Example-based Image Colorization via Automatic Feature Selection and Fusion

Bo Li

School of Mathematics and Information Sciences, Nanchang Hangkong University, Nanchang, China.

Yu-Kun Lai, Paul L. Rosin

School of Computer Science and Informatics, Cardiff University, UK

Abstract

Image colorization is an important and difficult problem in image processing with various applications including image stylization and heritage restoration. Most existing image colorization methods utilize feature matching between the reference color image and the target grayscale image. The effectiveness of features is often significantly affected by the characteristics of the local image region. Traditional methods usually combine multiple features to improve the matching performance. However, the same set of features is still applied to the whole images. In this paper, based on the observation that local regions have different characteristics and hence different features may work more effectively, we propose a novel image colorization method using automatic feature selection with the results fused via a Markov Random Field (MRF) model for improved consistency. More specifically, the proposed algorithm automatically automatically classifies image regions as either uniform or non-uniform, and selects a suitable feature vector for each local patch of the target image to determine the colorization results. For this purpose, a descriptor based on luminance deviation is used to estimate the probability of each patch being uniform or non-uniform, and the same descriptor is also used for calculating the label cost of the MRF model to determine which feature vector should be selected for each patch. In addition, the similarity between the luminance of the neighborhood is used as the smoothness cost for the MRF

Email addresses: libo@nchu.edu.cn (Bo Li), Yukun.Lai@cs.cardiff.ac.uk, Paul.Rosin@cs.cf.ac.uk (Yu-Kun Lai, Paul L. Rosin)

model which enhances the local consistency of the colorization results. Experimental results on a variety of images show that our method outperforms several state-of-the-art algorithms, both visually and quantitatively using standard measures and a user study.
Keywords: image colorization, automatic feature selection, Markov random field, Bayesian inference

---

1. Introduction

The aim of example-based image colorization is to transfer the chrominance information from a reference image with color to a target grayscale image. It is an important research topic in image processing, and has many applications in different areas, such as heritage restoration [1] and image stylization [2, 3]. However, it is ill-posed and difficult because the common grayscale information between the reference and target images may not be sufficiently distinctive for reliable transfer. Most existing image colorization methods use feature matching: given a reference image with color information, the target grayscale image will be colorized by finding correspondences from the reference image based on feature similarity. Therefore, choosing suitable features is key to the colorization performance. In the pioneering work by Welsh et al. [4], luminance features are used to find the correspondences. However, such features perform poorly for non-uniform (e.g. textured) regions, leading to artifacts in the colorized images. More recent work has used many advanced texture features for image colorization, such as Gabor wavelets [5], SIFT [6], SURF [7], etc. To improve results, most existing methods use multiple features as a combined vector for matching, which implies that individual features contribute *equally* to region matching across the entire image. However, a specific type of feature is often more effective for certain types of regions. It is thus beneficial to treat regions *differently* according to their local characteristics. For example, pixels in uniform regions are more suitable to be matched by intensity features whereas texture descriptors should be used for highly non-uniform regions. An example is shown in Fig. 1. We can see that the intensity feature is suitable for the sky region but not the castle (Fig. 1(f)), whereas the texture descriptor performs well for the non-uniform castle regions but produces erroneous matches in the uniform

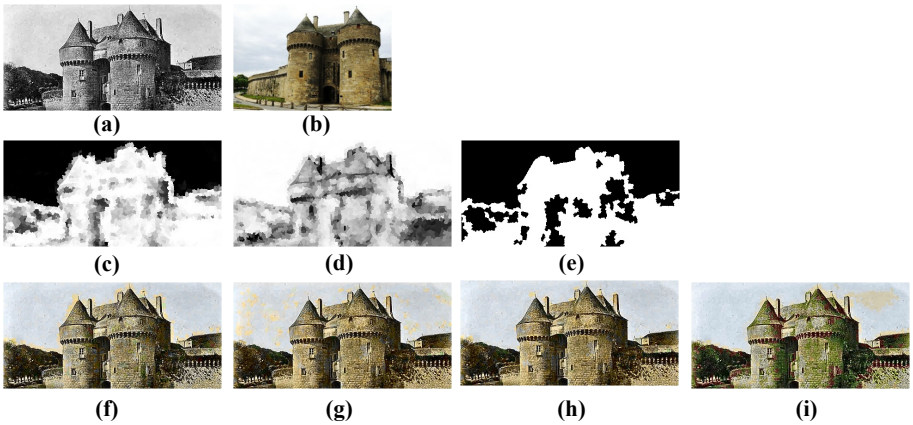


Figure 1: Illustration of automatic feature selection for colorization. (a) target grayscale image, (b) reference color image, (c)(d) probability maps of uniform and non-uniform regions (black to white for 0 to 1), (e) the optimal label image learned by an MRF model (black: uniform, white: non-uniform), (f)(g): colorization results using intensity feature and SURF texture feature respectively, (h) colorization result using our proposed automatic feature selection and fusion, (i) the result using direct combination of intensity and SURF texture features.

regions (Fig. 1(g)). Combining these two features improves the result (Fig. 1(i)), but numerous matching errors remain which lead to the green tint on the castle and the yellow tint in the sky. Our automatic feature selection and fusion is effective at avoiding such problems (Fig. 1(h)).

In this paper, we propose a novel image colorization method via automatic feature selection within a Markov Random Field (MRF) framework. To the best of our knowledge, this is the first work that exploits automatic feature selection and fusion for image colorization. Specifically, image regions can be generally classified as being uniform or non-uniform. In uniform regions, the luminance of pixels is evenly distributed, so the intensity distribution can represent these regions well, whereas in non-uniform regions, texture feature descriptors are effective to represent the patterns. Based on the learned distribution of intensity deviation for uniform and non-uniform regions, the probability of a given region being assigned a uniform or non-uniform label is estimated using Bayesian inference, which is then used for selecting suitable features. Instead of making individual decisions locally, we further develop an MRF model to

improve the labeling consistency where the probability is used for the label cost and similarity between the luminance of the neighboring regions for the smoothness cost. The MRF model can be efficiently solved by the graph cut algorithm, enhancing the local consistency of the colorization result. Finally, the colorization results are obtained by transferring corresponding chrominance information from the reference image to the target grayscale image.

The main contributions of the paper are summarized as follows: 1) We propose a novel approach to improving image colorization by local feature selection and fusion. 2) We develop a novel algorithm that classifies local image regions into uniform and non-uniform regions and applies suitable features. An MRF framework guided by Bayesian probability inference is further proposed to improve locality coherence. 3) We perform extensive experimental analysis both visually and quantitatively, which shows that the proposed method outperforms state-of-the-art methods.

The rest of this paper is organized as follows. We review work most relevant to this paper in Sec. 2, and then describe the proposed algorithm in detail in Sec. 3. Experimental results are shown in Sec. 4 and finally conclusions are drawn in Sec. 5.

2. Related Work

In general, existing image colorization methods can be divided into three categories: user-scribble based methods, example-based methods and methods that use a large number of training images. User-scribble based methods are semi-automatic, and they often require substantial user interaction as input. In the pioneering work by Levin et al. [8], some color scribbles on the target image are required as input, and then the color will be propagated based on least squares diffusion. However, there are obvious color bleeding effects around edges due to the isotropic nature of the diffusion. In order to better preserve the edge structure, an adaptive edge detection based colorization algorithm was proposed in [9]. To make the color region boundaries more consistent with human judgement, a saliency guided colorization technique was proposed in [10]. The approach first generates a saliency map of the reference and target images to predict the visual attention of human viewers, softly segmenting the images into foreground and

background regions. Color transfer is then performed first to the foreground and then the background using a weighted color transfer algorithm. In [11], a fast colorization method based on the geodesic distance weighted chrominance blending was proposed. Thanks to the use of luminance-weighted chrominance blending model and efficient intrinsic distance computation, the method is efficient for both image and video colorization. However, for all these scribble-based methods, it is time-consuming and the quality of colorization results highly depends on the appropriateness of user scribbles.

Compared with user-scribble based methods, example-based methods can be fully automatic without any user interaction. For example-based methods, typically only one reference image with color information is needed, and the target grayscale image is colorized automatically. The pioneering work by Welsh et al. [4] first finds the best matching sample in the reference image for each pixel in the target image, and then the chrominance information is transferred to the target grayscale image from the color reference image by the matching results to form the colorized images. Most of the existing example-based colorization algorithms follow this framework involving the steps of feature matching and color transfer. As feature matching is critical to the quality of results and the proposed method, Welsh's method resorts to manually specified swatches when automatic matching fails to produce satisfactory results.

In order to improve the feature matching performance, different features or different combinations of features have been proposed. Ying et al. [12] proposed using a more extensive neighborhood descriptor computed using co-occurrence matrix based texture features. To reduce artifacts caused by outliers, the edit-nearest-neighbor method [13] is used to try to remove the outliers. While the paper presents examples showing improved results, the co-occurrence matrix is expensive to compute. Chen et al. [14] combined [4] with foreground/background image matting to improve the colorization results but user interaction is needed to guide the grayscale image matting. Most of the existing methods focus on finding a proper combination of features for the *whole* image, rather than *selecting* proper features for each local pixel or region, as we propose to do in this paper.

As the methods above match each pixel in isolation, spatial consistency cannot be guaranteed in general. In order to enhance locality consistency and reduce color

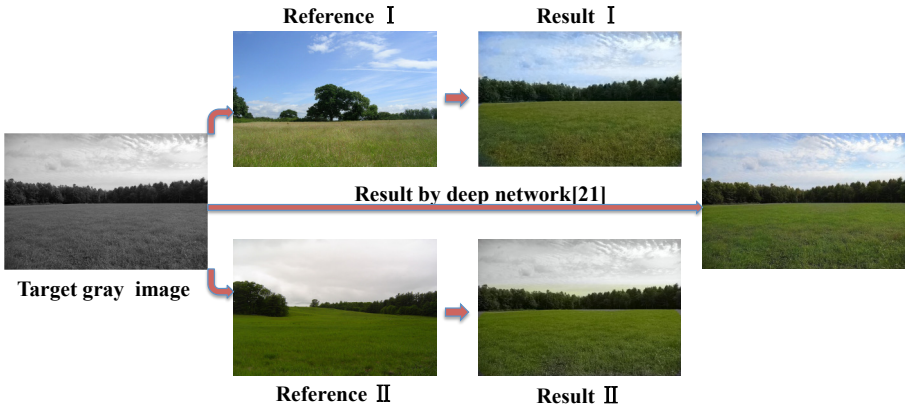


Figure 2: Colorization results with different reference images. While all the colorized images look plausible, our method is able to produce different colorized images based on different reference images. In comparison, deep learning based method [18] can only produce one output.

bleeding effects at edges, an edge-preserving total variation based image colorization algorithm was proposed in [15]. However, since only chrominance information is involved in the variational formulation, the results of [15] suffer from halo effects near strong contours. In [16], a coupled regularization term with luminance and chrominance channels is introduced to preserve image contours during the colorization process. The method produces colorization results which are better aligned with edge structures. However, significant artifacts can still be produced by incorrect feature matching. Compared with the local matching based methods, a novel global colorization method based on histogram regression was proposed in [17]. The basic assumption is that the final colorized image should have a similar color distribution as the reference image, and color matching is conducted by finding and adjusting the zero-points of the color histogram. The method however may not work well for complicated scenarios where the color mapping cannot be effectively represented using global histograms.

An alternative category of approaches resorts to a large number of training images. For example, the proliferation of internet images can be utilized for image colorization. In [19, 20, 21], target grayscale images are colorized by internet images. The reference images are searched from the internet based upon a semantic label given by

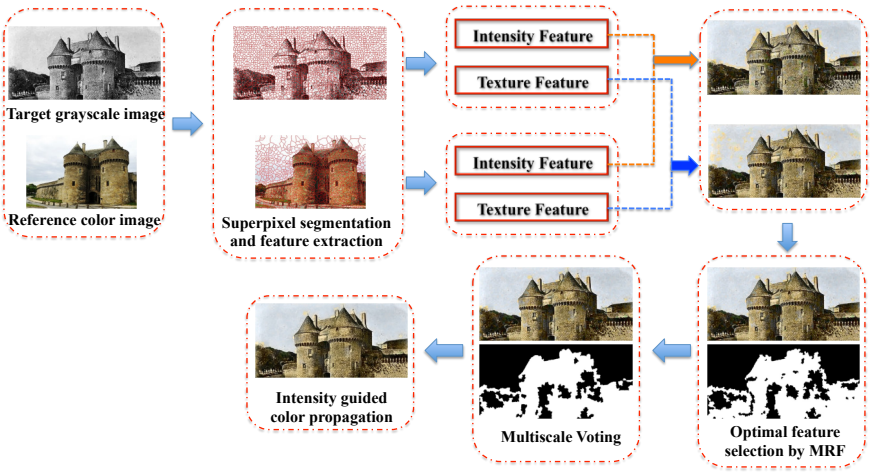


Figure 3: The pipeline of the proposed method.

the user and from the vast number of returned images a subset is chosen by means of a combined similarity metric. However, it is computationally expensive and depends on the accuracy of the semantic segmentation. Recently, deep learning based methods have been proposed for image colorization [6, 7, 18, 22] and produce promising results. However, unlike example-based methods, the colorization results cannot be controlled by users. Since image colorization is essentially an ill-posed problem: the target images can often be naturally colorized in different ways due to semantic ambiguities and style preference. One example is shown in Fig. 2. We can see that given different style reference images, our method, as an example-based method, can generate multiple distinct and plausible colorization results (e.g. the sky can be blue for a sunny day or gray for a cloudy day), while the recent deep learning based method [18] does not provide flexible control and can only produce one output image.

In this paper, a novel example-based image colorization method is proposed. Unlike existing methods, we aim to automatically find suitable features for each local region rather than using the same feature for the whole image globally. We further propose an MRF framework to solve feature selection and locality consistency simultaneously, which can be efficiently solved using the graph cut algorithm. As we will show later, our automatic feature selection helps to significantly improve feature matching,

and thus provides an effective solution to a major challenge of image colorization.

3. Our Method

The pipeline of the proposed algorithm is shown in Fig. 3. In order to suppress the influence of global luminance difference between the reference and target images, a global linear luminance remapping to the reference image is applied as in [4]. For computational efficiency, and to help improve the spatial consistency of the results, both the reference and target images are segmented into superpixels, and intensity and texture features are extracted from each superpixel. Using either of these features, we can find the corresponding best matching result for each target superpixel based on the Euclidean distance in the feature space (efficiently computed using the ANN library [23]). Then a two-label MRF model is formulated to choose the optimal correspondence according to different features, based on the probability of superpixels belonging to uniform or non-uniform regions. Following the initial labeling, a multi-scale voting process is performed to eliminate the isolated outliers and enhance locality consistency. Finally, the chrominance channels are filtered by the standard guided filter [24] with the guidance of the luminance channel.

3.1. Image Segmentation and Feature Extraction

For example-based image colorization, the most time-consuming step is finding the correspondence from the reference color image for each pixel in the target grayscale image. It is computationally expensive, especially when the feature dimensionality is high. On the other hand, neighboring pixels in natural images often share similar characteristics, and can be processed simultaneously and in the same manner. As we will demonstrate later, doing so also ensures coherent colorization in local neighborhoods and reduces mismatches. Based on the above observation, both the reference image and the target grayscale image are first segmented into superpixels. For the reference image, superpixel segmentation is performed using the color information, whereas the target grayscale image is segmented using the luminance information only. In this paper, we adopt the Turbopixel algorithm [25], which can process color and grayscale

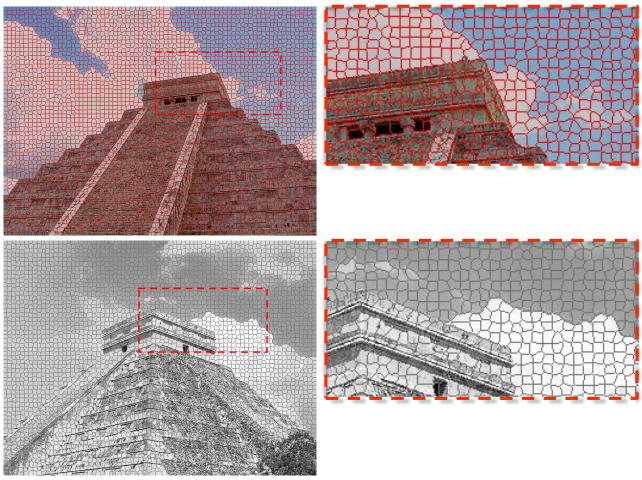


Figure 4: Examples of superpixel segmentation shown with magnified selection.

images while preserving the edge structure well. The superpixel number is set to 4000 in our experiments which provides a good balance between efficiency and quality, as we will demonstrate later. Note that the required number of superpixels depends on the image content, rather than its resolution. Fig. 4 shows the superpixel segmentation results for both color and grayscale images. From the magnification of the boundary areas, we can see that the edge structure is well preserved.

For each superpixel from the reference color image and the target grayscale image, two types of features are extracted:

Intensity feature. The intensity feature we used is a 27-dimensional vector computed based on the luminance value. It is composed of three parts: the mean intensity within the superpixel ($\bar{l}_1$), the mean intensity of the neighboring superpixels ($\bar{l}_2$) and the intensity distribution within the superpixel ($\mathbf{h}_l$), where $\bar{l}_2 = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\bar{l}_1(j)$ and $\mathcal{N}_i$ is the set of neighboring superpixels of the i^{th} superpixel. The intensity distribution $\mathbf{h}_l$ is a histogram of the intensity distribution within each superpixel. In our experiments, the intensity range 0–255 is divided into 25 bins, and each entry represents the proportion of the pixels whose the intensity is in the range of the bin compared with the total number of pixels in the superpixel. The intensity feature is denoted as $\mathbf{f}^I$.

Texture feature. In this paper, the scale-invariant and rotation-invariant SURF feature [26] is used as the texture descriptor. At each pixel a 128-dimensional SURF descriptor is extracted, and then the average SURF feature within the superpixel is computed as the texture feature of the superpixel. We denote the texture feature as $\mathbf{f}^T$.

These intensity and texture features are chosen because they are representative for their feature types and produce competitive results (see also the comparative results between our method and state-of-the-art methods). Since the focus of this paper is novel feature selection and fusion, these elementary features are fixed, although our method can be directly combined with alternative features. To help choose intensity or texture feature for matching, the intensity standard deviation d_i within each superpixel i is also computed. A small value of the standard deviation implies that the intensity distribution within the superpixel is even. The standard deviation is used for automatic selection of features (see the next subsection for detail), rather than finding the best candidate.

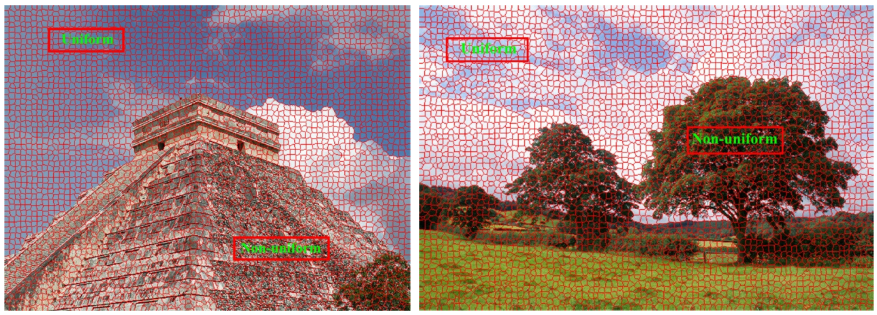


Figure 5: Examples of training samples from uniform and non-uniform regions.

3.2. MRF-based Automatic Feature Selection

In this subsection, we discuss our novel automatic feature selection for image colorization. The intuition is that different regions can be better represented by different features. In general, a natural image can be decomposed into uniform and non-uniform regions. We use the term non-uniform in a broad sense to refer to regions containing sufficient details, to allow texture descriptors to work effectively. This is different from

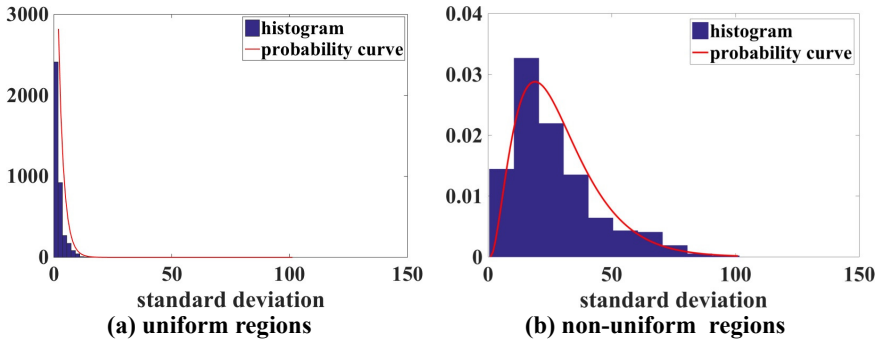


Figure 6: The histograms of standard deviation of superpixels extracted from uniform and non-uniform training examples, and their fitted Gamma distributions.

the standard texture/non-texture classification [27, 28] so we develop a simple approach for our purpose. In uniform regions, the luminance of pixels is evenly distributed, and the intensity feature $\mathbf{f}^I$ can represent these regions well. While in non-uniform regions, texture feature descriptors $\mathbf{f}^T$ are a good choice to represent repeated patterns. One of the main contributions of this paper is to design an automatic feature selection framework for each superpixel within an MRF framework. In addition to feature selection, locality consistency can also be simultaneously enhanced by the MRF model.

3.2.1. Probability Estimation for Region Uniformity

When the superpixel segmentation is dense enough, the intensity standard deviation of each superpixel can be seen as a good descriptor to determine its characteristics, i.e. smaller deviation implies uniform while bigger value means non-uniform. Therefore we adopt intensity standard deviation as the feature variable to estimate the probability distribution of each type of region.

In this paper, Bayesian inference is used to determine the probability of each superpixel belonging to each type of region. For the i^{th} superpixel x_i, given its corresponding standard deviation d_i, we denote its probability of belonging to a uniform region as $P(x_i \in U|d_i)$. Using Bayesian inference, the posterior probability can be computed by

$$P(x_i \in U|d_i) = \frac{P(U)P(d_i|x_i \in U)}{P(U)P(d_i|x_i \in U) + P(N)P(d_i|x_i \in N)}, \tag{1}$$

where $P(U)$ (or $P(N)$) denotes the a priori probability for a region to be uniform (or non-uniform), and $P(d_i|x_i \in U)$ (or $P(d_i|x_i \in N)$) is the conditional probability of having given standard deviation d_i for a region known as uniform (or non-uniform). In general, we assume uniform and non-uniform regions are equally common and the a priori probabilities $P(U)$, $P(N)$ can be set as 0.5.

In order to estimate the conditional probability $P(d_i|x_i \in U)$ and $P(d_i|x_i \in N)$, we create a training set by manually selecting superpixels from uniform and non-uniform regions, and calculating the corresponding standard deviation values. In this paper, we collected 3000 superpixels for each type of region from 10 images, which are sufficient to estimate the distributions of standard deviation. An example is shown in Fig. 5. The histograms of the standard deviation for uniform and non-uniform regions are shown in Fig. 6. The shape of the histogram can be well approximated by a Gamma distribution

$$\Gamma(\alpha, \beta) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-x\beta}}{\Gamma(\alpha)} \tag{2}$$

where α and β are the shape and scale parameters of the Gamma distribution. Given our training samples, the Gamma distributions fitted are shown in Fig. 6, with the parameters for uniform regions and non-uniform regions being: $\{\alpha_U = 0.9544, \beta_U = 2.3424\}$ and $\{\alpha_N = 2.8295, \beta_N = 9.8095\}$.

Given a superpixel with luminance standard deviation d_i, its probability of belonging to uniform regions $P(x_i \in U|d_i)$ can be computed using Eqn. 1. We can similarly compute its probability of belonging to non-uniform regions $P(x_i \in N|d_i)$, which satisfies $P(x_i \in U|d_i) + P(x_i \in N|d_i) = 1$. The probability of uniformity is also useful for feature matching as uniform regions should generally be colorized using samples from uniform regions, and the same for non-uniform regions. We thus add $P(x_i \in U|d_i)$ to each type of the features introduced in the previous section which serves as a soft constraint.

3.2.2. MRF-based Labeling for Feature Selection

With the estimated posterior probability, a trivial way of labeling superpixels as uniform or non-uniform is thresholding. Such approach however does not take into account spatial consistency. In this paper, we propose a novel automatic feature selection

approach for image colorization within an MRF framework. For each superpixel in the target image, we can find two matched superpixels from the reference image based on the intensity and texture features. The searching process can be efficiently performed using the approximate nearest neighbour (ANN) tree searching algorithm [23]. An example is shown in Fig. 1. We can see that the intensity features perform well within uniform regions (Fig. 1(f)), whereas the texture descriptor is effective at non-uniform regions (Fig. 1(g)). Most existing methods combine these two types of features as a combined feature and use the same searching process in both cases. The result as shown in (Fig. 1(i)) still contains many incorrect matching results. Rather than combining the two features, in this paper we assume that for each superpixel the matching result is determined by only the single optimal feature. We assume that in uniform regions the colorization should be determined by the intensity feature, whereas in the non-uniform regions the texture feature should be dominant. Therefore, the problem of feature selection can be regarded as a binary labeling problem, where label 0 denotes intensity feature and label 1 means texture feature.

Let us denote the i-th superpixel in the target image as x_i, and the set of all superpixels in the target image as Ω. $\mathcal{N}$ represents the set of adjacent superpixel pairs. The task of feature selection is to divide the whole set Ω into two disjoint sets, Ω_I for regions determined by intensity features and Ω_T for regions determined by texture features. The binary labeling problem can be formulated as the minimization of the following MRF energy function

$$\min_{\mathbf{S}} E(\mathbf{S}) = \sum_{i \in \Omega} D(S_i) + \lambda \sum_{(i,j) \in \mathcal{N}} f(S_i, S_j), \tag{3}$$

where S_i is the label for the i-th superpixel, which takes 0 or 1 to indicate whether the intensity feature or the texture feature is selected.

The first term $D(S_i)$ is the label cost which measures the cost to assign label S_i to the i-th superpixel x_i. Based on our assumption, the label cost of each superpixel should be determined by its probability of belonging to either the uniform regions or

non-uniform regions, i.e. the posterior probabilities:

$$D(S_i) = \begin{cases} P(x_i \in N|d_i) & \text{if } S_i = 0, \\ P(x_i \in U|d_i) & \text{if } S_i = 1. \end{cases} \tag{4}$$

The second term $f(S_i, S_j)$ is the pairwise term, which indicates the cost for assigning label S_i to the i-th superpixels while assigning label S_j to the j-th superpixel. In this paper, the pairwise term is defined as

$$f(S_i, S_j) = \begin{cases} 0, & \text{if } S_i = S_j \\ s(i,j), & \text{otherwise} \end{cases} \tag{5}$$

where $s(i, j)$ describes the similarity between adjacent superpixels. For improved coherence, we assume that neighboring superpixels with similar intensity values are likely to have the same label. Therefore, the function $s(i, j)$ is defined as follows:

$$s(i,j) = \exp\left\{-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\sigma_1^2}\right\} \exp\left\{-\frac{(\bar{l}_1(i) - \bar{l}_1(j))^2}{2\sigma_2^2}\right\}, \tag{6}$$

where $\mathbf{c}_i$ and $\bar{l}_1(i)$ are the central location and the mean intensity of the i-th superpixel. σ_1, σ_2 are parameters set as $\sigma_1 = 100$ and $\sigma_2 = 1$ (for images with approximately 1 megapixels). From the definition, the pairwise costs only exist for adjacent superpixels with different labels, and the effect is determined by the similarity between superpixels: i.e., the cost will be big if different labels are assigned to nearby superpixels with high similarity. Therefore, the main effect of the pairwise term is to enhance locality consistency, which is important for colorization.

Given the label cost and the pairwise terms, the two-label MRF model (3) can be optimized by the graph cut algorithm [29, 30]. Since our energy function is not sub-modular, the alpha-beta swap algorithm is adopted, which randomly selects two labels from the label set and tries to reduce the energy by swapping these labels. The algorithm is efficient, and runtime is less than 0.1 s for 4000 superpixels.

Finally, feature selection is determined by the labeling. For regions with label 0, i.e., uniform regions, the intensity feature is used for feature matching. Otherwise, the texture feature is used. Fig. 1 shows the colorization result by the proposed feature selection method. (c) and (d) are the corresponding label costs for the intensity

feature and the texture feature. (f) and (g) are the corresponding best matches using intensity and texture features, respectively. (e) is the optimal labeling obtained by the MRF model and (h) is the final colorization result. Our feature selection performs significantly better than using individual features, or their simple combination as shown in (i).

3.3. Consistency enhancement by multiscale voting

Although locality consistency has been taken into account in the MRF energy function, there still exist isolated wrong matches as shown in Figure 7(a). In order to improve the color consistency, a multiscale voting is further performed. The basic intuition is that the label assignment for a superpixel is likely to be wrong if most of its neighboring superpixels with similar image properties are assigned another label. Therefore, we can exploit neighboring superpixels to identify and correct invalid label assignments.

The target image is also oversegmented to produce coarse-scale superpixels by the Turbopixels method. In this paper, the number of coarse-scale superpixels is chosen as a quarter of the original superpixel segmentation. For each superpixel in the coarse scale, the final label is determined by the majority label obtained in the fine scale. Finally, median filtering is applied for the chrominance channels in the lab color space such that the color of each superpixel is replaced by the median of its neighboring superpixels, to further enhance color consistency. The result of multiscale voting is shown in Fig. 7(b). From the magnified selection, we can see that many isolated wrong matches are corrected.

3.4. Color propagation by guided filter

Although the multiscale voting process can reduce isolated matching errors, the resulting color images still show block effects as chrominance information is transferred on a superpixel basis, and can have quite a few sparse outliers, as shown in Fig. 8(a). Different from other image processing tasks, the target grayscale image is accurate for image colorization. Therefore, the intensity information of the target image can be

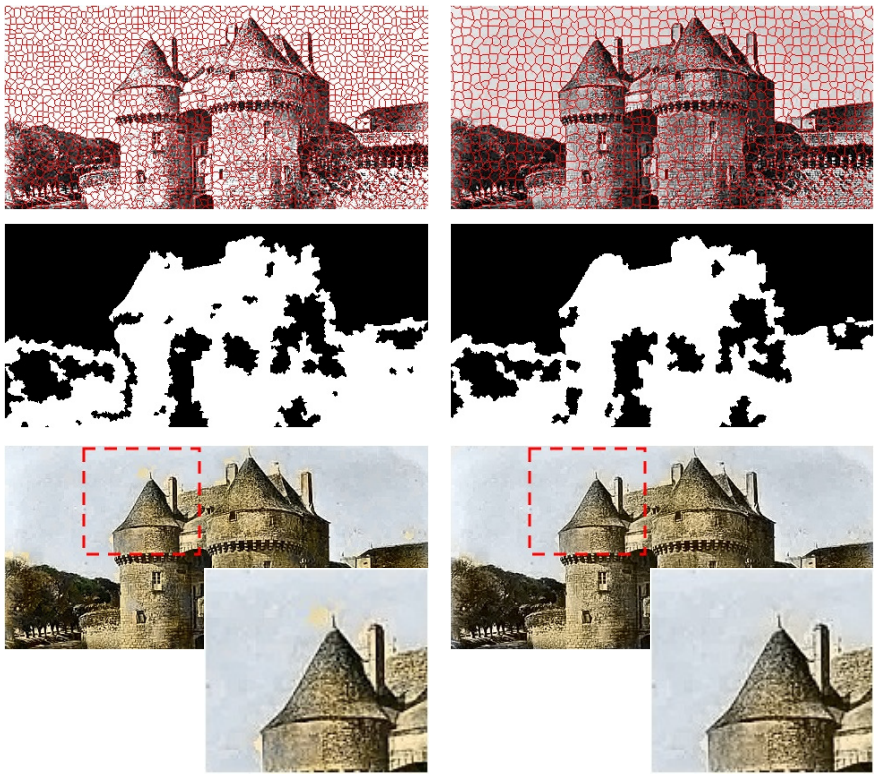


Figure 7: Example of multiscale voting. (a) fine scale matching without multiscale voting, (b) result with multiscale voting. From top to bottom: fine and coarse scale superpixels, uniform/non-uniform labels without and with multiscale voting, and colorization results without and with multiscale voting.

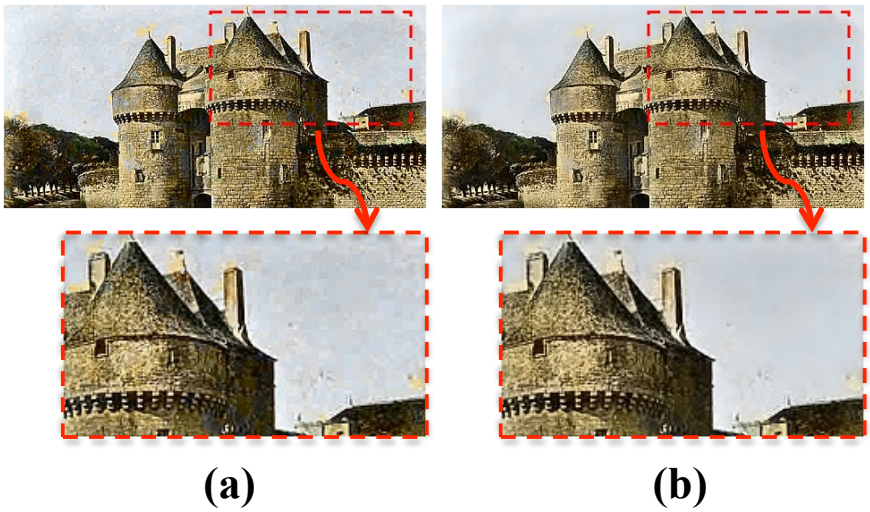


Figure 8: Example of luminance guided image filtering. (a) the original matching result, (b) the result after guided image filtering.

used to enhance the results, by exploiting the fact that nearby pixels with similar intensity values should come with similar colors. For this purpose, we adopt the guided filter [24] for color propagation where the target grayscale image is used as guidance, which is an edge-preserving smoothing operator which better utilizes the structures in the guidance image.

$$\mathbf{J}^* = \sum_j \mathbf{W}_{ij}(\mathbf{I})\mathbf{J}_j \tag{7}$$

where $\mathbf{I}$ means the target grayscale image which is used as the guidance image, $\mathbf{J}$ is the obtained chrominance channel from the last step, and $\mathbf{W}$ is the adapted weight function as defined in [24]. The filtered image is shown in Fig. 8(b). We can see that block effects are smoothed while the edge information is preserved well.

4. Experiments

In this section, experimental results are presented to evaluate the performance of the proposed method, and compare it with several state-of-the-art methods [15, 16, 17]. In

order to make a fair comparison, the results of [17] are generated using the code provided by the author, and the results of [15] and [16] are provided by the authors. The results of different algorithms are evaluated both by visual inspection and by quantitative evaluation. In addition to example-based methods, two of the latest deep learning based methods [18, 22] are also included for visual comparison, using the code provided by authors. Since the standard quantitative measures we used to compare the results are not designed for the task of image colorization, in order to get fair and reliable comparison, a user study is also performed. The experiments were carried out on a computer with an Intel i7 2.8GHz CPU and 16GB memory.

4.1. Visual inspection

Thirteen natural images of different types, e.g., landscape, animals, and buildings, are chosen for the experiment. The colorization results are shown in Fig. 9. In order to evaluate the performance of automatic feature selection, the colorization results using the combined intensity and texture features are also presented as a baseline. From visual inspection, we can see that in most cases the proposed algorithm achieves the best results compared against the other four methods. In particular, our method substantially reduces wrong color matches, especially between uniform and non-uniform regions.

Method [17] is based on finding and adjusting the zero-points of the histograms of both the reference and target images. While less likely to be affected by local content, the global mechanism can result in mismatches within large regions when the zero-point based correspondence contains errors (see the 1st, 3rd and 4th rows of Fig. 9). Both the methods [15] and [16] solve the image colorization problem by automatically selecting the best color among a set of color candidates via a total variational framework. Method [15] only retains the U and V channels without coupling of the chrominance channels with the luminance, and as a result their regularization algorithm creates halo effects near strong contours, as shown in the 1st, 4th, 7th and 11th rows of Fig. 9. In [16], a strong regularization coupling the luminance and chrominance channels is proposed to preserve the structural information of the image during the colorization process, such as edge and color consistency. It achieves significant

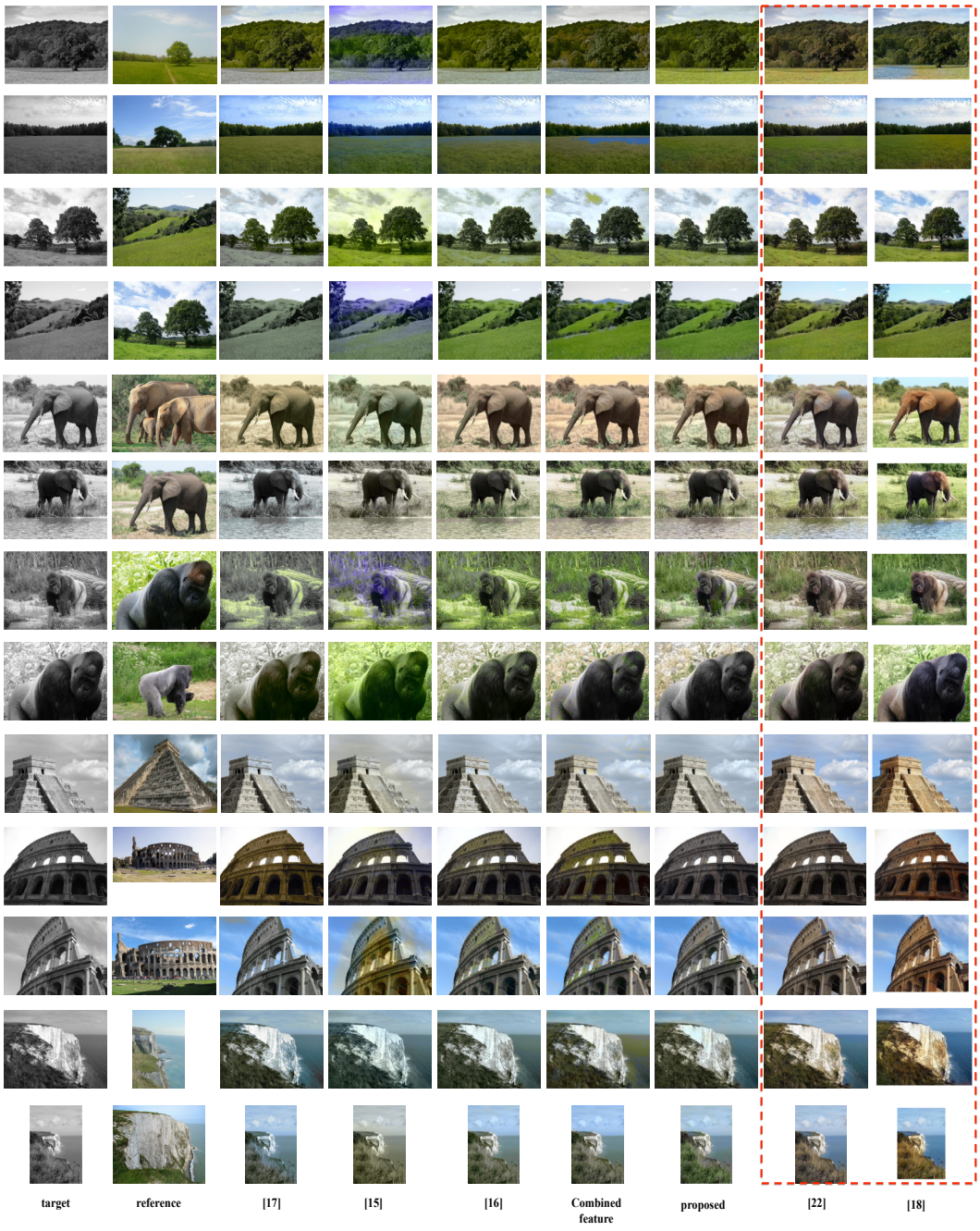


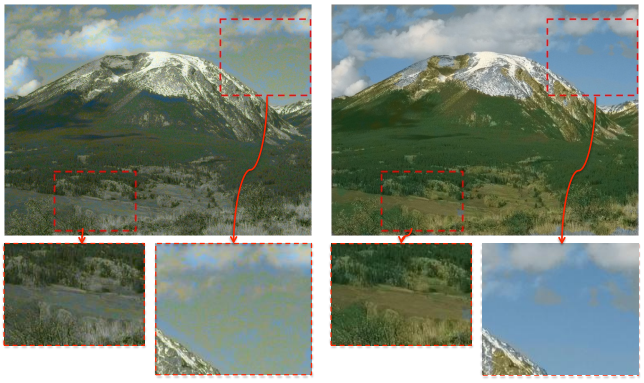
Figure 9: Comparison of our colorization results with alternative methods. From left to right: target and reference images, and the results of different methods.

Figure 10: Comparison of our colorization results with alternative example-based colorization methods.

improvement over the method [15]. However, wrong matches remain for challenging scenarios (e.g. 1st-3rd rows of Fig. 9). The sixth column shows the results generated with trivially combined intensity and texture features. We can see that there are numerous incorrect matches, e.g., the green grass is mistakenly matched to the blue sky in the 2nd row of Fig. 9, and the uniform gray sky is mistakenly matched to the grass in the 3rd row of Fig. 9. In contrast, with automatic feature selection using our optimization framework, the proposed method improves color matching substantially and achieves better results as shown in the seventh column.

In addition to example-based methods, we also compare our colorization results against the latest deep-learning based methods [18, 22]. Compared with example-based methods, the deep learning based colorization algorithms use millions of images for training the neural networks. In general, they can generate reasonable color images, as shown in the 8th and 9th columns of Fig. 9. However, there are some obvious artifacts shown e.g. in the 1st and 5th rows where parts of the meadow and elephant are colored in blue. In addition, the output of such deep learning based methods cannot be controlled by the user.

Some colorization results for scenes with complex structure and large color variations are shown in Fig. 10. These examples are more challenging, as regions with substantially different colors can have similar local characteristics in grayscale images. We



(a) Result generated by pixel (b) Result generated by superpixel

Figure 11: Colorization results by matching of pixels and superpixels.

compare our results with state-of-the-art example-based colorization methods[1]. Due to the global mechanism of the method [17], it fails to reproduce correct colors from the reference images, as shown in the first two rows of Fig. 10. More specifically, it confuses the sand beach with the sea (first row) and grass meadow with the sky (second row), and produces large wrongly colored blue regions. It also generates obvious halo effects around the boundary in the third row. The method [16] finds the optimal color matching using a variational framework. Thanks to the edge preserving capability of the variational function, [16] can reduce the halo effects around the boundary effectively. However, as only one type of feature is used to find the matching candidates, many obviously wrong colors are assigned. For example, in the first row, a large portion of the beach is wrongly colored in blue as it is matched to the sky. This can be effectively avoided by using intensity feature in uniform regions as suggested in the proposed method. Similar problems can also be found in the remaining examples. In comparison, our method produces significantly better results than [15] and [16] for all of these cases.

Our method uses superpixels rather than pixels for matching. In this paper, an approximate nearest neighbour (ANN) tree search algorithm is used to find the nearest

[1]For this experiment, we compare with methods where code is publicly available, so the result of [15] is not included.

Figure 12: Colorization results with different numbers of superpixels. The first and third rows are the target and reference images. (a)-(e) in the second and fourth rows are the corresponding results with 1000, 2000, 4000, 6000 and 8000 superpixels.

match. ANN is computationally expensive when the feature dimensionality is high. For example, assuming both the reference image and the target image are of size 400×300, for the 128-dimensional SURF descriptor, it takes *over a day* to find matching pixels by using the fast ANN algorithm[2]. In contrast, if the images are segmented into 4000 superpixels, the ANN algorithm just needs $1.58s$. In addition, neighboring pixels in natural images often share similar characteristics, so processing them simultaneously is not only substantially faster, but also improves local coherence and reduces the chance of mismatches. An example is shown in Fig. 4.1. To make pixel-based colorization more tractable, we reduce the SURF feature dimension to 30 using PCA (Principal Component Analysis) which generally produces an output with similar quality. However, the ANN step still takes $863.82s$, and the result is clearly less coherent than that with superpixels. Based on the above observation, both the reference image and the target grayscale image are firstly segmented into superpixels.

[2]http://www.cs.ubc.ca/research/flann/

An additional experiment is designed to evaluate the influence of the number of superpixels. We choose the number of superpixels as 1000, 2000, 4000, 6000 and 8000, and apply our colorization algorithm. Fig. 12 shows the results with different number of superpixels and Table 1 shows the corresponding total running times, where both target and reference images are of the size 768×480. It can be seen that when the number of superpixels is too small, e.g., 1000, a superpixel may cover regions of mixed characteristics, which could result in a small number of mismatches. When the superpixel number is larger than 4000, the result is visually similar. However, the computational complexity and running times increase dramatically. In this paper, the number of superpixels is set to 4000 by default.

Table 1: The running times of our algorithm using different numbers of superpixels.

Number	1000	2000	4000	6000	8000
Time (s)	41.53	73.27	135.07	262.53	381.13

4.2. Quantitative comparisons

In addition to visual inspection, we also make quantitive comparisons for the results shown in Figure 9. In this subsection, the results of deep learning methods [18, 22] are not included. On one hand, deep learning methods use millions of images for training while example-based methods take only one reference image. On the other hand, example-based methods generate results according to the given reference image, whereas the results of deep models are not controlled and thus can be irrelevant to the reference image. We thus focus on comparing our method with state-of-the-art example-based methods in the quantitative analysis to avoid misinterpretation. In this paper, we use the standard Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) [31] as the measurements. Given an $m \times n$ ground truth color image u_0 and the colorization result u, PSNR (in db) is defined as

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_I^2}{MSE}) \tag{8}$$

Table 2: The quantitative comparisons of different algorithms using the standard PSNR and SSIM measures. I–III are the corresponding results of methods [17, 15, 16], IV are the colorization results obtained by the combined features, and V are the results of the proposed feature selection method. 1–13: corresponding test images as shown in Fig. 9.

	PSNR					SSIM				
	I	II	III	IV	V	I	II	III	IV	V
1	20.91	16.01	21.63	20.71	**24.71**	0.64	0.25	0.72	0.68	**0.79**
2	22.66	17.40	22.68	20.91	**24.49**	0.72	0.39	0.69	0.64	**0.75**
3	21.03	17.97	24.88	24.03	**25.61**	0.73	0.39	0.84	0.82	**0.87**
4	20.28	17.09	**26.92**	23.58	23.19	0.69	0.45	**0.91**	0.80	0.87
5	23.38	22.65	**24.31**	21.36	22.39	0.69	0.65	**0.75**	0.64	0.68
6	24.39	22.39	**24.55**	18.18	23.49	**0.87**	0.75	0.85	0.56	0.83
7	19.78	14.94	21.84	20.84	**23.02**	0.68	0.37	0.82	0.79	**0.83**
8	18.37	17.89	18.82	18.37	**18.91**	0.53	0.51	0.59	0.55	**0.63**
9	**20.17**	16.26	18.35	18.07	19.598	**0.78**	0.30	0.68	0.63	0.76
10	23.93	22.75	**31.12**	25.15	23.50	0.66	0.42	**0.90**	0.69	0.79
11	14.33	13.06	14.04	14.31	**14.38**	0.47	0.33	0.46	0.48	**0.49**
12	19.76	19.61	20.33	18.40	**24.42**	0.71	0.64	0.73	0.55	**0.85**
13	23.06	20.36	**25.51**	24.59	23.89	0.87	0.54	**0.89**	0.88	0.86

where MAX_I is the maximum possible pixel value of the image, which is 255 (with standard 8-bit samples). MSE is the mean squared error, which is defined as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [u(i,j) - u_0(i,j)]^2. \tag{9}$$

SSIM is designed to improve on traditional methods such as PSNR and MSE for better consistency with human visual perception. The scores for different algorithms are shown in Table 2. The quantitive measurements are generally in line with our visual inspection.

4.3. User study

However, since the standard measures are not designed for the task of image colorization, in some cases, visually better results may get worse scores. For example, for the examples in 9th and 10th row of Fig. 9, the results of the proposed method appear more natural by visual inspection, but the PSNR and SSIM scores of our method are lower than those of [16, 17]. Therefore, in order to make a fair comparison, we also perform a user study to quantitatively evaluate our method against other methods.

We designed the user study following the 2AFC (Two-Alternative Forced Choice) paradigm, which is widely used in psychological studies as it is both simple and reliable. 200 users participated in the user study, with ages ranging from 18 to 38. For each test image, every pair of results generated by the different algorithms is shown to the user and he/she is asked to choose the one of them, that looks better. To avoid potential bias, we randomize the order of image pairs shown to the user as well as their left/right position. We record the total number of user preferences (clicks) for each method, and treat these as random variables. The distribution of user preferences for each method is summarized in Fig. 13. Note that since each method is compared with 4 alternative methods, and 13 examples were tested, the maximum number of user preference for each method is 52. The one-way analysis of variance (ANOVA) is used to analyze the user study results. ANOVA is designed to determine whether there exist statistically significant differences between two or more independent groups. ANOVA analysis gives the p-value for the null hypothesis that the means of the groups are equal. Smaller p-value means the groups are more significantly different.

In this paper, the p-values are computed between our method and each alternative method. The results are shown in Table 3. It is clear that all of the p-values are very small which demonstrates the difference between our method and each alternative method is statistically significant, based on the user study. From Fig. 13 we can see that majority of users prefer the method proposed in this paper (Fig. 13 V) which has the highest mean score.

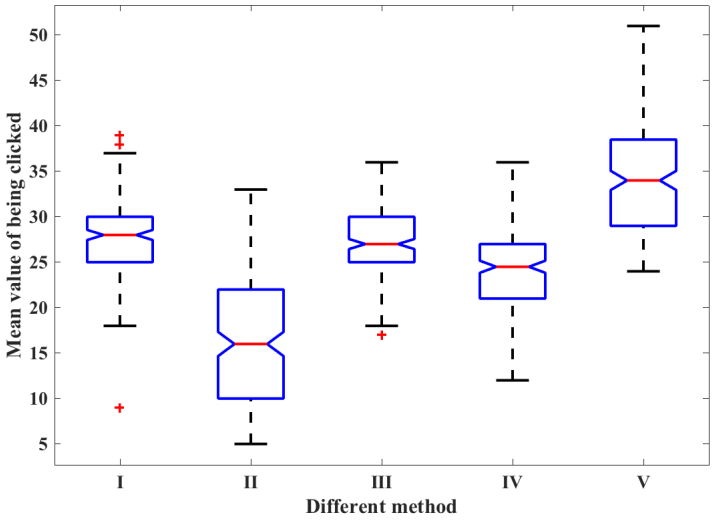


Figure 13: Boxplots of user preferences for different methods, showing the mean (red lines), quartiles (blue lines), and extremes (black lines) of the distributions.

Table 3: The p-values of the ANOVA tests of the proposed method against other methods based on the user study results.

method	I [17]	II[15]	III[16]	IV (combined)
p-value	3.22e-30	2.05e-93	4.09e-38	1.43e-59

5. Conclusion

In this paper, we propose a novel approach to image colorization based on automatic feature selection. By using suitable features for local regions based on their uniform/non-uniform characteristics, feature matching quality can be significantly improved, producing visually better colorization results. To achieve this, Bayesian inference is used to estimate the label (type) of each region, and the optimization of feature selection is effectively formulated using a two-label MRF model. We further use multiscale voting and luminance guided image filtering to improve the consistency of the final colorization results. By feature selection, our method is able to produce colorization results better than individual features. However, both intensity and texture features used are low-level features and may fail to find semantically correct matches. Our

method can produce unsatisfactory results in case both features give wrong matches. To address this, as the idea of using feature selection for improved image colorization is general, in addition to the algorithm pipeline proposed in the paper, we would like to investigate automatic feature selection in alternative colorization frameworks e.g. by using more robust and semantically meaningful features, as well as for other applications such as color transfer.

Acknowledgements

We would like to thank the authors of [17, 18, 22] for providing their code, and the authors of [15, 16] for helping to generate comparative experimental results.

[1] S. A. Tsaftaris, F. Casadio, J.-L. Andral, A. K. Katsaggelos, A novel visualization tool for art history and conservation: Automated colorization of black and white archival photographs of works of art, Studies in Conservation 59 (3) (2014) 125–135.

[2] H. Chang, O. Fried, Y. Liu, S. DiVerdi, A. Finkelstein, Palette-based photo recoloring, ACM Transactions on Graphics (TOG) 34 (4) (2015) 139.

[3] Y. Chang, S. Saito, K. Uchikawa, M. Nakajima, Example-based color stylization of images, ACM Transactions on Applied Perception 2 (3) (2006) 322–345.

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

[5] Y. Ling, O. C. Au, J. Pang, J. Zeng, Y. Yuan, A. Zheng, Image colorization via color propagation and rank minimization, in: International Conference on Image Processing, IEEE, 2015, pp. 4228–4232.

[6] A. Deshpande, J. Rock, D. Forsyth, Learning large-scale automatic image colorization, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 567–575.

[7] Z. Cheng, Q. Yang, B. Sheng, Deep colorization, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 415–423.

[8] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, ACM Trans. Graph. 23 (3) (2004) 689–694.

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

[10] N. Anagnostopoulos, C. Iakovidou, A. Amanatiadis, Y. Boutalis, S. Chatzichristofis, Two-staged image colorization based on salient contours, in: International Conference on Imaging Systems and Techniques (IST), 2014, pp. 381–385.

[11] L. Yatziv, G. Sapiro, Fast image and video colorization using chrominance blending, IEEE Trans. Image Processing 15 (5) (2006) 1120–1129.

[12] J. Ying, L. Ji, Pattern recognition based color transfer, in: International Conference on Computer Graphics, Imaging and Vision: New Trends, IEEE, 2005, pp. 55–60.

[13] F. J. Ferri, J. V. Albert, E. Vidal, Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 29 (5) (1999) 667–672.

[14] T. Chen, Y. Wang, V. Schillings, C. Meinel, Grayscale image matting and colorization, in: Proceedings of Asian Conference on Computer Vision, Citeseer, 2004, pp. 1164–1169.

[15] A. Bugeau, V.-T. Ta, N. Papadakis, Variational exemplar-based image colorization, Image Processing, IEEE Transactions on 23 (1) (2014) 298–307.

[16] F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, V.-T. Ta, Luminance-chrominance model for image colorization, SIAM Journal on Imaging Sciences 8 (1) (2015) 536–563.

[17] S. Liu, X. Zhang, Automatic grayscale image colorization using histogram regression, Pattern Recognition Letters 33 (13) (2012) 1673–1681.

[18] R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in: European Conference on Computer Vision, 2016, pp. 649–666.

[19] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, S. Lin, Semantic colorization with internet images, in: ACM Transactions on Graphics (TOG), Vol. 30, ACM, 2011, p. 156.

[20] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, P.-A. Heng, Intrinsic colorization, in: ACM Transactions on Graphics (TOG), Vol. 27, ACM, 2008, p. 152.

[21] X. Wang, J. Jia, H. Liao, L. Cai, Image colorization with an affective word, in: Computational Visual Media, Springer, 2012, pp. 51–58.

[22] S. Iizuka, E. Simo-Serra, H. Ishikawa, Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, ACM Transactions on Graphics (TOG) 35 (4) (2016) 110.

[23] D. M. Mount, S. Arya, ANN: library for approximate nearest neighbour searching.

[24] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (6) (2013) 1397–1409.

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

[26] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding 110 (3) (2008) 346–359.

[27] J. Li, J. Z. Wang, G. Wiederhold, Classification of textured and non-textured images using region segmentation, in: Proceedings of International Conference on Image Processing, Vol. 3, IEEE, 2000, pp. 754–757.

[28] L. Costantini, L. Capodiferro, M. Carli, A. Neri, Textured areas detection and segmentation in circular harmonic functions domain, in: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2012, pp. 829504–829504.

[29] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239.

[30] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1124–1137.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

Appendix

In this appendix, the original scores of the user study are listed in Fig. 1 and Fig. 2. We choose the different colorized images of the five methods (I [17], II [15], III [16], IV (results by combined feature) and V (our results by selective feature)) on 13 test images. With the given five results of each test image, we generate 10 pairs of colourized images, by which the user study results can be properly ordered. And these 10 pairs of images are randomly shown to the users to identify which method is better. If the user prefers the first result of the current pair of images, the score of the first method will be incremented by 1. 200 users joined the user study, and each user gave their judgment on 10×13 pairs of images; 26,000 clicks were collected in total. The score of each method by 200 users is shown in Fig. 1 and Fig. 2.

user / method	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
I[17]	25	20	28	28	21	30	32	30	27	27
II[15]	9	9	20	20	29	9	16	15	14	14
III[16]	34	24	32	32	19	36	25	31	24	24
IV(combined)	19	26	25	25	32	15	21	24	23	24
V(ours)	43	51	25	25	29	40	36	30	42	41

user / method	user 11	user 12	user 13	user 14	user 15	user 16	user 17	user 18	user 19	user 20
I[17]	29	29	33	28	31	29	35	20	21	20
II[15]	17	18	10	12	8	11	8	14	24	24
III[16]	26	27	27	26	28	23	27	27	30	25
IV(combined)	25	23	20	21	32	30	20	28	27	33
V(ours)	33	33	40	43	31	37	40	41	28	28

user / method	user 21	user 22	user 23	user 24	user 25	user 26	user 27	user 28	user 29	user 30
I[17]	25	34	26	26	39	28	28	22	27	28
II[15]	19	14	28	28	9	15	15	31	12	9
III[16]	23	23	21	21	32	28	28	22	26	29
IV(combined)	32	25	25	25	15	19	19	30	26	19
V(ours)	31	34	30	30	35	40	40	25	39	45

user / method	user 31	user 32	user 33	user 34	user 35	user 36	user 37	user 38	user 39	user 40
I[17]	24	34	27	31	34	26	29	25	28	25
II[15]	28	15	13	15	9	18	21	20	20	8
III[16]	26	26	28	29	28	26	19	26	25	26
IV(combined)	24	21	22	17	19	25	28	21	29	31
V(ours)	28	34	40	38	40	35	33	38	28	40

user / method	user 41	user 42	user 43	user 44	user 45	user 46	user 47	user 48	user 49	user 50
I[17]	24	38	24	33	25	35	20	31	34	27
II[15]	23	13	26	18	20	8	12	11	15	21
III[16]	33	29	30	29	26	28	31	28	24	29
IV(combined)	20	12	24	18	30	21	31	26	21	27
V(ours)	30	38	26	32	29	38	36	34	36	26

user / method	user 51	user 52	user 53	user 54	user 55	user 56	user 57	user 58	user 59	user 60
I[17]	27	24	31	29	30	32	22	32	25	25
II[15]	22	25	21	10	17	10	5	9	33	17
III[16]	29	29	27	29	28	30	28	27	20	30
IV(combined)	26	24	30	25	19	18	34	24	26	22
V(ours)	26	28	21	37	36	40	41	38	26	36

user / method	user 61	user 62	user 63	user 64	user 65	user 66	user 67	user 68	user 69	user 70
I[17]	28	31	31	27	28	28	30	31	29	35
II[15]	23	29	29	6	8	20	19	15	7	12
III[16]	28	17	17	31	30	29	28	32	30	27
IV(combined)	25	22	22	25	22	21	16	17	24	19
V(ours)	26	31	31	41	42	32	37	35	40	37

user / method	user 71	user 72	user 73	user 74	user 75	user 76	user 77	user 78	user 79	user 80
I[17]	21	29	26	25	32	30	24	39	24	23
II[15]	24	11	13	13	6	8	10	10	22	22
III[16]	22	26	31	25	27	30	23	28	32	27
IV(combined)	36	22	27	24	27	26	29	18	19	30
V(ours)	27	42	33	43	38	36	44	35	33	28

user / method	user 81	user 82	user 83	user 84	user 85	user 86	user 87	user 88	user 89	user 90
I[17]	29	24	24	30	27	18	32	34	24	24
II[15]	12	29	29	9	20	8	11	18	21	21
III[16]	33	20	20	26	22	29	27	25	27	27
IV(combined)	25	25	25	20	27	34	20	17	26	26
V(ours)	31	32	32	45	34	41	40	36	32	32

user / method	user 91	user 92	user 93	user 94	user 95	user 96	user 97	user 98	user 99	user 100
I[17]	25	36	28	28	33	23	29	27	22	29
II[15]	11	6	10	10	18	14	7	25	21	12
III[16]	26	22	26	26	27	26	35	23	33	29
IV(combined)	27	20	21	21	18	27	24	29	26	23
V(ours)	41	46	45	45	34	40	35	26	28	37

Fig. 1. The original score of user study.

user / method	user 101	user 102	user 103	user 104	user 105	user 106	user 107	user 108	user 109	user 110
I[17]	22	30	27	27	23	26	27	27	34	31
II[15]	15	16	23	10	27	18	34	34	10	12
III[16]	29	28	28	26	27	32	17	17	32	33
IV(combined)	22	20	28	24	31	19	33	33	19	21
V(ours)	42	36	24	43	22	35	19	19	35	33

user / method	user 111	user 112	user 113	user 114	user 115	user 116	user 117	user 118	user 119	user 120
I[17]	29	34	29	30	26	31	28	9	26	30
II[15]	17	13	20	6	16	11	15	32	24	8
III[16]	29	33	30	30	27	32	34	26	30	31
IV(combined)	22	18	24	26	34	18	15	35	26	21
V(ours)	33	32	27	38	27	38	38	28	24	40

user / method	user 121	user 122	user 123	user 124	user 125	user 126	user 127	user 128	user 129	user 130
I[17]	21	22	22	21	26	22	30	29	32	29
II[15]	25	23	23	21	17	8	20	20	13	5
III[16]	25	29	29	31	33	30	28	28	28	30
IV(combined)	28	25	25	28	18	26	24	24	24	26
V(ours)	31	31	31	29	36	44	28	29	33	40

user / method	user 131	user 132	user 133	user 134	user 135	user 136	user 137	user 138	user 139	user 140
I[17]	32	24	31	30	25	26	25	27	27	27
II[15]	7	7	10	13	13	29	23	28	28	28
III[16]	22	32	29	27	26	30	25	21	21	21
IV(combined)	29	23	22	23	34	21	20	26	26	26
V(ours)	40	44	38	37	32	24	37	28	28	28

user / method	user 141	user 142	user 143	user 144	user 145	user 146	user 147	user 148	user 149	user 150
I[17]	34	30	27	28	24	26	30	29	22	32
II[15]	7	14	19	26	19	10	22	12	11	7
III[16]	32	30	25	25	31	33	26	29	33	30
IV(combined)	21	19	24	22	21	23	28	25	30	19
V(ours)	36	37	35	29	35	38	24	35	34	42

user / method	user 151	user 152	user 153	user 154	user 155	user 156	user 157	user 158	user 159	user 160
I[17]	33	33	26	26	26	24	24	31	31	31
II[15]	14	20	25	25	25	29	28	8	25	25
III[16]	28	23	26	26	26	25	25	26	22	22
IV(combined)	19	25	27	27	27	25	25	23	29	29
V(ours)	36	29	26	26	26	27	28	42	23	23

user / method	user 161	user 162	user 163	user 164	user 165	user 166	user 167	user 168	user 169	user 170
I[17]	26	28	27	28	28	25	28	28	34	21
II[15]	21	19	9	18	18	7	19	19	29	26
III[16]	25	21	22	30	30	34	24	24	18	22
IV(combined)	26	31	30	18	18	26	30	30	20	33
V(ours)	32	31	42	36	36	38	29	29	29	28

user / method	user 171	user 172	user 173	user 174	user 175	user 176	user 177	user 178	user 179	user 180
I[17]	27	27	28	25	25	27	30	26	25	30
II[15]	16	16	21	24	24	23	23	26	24	23
III[16]	29	29	27	26	25	25	23	25	24	23
IV(combined)	25	29	21	25	25	26	28	30	26	27
V(ours)	33	29	33	30	31	29	26	23	31	27

user / method	user 181	user 182	user 183	user 184	user 185	user 186	user 187	user 188	user 189	user 190
I[17]	23	30	25	29	27	19	25	22	32	27
II[15]	23	8	24	8	20	22	24	13	9	12
III[16]	30	30	29	27	29	23	21	29	27	32
IV(combined)	33	23	22	23	26	32	28	27	22	26
V(ours)	21	39	30	43	28	34	32	39	40	33

user / method	user 191	user 192	user 193	user 194	user 195	user 196	user 197	user 198	user 199	user 200
I[17]	31	38	27	39	36	26	30	22	22	28
II[15]	10	13	16	7	21	21	13	20	20	9
III[16]	32	30	29	23	24	25	25	27	27	31
IV(combined)	21	21	22	26	25	32	22	29	29	26
V(ours)	36	28	36	35	24	26	40	32	32	36

Fig. 2. The original score of user study.