$ to compute a 40-dimensional feature at each pixel. The Gabor feature for the superpixel is then computed as the average Gabor feature of all pixels within the superpixel.

Speeded Up Robust features Similar to the Gabor features, we also extract a 128-dimensional extended SURF descriptors [1] at each image pixel. Extended SURF descriptors for each superpixel in then computed in the same way as the computation of the Gabor features.

3.3 Cascade feature matching scheme for initial color assignment

We exploit the features which are extracted in the previous section to find correspondences between the reference and target superpixels, and harness these correspondences to assign a set of initial colors to the target superpixels. Here, for each target superpixel, one can search among *all* reference superpixels across *all* feature types to find the reference superpixel which is most similar to the target. This however demands large processing time. For greater efficiency, we instead employ a fast cascade feature matching scheme which continually prunes the search space at each step of the cascade and concentrates the search only on reference superpixels which are sufficiently similar to the target. To ensure that the search space are pruned reliably, we exploit the more discriminative Gabor and SURF features at the initial cascade steps to sieve out a set of matching reference superpixels for a target superpixel, before relying on the intensity and standard deviation features to find its final matching reference superpixel. In our work, we found feature matches to be largely unaffected by using SURF before/after Gabor and intensity before/after standard deviation.

Let $\{r_i\}$ denote the set of reference superpixels which are extracted from the reference image I_r. Consider a target superpixel t_i. Starting at the first cascade step, we find a set of α reference superpixels from $\{r_i\}$ which are most similar to t_i based on the Gabor features. Let this set of α reference superpixels be denoted as Φ_i. We compute distance between two features of the same type by the Euclidean distance measure. Following that, at the second level, we find $\frac{\alpha}{2}$ reference superpixels from Φ_i which is most similar to the currently considered target superpixel t_i based on the SURF features. Intensity and standard deviation features are then used in the third and fourth levels respectively to find the set of reference superpixels which are most similar to t_i. Let Υ_i denote the set of reference superpixels found by the cascade filtering process to be most similar to t_i at the final step of the cascade. The reference superpixel r_a within Υ_i which correspond to t_i is then identified as one with the least matching cost across different feature types to t_i,

$$a = \arg\min_b \; F(r_b, t_i), \quad r_b \in \Upsilon_i, \tag{3}$$

and

$$\begin{aligned} F(r_b, t_i) &= w_1 C_1(r_b, t_i) + w_2 C_2(r_b, t_i) \\ &+ w_3 C_3(r_b, t_i) + w_4 C_4(r_b, t_i). \end{aligned} \tag{4}$$

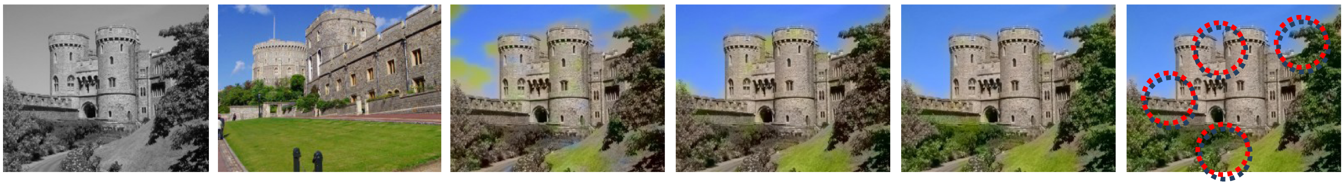
We denote C_1, C_2, C_3 and C_4 as the Euclidean distance between the Gabor, SURF, intensity and standard deviations features, and w as their accompanying weights. For all experiments in this paper, we fixed w_1, w_2, w_3 and w_4 to be 0.2, 0.5, 0.2 and 0.1 respectively.

We use the CIELab color space to transfer the color from reference superpixel to target superpixel. While transferring the color, we only transfer the chromatic color values a and b of the reference superpixel as micro-scribbles to the center of its matching target superpixel. These color scribbles are spread across all the pixels with an optimization-based color interpolation algorithm [9]. The algorithm is based on the principle that neighboring pixels with similar luminance should also have similar colors. The algorithm attempts to minimize the difference $J(C)$ between the color assigned to a pixel p and the weighted average of the colors assigned to its neighbors,

$$J(C) = \sum_{p\in l} \left(C(p) - \sum_{q\in N(p)} w_{pq}\; C(q) \right)^2, \tag{5}$$

where the weights w_{pq} are determined by the similarity of

[1]http://www.cs.toronto.edu/ babalex/research.html

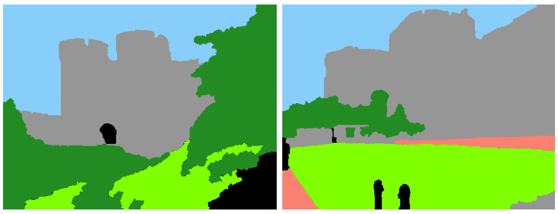


(a) (b) (c) (d) (e) (f)

Figure 2: Colorization results obtained at different steps of cascade feature matching. We assign chromatic values ab of the reference superpixels as micro-scribbles to the center of its matching target superpixels, and propagate the colors across the entire image by [9]. Target gray and reference color images are shown in (a) and (b) respectively. Colorizations obtained at the first (Gabor only), second (Gabor + SURF only), third (Gabor + SURF + intensity only) and fourth (Gabor + SURF + intensity + standard deviation) steps of cascade are shown respectively in (c) to (f). Circled regions in (f) depict visually invalid color assignments at the final step of the cascade, which are automatically identified and corrected by image space voting as detailed in Section 3.4. Best viewed on screen.

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

Figure 3: Colorization results obtained by a single feature type of (a) Gabor, (b) SURF, (c) intensity and (d) standard deviation. Poor colorizations are obtained using a feature type independently, since no one feature type can model difference image regions sufficiently well.



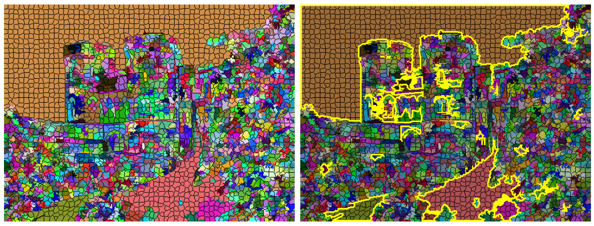
(a) (b)

Figure 4: Manually labeled regions on (a) reference and (b) target images. Regions with similar labels are depicted by similar colors.

their luminance (Y),

$$w_{pq} \propto e^{-(Y(p)-Y(q))^2/2\sigma_p^2}. \tag{6}$$

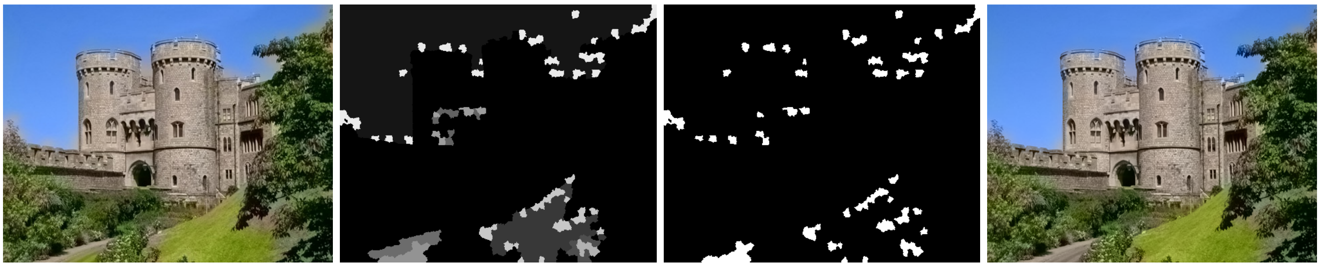
Figure 2 shows intermediate colorization results obtained at each cascade step. For comparison, we show in Figure 3 colorization results that are obtained by using each feature type independently. It can be seen that colorizations are poor with a single feature type (Figure 3). This is due to the inability of a single feature type to correctly find correspondences across different regions of an image. On the other hand, by collectively exploiting different feature types, we show the synergy of different feature types can markedly improve colorization quality (Figure 2 (f)). To illustrate the improvement following each cascade step, we manually label image regions (i.e. sky, castle etc.) in both the target gray and reference images of Figures 2 (a) and (b) as shown in Figures 4 (a) and (b) respectively. These labels are used to quantify the extent of matching errors at each cascade step, where the matching error is computed based on the target gray image superpixels that have the same labels as its matching reference image superpixels.



(a) (b)

Figure 5: Extracted superpixels and image segments are shown overlaid on target image in (a) and (b) respectively. Image segments which participate in the image space voting are highlighted within yellow border in (b).

Without using the cascade feature matching scheme, the matching errors obtained with Gabor, SURF, intensity and standard deviation features independently are 48.7%, 37.9%, 33.1% and 46.1%, respectively. The matching errors reduce significantly with the proposed cascade feature matching scheme. Specifically, the matching errors at the first, second, third and fourth cascade steps are 48.7% (Gabor), 34.8% (Gabor + SURF), 27.1% (Gabor + SURF + Intensity) and 16.5% (Gabor + SURF + Intensity + Standard deviation), respectively. This error can be further reduced using image space voting, as discussed in the following section.



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

Figure 6: Comparison of colorization results following the color reassignment step. (a) Colorization from the feature matching step. (b) Confidence of correct color assignment, where brighter superpixels indicate weaker confidence for its color assignment. (c) Superpixels (indicated in white) whose colors are reassigned. (d) Final colorization following color reassignment.

3.4 Image space voting for color reassignment

The correspondences found by the above matching step assign colors to superpixels based solely on image features. While our use of multiple feature types improves color assignments significantly as compared to that using a single feature type, there could be some visually invalid assignments due to incorrect correspondences found (circled regions in Figure 2(f)). To improve the color assignments, we enforce spatial consistency in the colorization by explicitly voting for the color assignments in the image space. Here, our basic intuition is that color assignment for a superpixel is likely to be correct if its neighboring superpixels which have similar image properties are also assigned similar colors. Consequently, we can exploit neighboring superpixels to identify and to correct invalid color assignments.

Let I be a target image, and $\{s_i\}$ be the set of image segments. To extract these image segments, we use the mean-shift algorithm proposed by Comaniciu *et al.* [4]. We use their source code[2] with the default input parameters *SpatialBandwidth* and *RangeBandwidth* as 2 and 3, respectively. After computing the image segments, we keep only such image segments for voting that contain at least three superpixels in it as shown in Figure 5, where the selected segments have been highlighted using yellow border. Intuitively, each image segment s_i is a grouping of connected superpixels which have similar image properties. For each image segment s_i, we cluster its corresponding superpixels based on their initial a and b chromatic color values (which are obtained from Section 3.3) with k-means clustering. Densely populated clusters provide strong evidence for the correct color assignments of its member superpixels, while superpixels from sparsely populated clusters indicate that such superpixels have little support for its color assignments. In this regard, the clustering procedure identifies invalid color assignments by pooling evidence from its neighboring superpixels together, where the confidence of a color assignment for a target superpixel is computed as the number of member superpixels belonging to the same cluster as the target superpixel. Here, we identify sparsely populated clusters as those which have less than $\frac{1}{2k}$ the number of superpixels in segment s_i. We reassign colors to superpixels from sparsely populated clusters by the average color values of the superpixels from the most populated cluster. For all experiments in this paper, we set k to be 2.

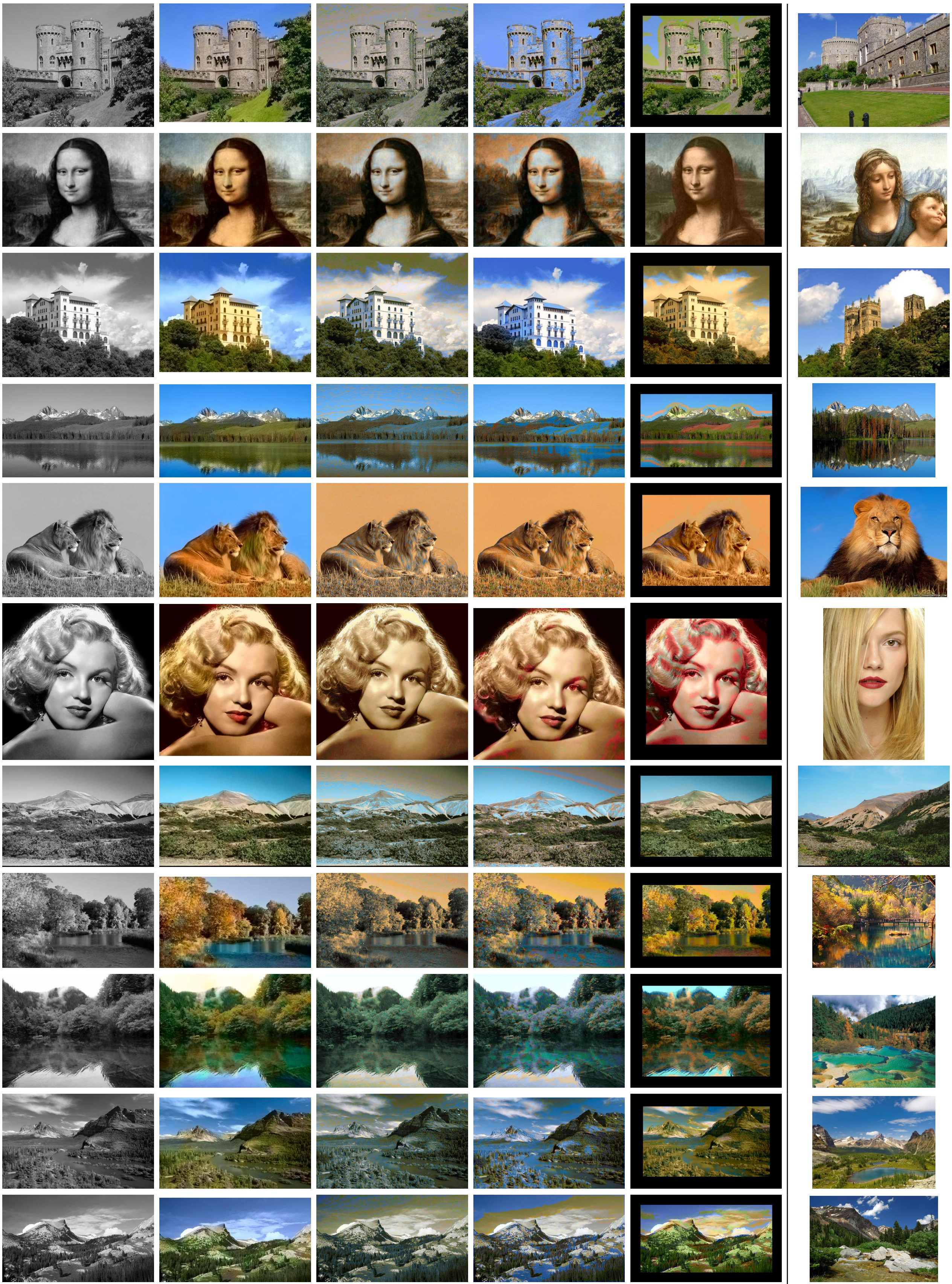
[2] http://coewww.rutgers.edu/riul/research/code/EDISON/

We show colorization obtained directly with the cascade feature matching scheme on an example image in Figure 6(a). The confidence of its color assignments that are obtained with the feature matching step is visualized in Figure 6(b), where brighter patches indicate weaker confidence for correct assignment. It is seen that most superpixels identified to have low confidences for their correct color assignments are located at image regions where there are sharp changes in image properties (e.g. along castle-sky and field-leaves boundaries). This is not surprising since patch based features (as used here) are often affected by changes in intensity and textures, and hence features extracted from such superpixels are often less reliably matched as those located on a more homogeneous image patches. We depict superpixels whose colors are reassigned as the white patches in Figure 6(c), and the resulting colorized results in Figure 6(d). It can be seen that visually invalid colors, such as those from the grass and sky segments, have been reassigned to yield a perceptually more appealing colorization where the matching errors are further reduced from 16.5% to 9.24%. These errors are concentrated in image segments with very few superpixels, which reduces the robustness of the image voting step. This is further discussed in Section 4.2.

4. EXPERIMENTS

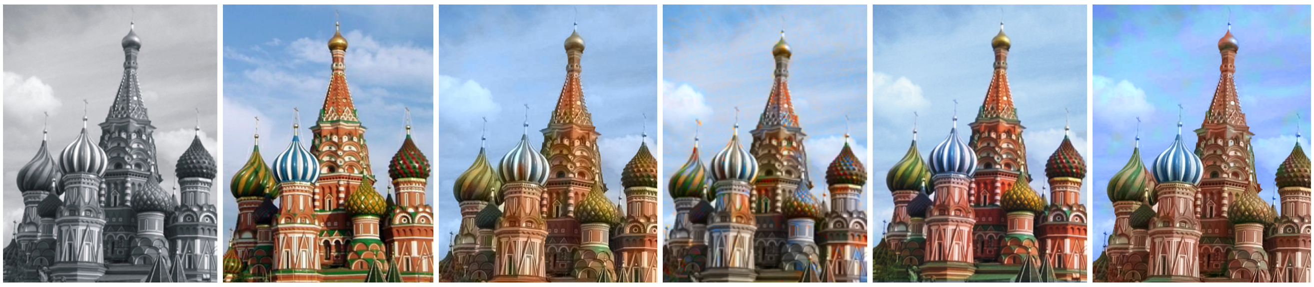
We present colorization results on a diverse range of images, and compare them to those obtained with existing state-of-the-art colorization methods. For all experiments, the following same fixed parameters are used for our method. We normalize the reference and target images to have a diagonal length of 500 pixels, and extract around 3000 superpixels [10] from an image using the input parameters mentioned in Section 3.1. The cascade feature matching step finds correspondences between reference and target superpixels with α equal to 600, and weights w_1, w_2, w_3 and w_4 equal to 0.2, 0.5, 0.2 and 0.1 respectively. Colors reassignment are obtained with k fixed as 2. For the comparison methods, we used the default parameters which are supplied by the authors.

Figure 7 shows colorization results of our method, with comparisons to existing state-of-the-art methods [17, 8, 2] which have similar extent of user interaction as our method. The algorithm [8] requires the segmentation mask of the reference color image as an additional input. We compute this mask by using automatic image segmentation with same input parameters as we used in Section 3.4. It is seen that our



Input image | Our method | Welsh *et al.* [17] | Irony *et al.* [8] | Charpiat *et al.* [2] | Reference image

Figure 7: Comparison with existing state-of-the-art colorization methods which have similar extent of user interaction as our method. Last column shows reference color images that are used by all algorithms for colorizing the input images. The input segmentation masks used by Irony *et al.* [8] are computed with the method discussed in Section 3.4.



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

Figure 8: Comparison against Liu *et al.* [12] and Chia *et al.* [3] which have additional constraints. (a) and (b) show the input gray image and reference color image used for color transfer. (c) shows our colorization result. (d) shows the colorization results obtained with Liu *et al.* [12] which requires reference image to have the exact object instances as the gray image. (e) shows colorization results obtained by Chia *et al.* [3] which requires segmentation masks. For comparison, results obtained with Irony *et al.* [8], which is most similar to our method, are included in (f).

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

Figure 10: Colorization using multiple exemplars by our method. (a) Input gray image. (b,c) Reference color images. (d) Colorization obtained using both reference images.

method yields perceptually more appealing colorization results than other methods. Our method is seen to work well on these complex images which exhibit variations in intensity and texture, even with the same fixed set of parameters.

Figure 8 compares our method against other colorization methods which have additional input constraints. From the figure, we can see that the algorithms [12] (which requires exact object instances to be present in both reference and gray images) and [3] (which requires segmentation masks of foreground objects) are able to transfer the exact color values at few pixel locations due to their use of spatial positions during the color transfer. The use of spatial position restricts the flexibility of these algorithms and works well only if the reference images used to transfer the color have been taken from the same viewing angle. Although, the proposed algorithm does not use such constraint, it is still able to achieve comparable colorization result and clearly outperforms Irony's method [8], which is most similar to the proposed algorithm.

In Figure 9, we demonstrate colorization with user-supplied keywords (rather than with user-supplied color images). Here, given a target gray image and a keyword, we automatically find semantically relevant reference color images for colorization. Specifically, based on the user-supplied keyword, we download 2000 images from photo sharing websites such as Flickr and Google Image Search. We evaluate the colorfulness of the downloaded images by using the method proposed by Hasler *et al.* [5], and discard those images whose

Figure 11: Examples of images displayed for quantitative evaluation during the user study. The second image is artificially colored with our colorization method, while the others are original color images.

colorfulness scores are below the recommended thresholds suggested in [5]. From among the remaining images, we used near duplicate key frame search method [19] to identify the internet image which is the most similar to the input gray image. Figure 9 shows colorization results obtained by user-supplied keywords, where the keywords are shown at the top of each reference color image. It is seen that colorization results are weak with coarse keyword (first colorized results with the 'lighthouse' keyword). This is not surprising since the retained internet image does not provide sufficient semantically similar image and color information to correctly colorize the input image. On the other hand, when more specific keywords are used (remaining results), colorization results improved markedly. Note that the first two rows of Figure 9 depict colorizations on the same grayscale image using different reference images.

Extension of our method to colorize gray image using multiple exemplars are depicted in Figure 10. Here, we extract superpixels from multiple exemplars and match the target superpixels with them in the cascade feature matching step. Figure 10(a) shows a grayscale image, while reference images used for its colorization are shown in Figures 10(b) and (c). We show the colorization result in Figure 10(d).

4.1 User study

We performed a user study to quantitatively compare our colorization method with the state-of-the-art methods. Here, we use the exact framework and the same set of test images as those used by Chia *et al.* [3] for the evaluation. We engage 30 volunteers and present to them a set of images. Some of these images are artificially colorized, while the rest are real color images. We show each subject a set of four different images at a time (such as in Figure 11) for a total of 30

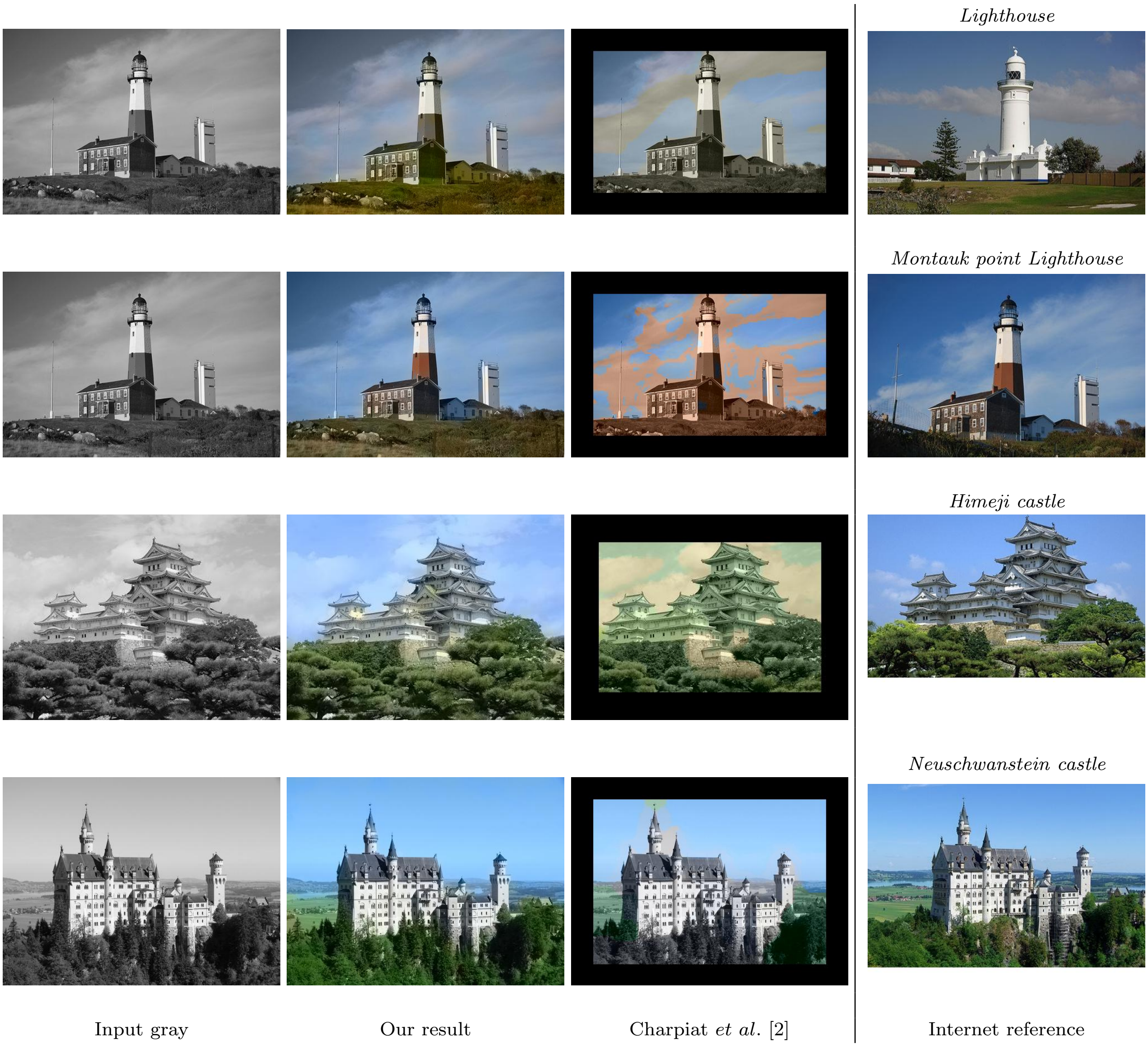


Figure 9: Colorization with user-supplied keywords. The first column shows the input gray images. The second and third columns show the colorization results obtained by our method and Charpiat *et al.* [2], respectively. Last column shows the reference color images, along with the keywords (shown at the top of each image) which used to download the reference images from the internet.

sets, and ask the subjects to identify all artificially colorized images in each set. Each subject was given five seconds to view an image set. Prior to the experiment, the subjects were told that at most two images from each set were artificially colorized. Using our method, the subjects classified colorized images obtained by our method as real 64.9% of the time. This compares favorably to 48.90% obtained by [17], 53.2% by [8], and 32.30% by [2]. While it attains slightly weaker colorization results as compared to Chia *et al.* [3] (which obtains 66.59%), we note that [3] directly exploits spatial information in their colorization and hence require users to manually segment major foreground objects from the input image. In contrast, our method only require users to present a semantically similar color image as input, which is substantially more intuitive.

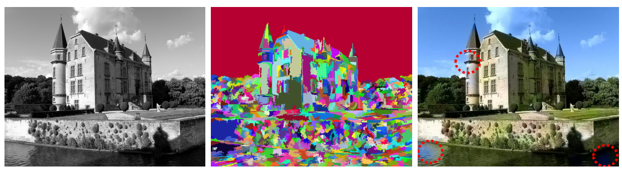


Figure 12: Limitations of automatic image colorization. (a)-(c) show the input gray image, automatic segmented input image and the colorization result, respectively. Circled regions in (c) depict some artifacts generated due to small image segments.

4.2 Limitations

There are a few limitations in our method. First, our use of superpixel representation, while supporting more spatial coherency in colorization, can be inaccurate at object boundaries or thin image structures. This could potentially lead to bleeding artifacts at object boundaries. Second, image segments generated in Section 3.4 are often very small in dense textured regions. This reduces the robustness of the image voting step since these segments have fewer superpixels within them than larger segments. Consequently, voting for the colors within these segments by its superpixels become less reliable. This is shown in Figure 12, where Figures 12(a) and (b) show an input gray image and segmented mask used for image space voting respectively, and colorization artifacts due to small segments are shown within circled regions in Figure 12(c). Finally, our method relies on the availability of color exemplar which is semantically similar to the gray image. Consequently, our method may fail when suitable color exemplars are unavailable.

5. CONCLUSIONS

In this paper, we present a new method for colorizing gray images using semantically similar reference images. Our method works at the resolution of superpixels, in which we extract a variety of features from the reference and target images. Correspondences between superpixels are found by a fast cascade feature matching scheme which examines different feature types at each cascade step. Leveraging on spatial information, we identify and correct invalid correspondences through an automatic image space voting paradigm. This, coupled with our superpixel representation, empowers our method to attain a strong extent of spatial consistency in the colorization. Experimental results on a wide array of images demonstrate our method achieves perceptually appealing colorizations, even with a fixed set of parameter settings. Additionally, comparisons against existing state-of-the-art methods also demonstrate our method to be more effective at colorization.

As future work, we would like to employ more features for finding correspondences between reference and target images. In particular, our current framework employs patch based features only, and we will explore how these features can be combined with contour fragment features. Additionally, we would also want to explore more advanced techniques to fuse various feature types to improve their overall discriminative potential. Finally, we have shown the potential of our method to colorize an image using internet images downloaded with user-supplied keywords. In the future, we also plan to study how we can identify semantically more relevant internet images, perhaps by incorporating object recognition abilities into the image filtering framework.

6. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

[2] G. Charpiat, M. Hofmann, and B. Scholkopf. Automatic image colorization via multimodal predictions. In *Proc. European Conference on Computer Vision*, 2008.

[3] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. *ACM Transactions on Graphics*, 30(6), 2011.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(05):603–619, 2002.

[5] D. Hasler and S. Séžűstrunk. Measuring colourfulness in natural images. *SPIE Electronic Imaging*, 5007:87–95, 2003.

[6] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. *Proc. SIGGRAPH*, pages 327–340, 2001.

[7] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *ACM Multimedia*, pages 351–354, 2005.

[8] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Proc. Eurographics Symposium on Rendering*, 2005.

[9] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694, 2004.

[10] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

[11] C. Liu, J. Yuen, and A. Torralba. Sift flow: dense correspondence across different scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5), 2011.

[12] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng. Intrinsic colorization. *ACM Transactions on Graphics*, 27(5):152:1–152:9, 2008.

[13] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proc. Eurographics Symposium on Rendering*, 2007.

[14] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[15] Y. Qu, T.-T. Wong, and P.-A. Heng. Manga colorization. *ACM Transactions on Graphics*, 25(3):1214–1220, 2006.

[16] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, 2008.

[17] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. *ACM Transactions on Graphics*, 21(3):277–280, 2002.

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

[19] J. Zhu, S. C. H. Hoi, M. R. Lyu, and S. Yan. Near-duplicate keyframe retrieval by semi-supervised learning and nonrigid image matching. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 7(1), 2011.

View publication stats