IMAGE COLORIZATION USING SPARSE REPRESENTATION

Jiahao Pang, Oscar C. Au, Ketan Tang and Yuanfang Guo

Department of Electronic and Computer Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong

ABSTRACT

Image colorization is the task to color a grayscale image with limited color cues. In this work, we present a novel method to perform image colorization using sparse representation. Our method first trains an over-complete dictionary in YUV color space. Then taking a grayscale image and a small subset of color pixels as inputs, our method colorizes overlapping image patches via sparse representation; it is achieved by seeking sparse representations of patches that are consistent with both the grayscale image and the color pixels. After that, we aggregate the colorized patches with weights to get an intermediate result. This process iterates until the image is properly colorized. Experimental results show that our method leads to high-quality colorizations with small number of given color pixels. To demonstrate one of the applications of the proposed method, we apply it to transfer the color of one image onto another to obtain a visually pleasing image.

Index Terms— colorization, sparse representation, color transfer, image restoration

1. INTRODUCTION

Image colorization, which is the process of adding color to a monochrome image, used to be a time-consuming and tedious task that requires tremendous user efforts. Recently, several effective colorization algorithms [1, 2, 3, 4] have been proposed to resolve this highly ill-posed problem with reasonable amount of user inputs. These algorithms receive inputs in form of either example images with similar color [2, 4] or scribbles that indicate colors of certain pixels [1, 3]; while their mechanisms to propagate chrominance vary a lot. For instance, the work by Levin *et al.* [1] optimizes a cost function based on the premise that adjacent pixels with similar intensities have similar colors. Another work by Yatziv *et al.* [3] colorizes images based on luminance-weighted chrominance blending and fast intrinsic distance computations.

Different from the previous works, in this paper we propose a novel and effective colorization technique based on sparse representation and dictionary learning. Sparse representation [5, 6] is a promising and powerful tool to model real-world signals, which suggests that the signals of interest live in a low-dimensional linear subspace defined by the combinations of atoms from the learned dictionary. Put it formally, for a class of signals $\Gamma \subset \mathbb{R}^n$, there exist a dictionary (a matrix) $\mathbf{D} \in \mathbb{R}^{n\times k}$ that contains k prototype signals (atoms); such that for any signal $\mathbf{x} \in \Gamma$, we can approximate it well with a sparse linear combination of atoms from $\mathbf{D}$. Mathematically, we have $\mathbf{x} \approx \mathbf{D}\alpha$ and $\|\alpha\|_0 \ll n$ where $\alpha \in \mathbb{R}^k$ is the sparse representation of $\mathbf{x}$ in terms of dictionary $\mathbf{D}$, and $\|\cdot\|_0$ counts the number of nonzero elements (cardinality) in a vector [7]. Although finding the sparsest representation and the dictionary is generally intractable, recent progresses show that approximation algorithms works quite well in practice [6]. Besides, sparse representation achieves state-of-the-art results in image processing applications like denoising, inpainting and demosaicing [7, 8], which encourages us to explore its potential in image colorization.

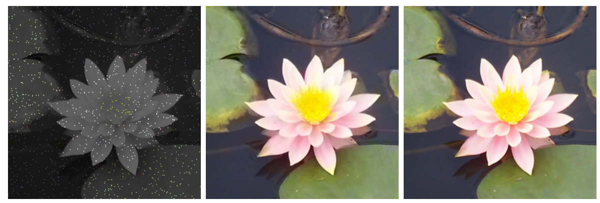


Fig. 1. Given a grayscale image with randomly scattered color pixels (left), our method colorize it faithfully (middle) compared to the original color image (right).

Our method works in the YUV color space, where the Y component (luminance) of the image is given by the user; the algorithm also takes as input a small subset of scattered pixels which indicate the desired U, V values (chrominance). Although other color spaces (e.g., YCbCr and Lab) may also be used in our work under the same rationale, we choose YUV color space which is the same as [1] so that comparisons with [1] are fairer. As a preparation, we first train an over-complete dictionary in YUV space offline, which uses image patches from a generic image database as training examples. After that, our method colorizes overlapping image patches by seeking for sparse representations of patches that are consistent with both the Y component and the known U, V values. We then aggregate the colorized patches with different confidences to form an intermediate colorized image. We iterate this process until all pixels are properly colorized. Fig. 1 presents an example of colorization using the proposed method, where we randomly remove 98% of the chrominance from the image, and colorize it to restore its U, V components. Note that in Fig. 1 we darken the given grayscale image and enhance the color pixels for better display. We recommend to read the electronic version of this paper for the best views of the figures.

2. YUV DICTIONARY LEARNING

To facilitate the colorization algorithm, we first train an over-complete dictionary offline, which is a common dictionary being used by all input images. This dictionary, denoted by $\mathbf{D}^{(\mathrm{YUV})}$, is a YUV dictionary and $\mathbf{D}^{(\mathrm{YUV})} = \left[\mathbf{D}^{(\mathrm{Y})^{\mathrm{T}}}\,\mathbf{D}^{(\mathrm{U})^{\mathrm{T}}}\,\mathbf{D}^{(\mathrm{V})^{\mathrm{T}}}\right]^{\mathrm{T}} \in \mathbb{R}^{3n\times k}$, where $\mathbf{D}^{(\mathrm{Y})}, \mathbf{D}^{(\mathrm{U})}, \mathbf{D}^{(\mathrm{V})} \in \mathbb{R}^{n\times k}$ are sub-dictionaries for the Y, U and V color components, respectively. Here we denote n as the length of their atoms, which is also the length of the vectorized

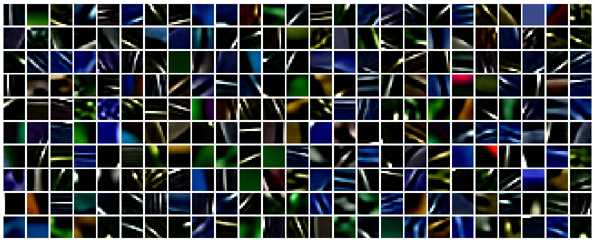


Fig. 2. Some atoms of the YUV color dictionary learned over the BSDS500 generic image database.

image patch; and k is the number of atoms in the dictionary.

To train the dictionary, we gather numerous sample image patches randomly from a generic color image database and convert all patches into YUV color space. After that, to make the learned dictionary less sensitive to luminance changes, we subtract the mean from the Y component of each patch such that the Y components of all patches have zero means. In other words, a pre-processed sample patch is a vector $\mathbf{s}^{(\mathrm{YUV})} = \left[\mathbf{s}^{(\mathrm{Y})\mathrm{T}}\ \mathbf{s}^{(\mathrm{U})\mathrm{T}}\ \mathbf{s}^{(\mathrm{V})\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3n}$, where $\mathbf{s}^{(\mathrm{Y})}, \mathbf{s}^{(\mathrm{U})}, \mathbf{s}^{(\mathrm{V})} \in \mathbb{R}^n$ are the vectorized Y, U and V components of that patch, and particularly, $\mathbf{s}^{(\mathrm{Y})}$ has zero mean. With all the pre-processed patches as training samples, we train the YUV color dictionary $\mathbf{D}^{(\mathrm{YUV})}$ using the online dictionary learning algorithm proposed in [9]. Note that in this work, we use patches of sizes 12×12, namely $n = 144$, and the dictionary has $k = 625$ atoms. We adopt the BSDS500 database [10] as the generic image database to extract the sample patches. Fig. 2 shows 250 atoms of the YUV color dictionary trained over 10^6 color patches, where we display the atoms in the same way as that of [7].

3. PATCH-BASED COLORIZATION USING SPARSE REPRESENTATION

Our method colorizes overlapping patches in the grayscale image based on the given scattered color pixels. Equipped with the learned dictionary, this section elaborates our algorithm to colorize an image patch using sparse representation.

3.1. Notations

To aid in the presentation, we first introduce some notations. We denote the given grayscale image (Y component, luminance) as a column vector $\mathbf{X}^{(\mathrm{Y})} \in \mathbb{R}^N$, where N is the number of pixels in the image. Besides, we define a binary indicator $\mathbf{B} \in \mathbb{R}^N$ such that an entry in $\mathbf{B}$ corresponds to a pixel with known color is set to be 1 or otherwise 0. We then define two vectors containing the given U, V components (chrominance) as $\mathbf{X}^{(\mathrm{U})}, \mathbf{X}^{(\mathrm{V})} \in \mathbb{R}^N$, such that for an entry in $\mathbf{X}^{(\mathrm{U})}$ (or $\mathbf{X}^{(\mathrm{V})}$) corresponds to a pixel with known color, its value is the given U (or V) value, otherwise it is 0. We also define the binary matrices $\mathbf{R}_{ij} \in \mathbb{R}^{n \times N}$ that extracts the $\sqrt{n} \times \sqrt{n}$ patch at (i, j) from the N-by-1 vectorized image, where the index (i, j) is the coordinate of the top-left corner of the patch.

With the patch extractors $\mathbf{R}_{ij}$, image patches of the Y, U and V components can be written as n-by-1 vectors. The (i, j)-th patch of the grayscale image with mean subtracted is $\mathbf{x}_{ij}^{(\mathrm{Y})} = \mathbf{R}_{ij}\mathbf{X}^{(\mathrm{Y})} - \mu_{ij}^{(\mathrm{Y})}\mathbf{1}_n$, where $\mu_{ij}^{(\mathrm{Y})}$ is the mean of $\mathbf{R}_{ij}\mathbf{X}^{(\mathrm{Y})}$, $\mathbf{1}_n$ is the n-by-1 vector with ones. The (i, j)-th patch of the U and V components are computed as $\mathbf{x}_{ij}^{(\mathrm{U})} = \mathbf{R}_{ij}\mathbf{X}^{(\mathrm{U})}$ and $\mathbf{x}_{ij}^{(\mathrm{V})} = \mathbf{R}_{ij}\mathbf{X}^{(\mathrm{V})}$, respectively.

Using the patch extractors $\mathbf{R}_{ij}$, we denote the binary indicator of the (i, j)-th patch as an n-by-1 vector $\beta_{ij} = \mathbf{R}_{ij}\mathbf{B}$; and $m_{ij} = \|\beta_{ij}\|_0$ as the number of given color pixels in the (i, j)-th patch.

3.2. Image Patch Colorization

Based on the luminance (Y) and the given chrominance (U and V) within the patch, we consider colorizing the (i, j)-th patch using sparse representation. Denote $\mathbf{x}_{ij}^{(\mathrm{UV})} = \left[\mathbf{x}_{ij}^{(\mathrm{U})\mathrm{T}}\ \mathbf{x}_{ij}^{(\mathrm{V})\mathrm{T}}\right]^{\mathrm{T}}$, $\beta_{ij}^{(\mathrm{UV})} = \left[\beta_{ij}^{\mathrm{T}}\ \beta_{ij}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{D}^{(\mathrm{UV})} = \left[\mathbf{D}^{(\mathrm{U})\mathrm{T}}\ \mathbf{D}^{(\mathrm{V})\mathrm{T}}\right]^{\mathrm{T}}$, and $\mathrm{diag}(\cdot)$ as a diagonal matrix formed from its vector argument. Then ideally, our aim is to find the sparse representation $\alpha_{ij} \in \mathbb{R}^k$ with minimal l_0-norm, such that

$$\mathbf{D}^{(\mathrm{Y})}\alpha_{ij} \approx \mathbf{x}_{ij}^{(\mathrm{Y})} \qquad \text{(luminance consistency)},$$

$$\mathrm{diag}(\beta_{ij}^{(\mathrm{UV})})\,\mathbf{D}^{(\mathrm{UV})}\alpha_{ij} \approx \mathrm{diag}(\beta_{ij}^{(\mathrm{UV})})\,\mathbf{x}_{ij}^{(\mathrm{UV})} \quad \text{(chromin. consistency)}.$$

Here the indicator $\beta_{ij}^{(\mathrm{UV})}$ works as a mask to eliminate the effects of the irrelevant pixels. In a word, we seek for a sparse representation α_{ij} that is consistent with both the luminance and the known chrominance within the (i, j)-th patch. Notice that here we implicitly assume, the sparse representation of the patch with $\mathbf{D}^{(\mathrm{Y})}$ is the same as that with $\mathbf{D}^{(\mathrm{UV})}$, which is a reasonable assumption for well-behaved YUV dictionary $\mathbf{D}^{(\mathrm{YUV})}$ that learned from natural color images.

A more concrete formulation of the image patch colorization problem stated above can be written as:

$$\hat{\alpha_{ij}} = \arg\min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \quad \text{subject to} \quad \left\|\mathbf{D}^{(\mathrm{Y})}\alpha_{ij} - \mathbf{x}_{ij}^{(\mathrm{Y})}\right\|_2^2 + \gamma\left\|\mathrm{diag}(\beta_{ij}^{(\mathrm{UV})})\,\mathbf{D}^{(\mathrm{UV})}\alpha_{ij} - \mathrm{diag}(\beta_{ij}^{(\mathrm{UV})})\,\mathbf{x}_{ij}^{(\mathrm{UV})}\right\|_2^2 \le \epsilon, \tag{1}$$

where coefficient γ controls the tradeoff between luminance consistency and chrominance consistency, and ϵ is the error tolerance.

By denoting $\mathbf{W}_{ij}(\sqrt{\gamma}) = \mathrm{diag}\left(\left[\mathbf{1}_n{}^{\mathrm{T}}\ \sqrt{\gamma}\beta_{ij}^{(\mathrm{UV})\mathrm{T}}\right]^{\mathrm{T}}\right)$ and $\mathbf{x}_{ij}^{(\mathrm{YUV})} = \left[\mathbf{x}_{ij}^{(\mathrm{Y})\mathrm{T}}\ \mathbf{x}_{ij}^{(\mathrm{UV})\mathrm{T}}\right]^{\mathrm{T}}$, we combine the two terms in the constraint of (1) and rewrite the problem into a tractable form,

$$\hat{\alpha_{ij}} = \arg\min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \quad \text{subject to} \quad \left\|\mathbf{W}_{ij}(\sqrt{\gamma})\,\mathbf{D}^{(\mathrm{YUV})}\,\alpha_{ij} - \mathbf{W}_{ij}(\sqrt{\gamma})\,\mathbf{x}_{ij}^{(\mathrm{YUV})}\right\|_2^2 \le \epsilon. \tag{2}$$

Though the l_0-norm minimization problem (2) is NP-hard, its solution can be approximated using either greedy algorithms (e.g., orthogonal matching pursuit) or convex relaxation techniques (e.g., basis pursuit) [6]. In this work, we employ orthogonal matching pursuit (OMP) like that in [7, 8] for its simplicity and efficiency.

With the solved $\hat{\alpha_{ij}}$, the colorized patch can be obtained as

$$\hat{\mathbf{x}_{ij}^{(\mathrm{U})}} = \mathbf{D}^{(\mathrm{U})}\hat{\alpha_{ij}}, \quad \hat{\mathbf{x}_{ij}^{(\mathrm{V})}} = \mathbf{D}^{(\mathrm{V})}\hat{\alpha_{ij}}, \tag{3}$$

in which $\hat{\mathbf{x}_{ij}^{(\mathrm{U})}}, \hat{\mathbf{x}_{ij}^{(\mathrm{V})}} \in \mathbb{R}^n$ are the recovered U and V components of the (i, j)-th patch, respectively.

The coefficient γ in (1) and (2) is a critical parameter that weights the importance of the known color pixels. Our experiments reveal that, a small γ permits the structure in the luminance $\mathbf{x}_{ij}^{(\mathrm{Y})}$ to guide the propagation of the chrominance, which benefits the colorizations. On the contrary, a big γ drives the colorizations look blur because too much emphasis is placed onto recovery of the known

Algorithm 1 Image colorization using sparse representation

INPUT: $\mathbf{X}^{(\mathrm{Y})}$, $\mathbf{X}_0^{(\mathrm{U})}$, $\mathbf{X}_0^{(\mathrm{V})}$ and $\mathbf{B}_0$
OUTPUT: $\mathbf{X}_t^{(\mathrm{U})}$ and $\mathbf{X}_t^{(\mathrm{V})}$

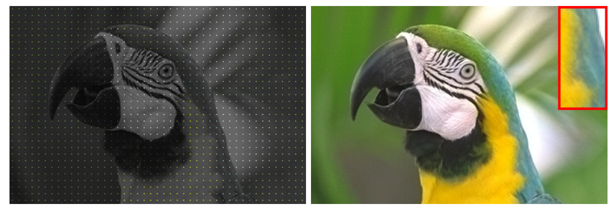
$t \leftarrow 0$;
while $\|\mathbf{B}_t\|_0 \neq N$ **do**
  $\mathbf{X}_{t+1}^{\hat{(\mathrm{U})}} \leftarrow \mathbf{0}$, $\mathbf{X}_{t+1}^{\hat{(\mathrm{V})}} \leftarrow \mathbf{0}$;
  for all (i,j) that defines an image patch with $0 < m_{ij} < n$ **do**
    Take $\mathbf{X}^{(\mathrm{Y})}$, $\mathbf{X}_t^{(\mathrm{U})}$, $\mathbf{X}_t^{(\mathrm{V})}$ and $\mathbf{B}_t$ as inputs, perform patch-based colorization described in Section 3 to recover $\mathbf{x}_{ij}^{\hat{(\mathrm{U})}}$ and $\mathbf{x}_{ij}^{\hat{(\mathrm{V})}}$;
    Confidence update: $\mathbf{C}_{t+1} \leftarrow \mathbf{C}_{t+1} + m_{ij}\mathbf{R}_{ij}^{\mathrm{T}}\mathbf{1}_n$;
    Colorized patches aggregation:
    $$\mathbf{X}_{t+1}^{\hat{(\mathrm{U})}} \leftarrow \mathbf{X}_{t+1}^{\hat{(\mathrm{U})}} + m_{ij}\mathbf{R}_{ij}^{\mathrm{T}}\mathbf{x}_{ij}^{\hat{(\mathrm{U})}},$$
    $$\mathbf{X}_{t+1}^{\hat{(\mathrm{V})}} \leftarrow \mathbf{X}_{t+1}^{\hat{(\mathrm{V})}} + m_{ij}\mathbf{R}_{ij}^{\mathrm{T}}\mathbf{x}_{ij}^{\hat{(\mathrm{V})}};$$
  end for
  for $i = 1$ to N **do**
    Binary indicator update:
    $$\mathbf{B}_{t+1}(i) \leftarrow \begin{cases} 1 & \mathbf{C}_{t+1}(i) \geq d \text{ or } \mathbf{B}_t(i) = 1, \\ 0 & \text{otherwise;} \end{cases}$$
    Colorization update:
    $$\mathbf{X}_{t+1}^{(\Omega)}(i) \leftarrow \begin{cases} \mathbf{X}_{t+1}^{\hat{(\Omega)}}(i)/\mathbf{C}_{t+1}(i) & \mathbf{B}_{t+1}(i) = 1, \\ 0 & \text{otherwise,} \end{cases}$$
    where $\Omega \in \{\mathrm{U}, \mathrm{V}\}$;
  end for
  Rectify $\mathbf{X}_{t+1}^{(\mathrm{U})}$ and $\mathbf{X}_{t+1}^{(\mathrm{V})}$ globally based on the known chrominance in $\mathbf{X}_t^{(\mathrm{U})}$ and $\mathbf{X}_t^{(\mathrm{V})}$;
  $t \leftarrow t + 1$;
end while

colors so that the details in the luminance is overlooked. As a result, we empirically set γ to be a small value 0.05. Besides, we also fix $\epsilon = 10^{-8}$ when solving (2). Having observed that the contrast and hue of $\mathbf{x}_{ij}^{\hat{(\mathrm{U})}}$ and $\mathbf{x}_{ij}^{\hat{(\mathrm{V})}}$ could be incorrect when γ is small, we carry out a simple procedure to rectify them. When m_{ij} is no less than a predefined value $l = 4$, we amplify $\mathbf{x}_{ij}^{\hat{(\mathrm{U})}}$ by a scalar then shift its mean by solving a least-square problem based on the given U components in the patch; when $m_{ij} < l$, we simply shift the mean of $\mathbf{x}_{ij}^{\hat{(\mathrm{U})}}$ such that $\mathrm{diag}(\beta_{ij})\,\mathbf{x}_{ij}^{\hat{(\mathrm{U})}}$ has the same mean as $\mathrm{diag}(\beta_{ij})\,\mathbf{x}_{ij}^{(\mathrm{U})}$. The same procedure is applied to rectify the recovered V component $\mathbf{x}_{ij}^{\hat{(\mathrm{V})}}$.

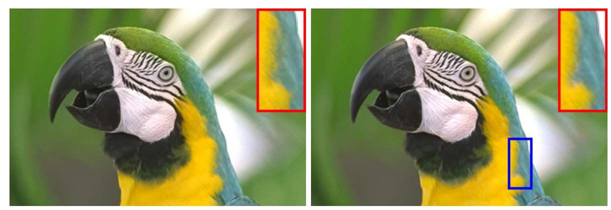
4. CONFIDENCE-BASED AGGREGATION

Our method colorizes the given image in an iterative manner so that the known chrominance can be propagated to the whole image. In every iteration, we apply the patch-based colorization algorithm presented in Section 3 to all the overlapping patches. After that, we combine them with proper weights to form the colorized image of that iteration. To describe the colorization confidence of each pixel, we define a confidence map $\mathbf{C} \in \mathbb{R}^N$. The aggregated U and V components of the image are denoted as two N-by-1 column vectors, $\mathbf{X}^{\hat{(\mathrm{U})}}$ and $\mathbf{X}^{\hat{(\mathrm{V})}}$; and the number of iteration is denoted as t. The proposed image colorization method is summarized in Algorithm 1.

Fig. 3. Ten test images from the Kodak image database. With the order of left to right then top to bottom, they are indexed as image 1, image 2, and so forth.



(a) Luminance with color pixels (b) Result of the proposed method



(c) Result of method [1] (d) Original color region of image 3

Fig. 4. Colorizations of a region within image 3 in Fig. 3. Note the transition between the yellow and cyan feather in the colorizations.

In Algorithm 1, we set the weight of a colorized patch, say the (i,j)-th patch, to be proportional to m_{ij} and update the confidence map correspondingly. We also update the binary indicator $\mathbf{B}$ in every iteration according to the confidence $\mathbf{C}$ and a predefined threshold d. Consequently for a particular pixel, if it is colorized for less than d times within an iteration, we regard its aggregated color as untrustworthy and its color is marked as unknown in the next iteration. In this work, we empirically set $d = 3$ for a reasonable trade-off between complexity and colorization quality. Note that similar to the post-processing step in Section 3.2, at iteration t we solve a least-square problem and rectify $\mathbf{X}_{t+1}^{(\mathrm{U})}$ based on the known U components in $\mathbf{X}_t^{(\mathrm{U})}$. This simple chrominance correction procedure is applied onto $\mathbf{X}_{t+1}^{(\mathrm{V})}$ as well.

5. EXPERIMENTAL RESULTS

In this section, we demonstrate our colorization results and compare them with those obtained by [1]. After that, we present an application of our method in color transfer, where the colors of a bright but blurred image are transferred onto another sharp but dim image.

Fig. 3 shows 10 test images selected from the Kodak PhotoCD [11], which are all of sizes 512×768. For every image, we uniformly sample 1 color pixel out of every 8×8 non-overlapping patch, therefore about 98.4% of chrominance are removed. This sampling scheme, as shown in Fig. 4(a), may not be optimal, we choose this typical sampling to evaluate our algorithm for its simplicity and representativeness. After that, we colorize the images using the pro-

Table 1. CPSNR qualities for the 10 test images (in dB)

Image	1	2	3	4	5
Result of [1]	36.29	38.79	35.85	35.19	35.32
Proposed method	**36.94**	**39.28**	**36.30**	**35.50**	**35.61**
Image	6	7	8	9	10
Result of [1]	39.15	33.45	**35.88**	**38.18**	**38.76**
Proposed method	**39.40**	**33.63**	35.87	38.12	38.34

posed method and the method in [1] using their provided implementation with the exact solver. To quantitatively assess the colorization quality, we adopt the color peak signal-to-noise ratio (CPSNR) as a measurement, which is defined as follows: for two color images I_1 and I_2 of the same sizes $H \times W$, the CPSNR is computed as

$$\mathrm{CPSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}},$$

$$\mathrm{MSE} = \frac{1}{3HW}\sum_{\Omega\in\{\mathrm{R,G,B}\}}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\left(I_1^{(\Omega)}(i,j) - I_2^{(\Omega)}(i,j)\right)^2,$$

where $I_1^{(\mathrm{R})}(i,j)$, $I_1^{(\mathrm{G})}(i,j)$, and $I_1^{(\mathrm{B})}(i,j)$ denote the R, G and B color components of the pixel locates at (i,j) in the color image I_1; which is similar for I_2. We compute the CPSNR values between the colorizations and their corresponding ground-truth color images, then tabulate the results in Table 1. It reveals that, with scattered color cues, our method gives superior results in terms of CPSNR.

Fig. 4 presents the colorizations of a region within image 3 of Fig. 3. Note that Fig. 4(a) is displayed in the same way as that of the left image in Fig. 1. The colorization of our method looks vivid and comparable to the original color image. Besides, notice the transition between the yellow feather and cyan feather of the parrot. Our colorization gives fluffy and faithful transition, while the result of [1] has too sharp and cartoon-like transition.

The proposed colorization method has several potential applications. For instance in lossy color image compression, instead of encoding all the color information of an image, the encoder removes most of the chrominance and store the colors of a few scattered pixels; then the decoder applies the proposed method to recover the chrominance. By this means, the image codec can achieve better compression ratios while keeping low distortions.

In Fig. 5, we present another application, that is, color transfer. Consider an image pair of the same scene taken in dark environment, one image is sharp but dim due to short exposure (Fig. 5(a)), while the other one is bright but blurred due to long exposure (Fig. 5(b)). We apply the proposed colorization method to recolor the sharp but dim image (denoted as I_{sd}) with color cues transferred from the bright but blurred image (denoted as I_{bb}), such that the resulting image looks bright and contains fine details. We first enhance the luminance of I_{sd} then use image registration technique (SIFT and RANSAC [12]) to align I_{bb} and I_{sd}. Since I_{bb} is well-exposed and contains correct color characteristics, we select a few scattered pixels from I_{sd} and seek for their corresponding colors in I_{bb} according to the alignment. After that, we denoise the luminance of I_{sd}, and apply the proposed method to recolor it with the color pixels transferred from I_{bb}. As a result, we combine the luminance of I_{sd} and chrominance of I_{bb} to obtain a visually pleasing image (Fig. 5(c)). To select the color pixels from I_{sd}, we use the same uniform sampling scheme as that in Fig. 4(a). Of course, rather than using image registration, one may consider applying example-based colorization methods (e.g., [2]) to transfer color cues to I_{sd} then proceeds by our algorithm, such that one may achieve better colorizations.

(a) (b) (c)

Fig. 5. Color transfer using the proposed method. (a) Sharp but dim image. (b) Bright but blurred image. (c) Color transfer result.

6. RELATION TO PRIOR WORK

Our work is partially inspired by Mairal *et al.*'s work [7], where they seek for sparse representations of image patches that are consistent with the given noisy patches. The work [7] achieves state-of-the-art performance in denoising, inpainting and demosaicing, but it is not capable of coloring images. Actually, their method stems from the K-SVD-based denoising algorithm [8], which limits its usage. Moreover, [7] is specifically tailored to work in RGB color space and it treats all the color channels equally. Different from that, we work in YUV color space and differentiate the importance of chrominance from luminance, which makes our formulation different from that of [7]. Besides, we also develop an effective scheme to guide the colors to propagate across the whole image iteratively.

On the other hand, our work is also closely related to Levin *et al.*'s work [1]. By assuming neighboring pixels with similar intensities should have similar colors, Levin *et al.* treat image colorization as an optimization problem and obtain photo-realistic colorizations; while our method endeavors to represent the image patches with a small amount of learned color atoms. Note that for a color atom in our YUV dictionary, adjacent pixels having similar intensities also have similar colors (Fig. 2) because the atoms are learned from natural color images. Therefore in our method, the assumption used in [1] is automatically learned during the dictionary learning process (Section 2). In other words, Levin *et al.*'s assumption is implicitly satisfied in our work, though our approach appears to be very different from theirs [1].

7. CONCLUSION

In this work, we tackle the problem of image colorization from the aspect of sparse representation. We first train an over-complete dictionary in YUV color space. Taking a grayscale image and a small subset of color pixels as inputs, we colorize overlapping patches by seeking for sparse representations that are consistent with both the luminance and the known chrominance. The intermediate colorized image is then obtained by aggregating the colorized patches with weights. This colorization process iterates until the colors are properly propagated across the entire image. Experiments and comparisons show that our method leads to high-quality colorizations in terms of CPSNR. To demonstrate the applications of our method, we use it to transfer the color of one image onto another to obtain a new image. Future improvements may focus on extending the current algorithm such that it also accepts scribble-based color cues.

8. REFERENCES

[1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Conf.*, 2004, vol. 23, pp. 689–694.

[2] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. 16th Eurograph. Symp. Rendering*, 2005, pp. 201–210.

[3] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. on Imag. Process.*, vol. 15, no. 5, pp. 1120–1129, 2006.

[4] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 152:1–152:9, 2008.

[5] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[6] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*, Springer, 2010.

[7] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Imag. Process.*, vol. 17, no. 1, pp. 53–69, 2008.

[8] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Imag. Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[9] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[11] R. Franzen, "Kodak lossless true color image suite," `http://r0k.us/graphics/kodak/`, 2012.

[12] M. Brown and D. G. Lowe, "Recognising panoramas," in *Proc. Int. Conf. Comp. Vis.*, 2003, vol. 2, pp. 1218–1225.