Neurocomputing 352 (2019) 22–32

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Shoot high-quality color images using dual-lens system with monochrome and color cameras

Xuan Dong [a], Weixin Li [b,*]

[a] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
[b] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China

ARTICLE INFO

Article history:
Received 21 December 2018
Revised 2 April 2019
Accepted 3 April 2019
Available online 25 April 2019

Communicated by Xianbin Cao

Keywords:
Gray-color correspondence prior
Color similarity network
Coarse-to-fine colorization network
Symmetry colorization based evaluation

ABSTRACT

In the dual-lens system with monochrome and color cameras, the gray image captured by the monochrome camera has better quality than the color image from the color camera, but does not have color information. To get high-quality color images, it is desired to colorize the gray image with the color image as reference. Due to occlusions, the colorization will inevitably fail in some cases. Thus, evaluating the colorization quality is also of great importance. We solve both problems in this paper. For colorization, we propose a gray-color correspondence prior, i.e. in local regions, if two patches are similar in the gray channel, it is very often that the two pixels centered at these two patches have similar colors. Based on this prior, a deep learning based and coarse-to-fine colorization method is proposed. For evaluating the colorization quality, we propose a symmetry colorization based evaluation method. Experimental results show that our method could largely outperform the state-of-the-art methods and is also efficient in computation.

1. Introduction

The dual-lens system with one monochrome camera and one color camera has been widely used in popular smart phones, e.g. Huawei P9 and P10. In the dual-lens system, the monochrome camera has better light efficiency than the color camera [1,2], so the image captured by the monochrome camera has higher quality (i.e. signal-to-noise ratio) than the image from the color camera, but does not have color information. To shoot high quality color images using dual-lens systems, it is desirable to colorize the gray images from the monochrome camera with the color images from the color camera as reference, so that the colorized images have high quality in the monochrome channel and correct colors as well. An example is shown in Fig. 1. However, due to occlusions between the pair of images, the colorization could not have correct results all the time. In these cases, the input color image from the color camera (with lower quality in the monochrome channel but correct colors) could be used as an alternative choice for the output color image. So, it is also desired to evaluate the colorization quality to make the choice. In this paper, we deal with both problems, i.e. colorization in the dual-lens system and colorization quality evaluation.

In the literature, the stereo matching based colorization method [1] is a straightforward solution for dual-lens colorization. However, we observe that there are big differences between estimating correct disparity values and colorization. For a given pixel, there may exist multiple pixels in the reference image that could provide correct color values, especially in repeated texture regions and textless regions (e.g. blue sky, white wall, etc.). Some examples are shown in Fig. 3. Because any of these pixel could help us obtain correct colorization result, searching for the pixels with correct disparity values using the computation consuming stereo matching methods is unnecessary. Thus, we propose a new method to do the colorization without estimating disparities of pixels between the pair of images.

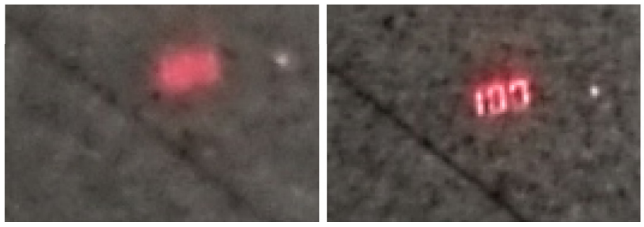
Our insight is the observation that in local regions of images, if two patches are similar in the gray channel, it is very often that the two pixels centered at these two patches have similar colors, no matter whether the centered pixels have correct disparity. We name this statistics-based property as the gray-color correspondence prior in this paper. This prior inspires us that, for each patch in the input gray image, any similar patches in the reference image, no matter whether with the correct disparity or not, can provide correct color for colorization. So the computationally costly full stereo matching can be omitted.

Motivated by the prior, for each patch in the input image, we propose a convolutional neural network, called color similarity network, to search for patches with correct colors in the reference

* Corresponding author.
E-mail address: weixinli@buaa.edu.cn (W. Li).

https://doi.org/10.1016/j.neucom.2019.04.007
0925-2312/

(a) The input pair of gray and color images. (b) The output color image.



(c) The input and output color images in the red box region.

Fig. 1. An example of the colorization in the dual-lens system. The input images are captured by the dual-lens system of Huawei P9 phone. And the colorization result, which has high quality in the monochrome channel and correct colors, is used as the output color image of the dual-lens system.

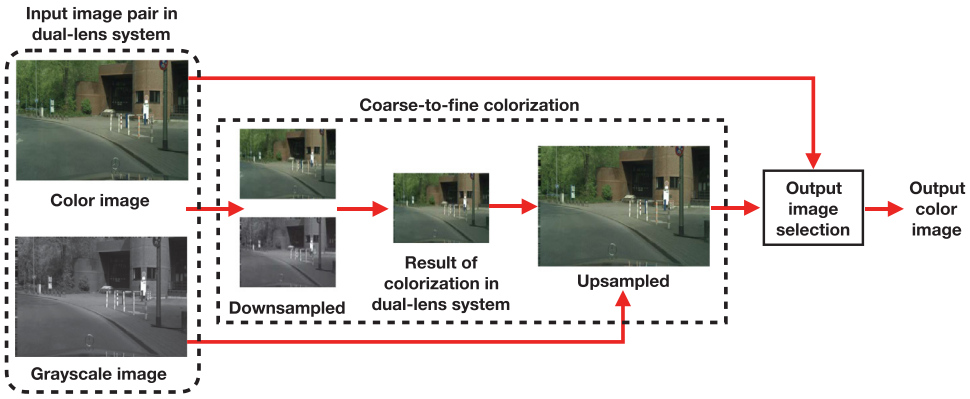


Fig. 2. Pipeline of our work. First, we propose a coarse-to-fine colorization method, which down-samples the input pair, performs the deep learning based colorization for the low resolution pair, and upsamples the colorization result using the original input gray image as guidance. Second, from the colorization result and the input color image captured by the color camera, we select which should be output by evaluating the colorization quality. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

image. We also use a color propagation method to colorize occluded pixels where no patches with correct colors exist in the reference image. In addition, we propose a convolutional neural network, called coarse-to-fine colorization network, to perform coarse-to-fine colorization for further acceleration. The whole system uses the color similarity network and color propagation method to perform the reference-based colorization at the low resolution and then uses the coarse-to-fine colorization network to upsample the coarse colorization result with the original input gray image as guidance.

To evaluate the colorization quality, we propose a symmetry colorization based evaluation method, according to our observation of the symmetry property of colorization. This enables us to select the output color image in the dual-lens system from either the colorized image or the original color image from the color camera. The pipeline of the proposed algorithm in this paper is shown in Fig. 2.

Experimental results show that our method could largely outperform the state-of-the-art algorithms and is also efficient in computation.

Our contributions include: (1) we propose the gray-color correspondence prior for monochrome-color dual-lens colorization. (2) We propose the color similarity network for reference-based colorization. (3) We propose the coarse-to-fine colorization network for accelerating the colorization. And (4) we propose the symmetry colorization based method for evaluating the colorization quality.

2. Related work

In the literature, there exist three kinds of colorization algorithms, including automatic colorization, scribble-based colorization, and reference-based colorization. Automatic colorization algorithms [3,4] directly colorize gray images without any reference. Using them in our case is not proper because the reference color image, which provides much useful color information, will not be utilized. Scribble-based colorization algorithms [5,6] need users to input some scribbles as guidance for colorization. Because user input is not available in the camera system, these algorithms are not suitable for our case too. Reference-based colorization [1,7–10] is related to our problem. But most of them have different assumptions from our problem. Between the input and reference images, the methods in [7–9] only assume that a small part of contents or even no content are the same, and the images usually

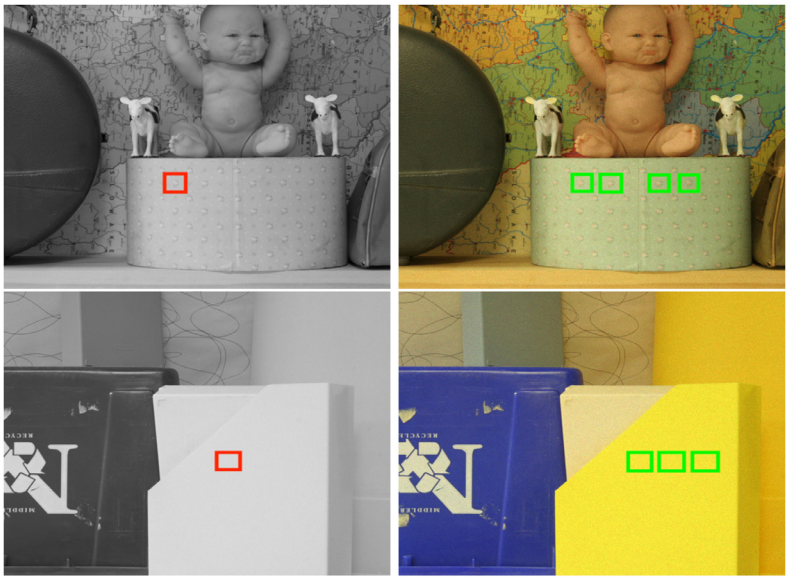


Fig. 3. Examples to show there usually exist several similar pixels (marked in green) in the reference image that could provide correct colors for a given pixel (marked in red) in the input gray image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

just share similar 'mood'. The method in [10] assumes that the images are manga images but in our problem the images are general images. Due to different assumptions of the input pairs of images, these methods cannot obtain competing results for our problem. A straightforward solution for colorization in the dual-lens system is to estimate the disparity between the pair of images, and then colorize the gray image using the estimated disparities and the color image, as is done in [1]. But, as mentioned in Section 1, pixels that could provide correct colors may not have correct disparity values. And it is not necessary to estimate disparity for our problem. We compare with these methods and analyze their weakness respectively in more details in Section 6.

Our gray-color correspondence prior shares similar insights with self-similarity based super resolution [11] and non-local based image denoising [12,13], i.e. the fractal nature of images [14] suggests that patches of a natural image recur in the same image. So, the enhancement of a patch could benefit from all of its similar patches. Similarly, in our problem, based on the prior, all the patches with correct colors in the reference image, no matter with correct disparity or not, can help the colorization.

To select the output color image, automatically evaluating the colorization quality in the dual-lens system is of great importance. But there is no related work in the literature to the best of our knowledge.

3. The gray-color correspondence prior

The dual-lens system of the smart phone is similar with stereo system [15], where for each pixel in the input gray image, its corresponding pixel in the reference image is with the same vertical position but different horizontal position due to disparity. So, in our problem, for each patch in the input gray image, it has high probability that the patches with correct color are with the same vertical position but different horizontal position. Thus, when we search for similar patches in the reference image for each patch in the input gray image, the search range is defined as the patches with the same vertical positions and the position differences in the horizontal position is from 0 to $d-1$. The constant value d is the maximum position difference in the horizontal direction (d is set as 30% of the image width). This motivates us to explore the property of the pixels in the search range.

We propose the gray-color correspondence prior that, in the search range, if two patches are similar in the gray channel, it is very often that the two pixels centered at these two patches have similar colors. An example is shown in Fig. 3.

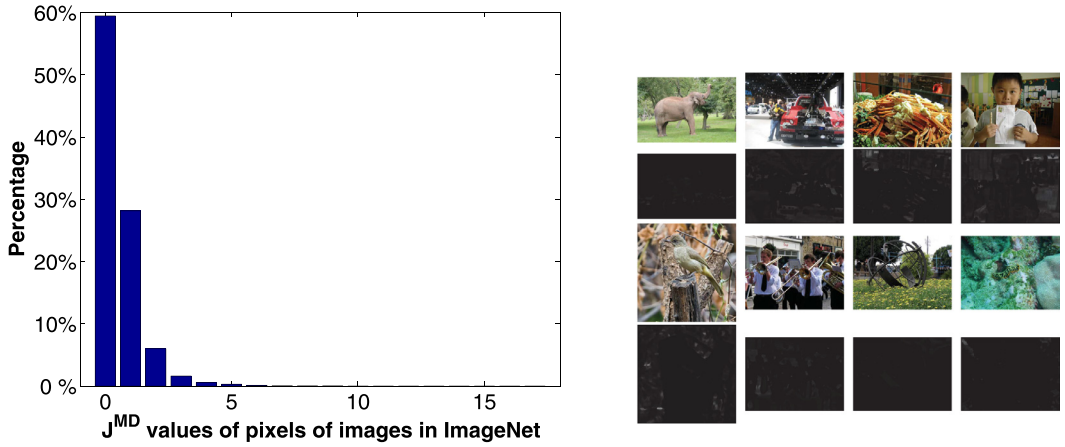
To formally describe the observation, we define

$$J^{MD}(\mathbf{x}) = \max_{\mathbf{y}\in S(\mathbf{x})} ||\mathbf{C_x} - \mathbf{C_y}||_2, \tag{1}$$

where $J^{MD}(\mathbf{x})$ measures the maximum color differences of the centered pixels between patch $\mathbf{x}$ and all the patches in $S(\mathbf{x})$. The colors of the centered pixels of patch $\mathbf{x}$ and $\mathbf{y}$ are $\mathbf{C_x}$ and $\mathbf{C_y}$, respectively. $\mathbf{x}$ is a patch in the input gray image and within the search range in the reference image corresponding to patch $\mathbf{x}$, all patches that are similar with $\mathbf{x}$ in the gray channel consist of the set $S(\mathbf{x})$. Formally, for any patch $\mathbf{y}$ in $S(\mathbf{x})$, $\frac{||Y_\mathbf{x}-Y_\mathbf{y}||_2}{|\Omega(\mathbf{x})|} < \varepsilon$, where $|\Omega(\mathbf{x})|$ is the number of pixels in the patch $\mathbf{x}$, and the patch size is 30×30 in this paper. $Y_\mathbf{x}$ represents the gray intensities of pixels in the patch $\mathbf{x}$ in a vector, and ε is a small constant value. According to our observation, for any patch that belongs to $S(\mathbf{x})$, the colors of their centered pixels should be similar, so $J^{MD}(\mathbf{x})$ tends to be zero.

The reasons that the prior is effective in most cases are as follows. (1) If two objects with different colors are under different lighting conditions, because the lighting conditions are different, patches from the two objects will mostly have different gray intensities. (2) Assuming that two objects with different colors are under the same lighting condition. According to the physics of color [17], different colors are caused by different wavelengths of electromagnetic radiation and different wavelengths will result in different intensities of the electromagnetic radiation. So, if two patches are from the two objects that have different colors and are under the same lighting condition, their gray intensities will be different.

To verify how good our prior is, we test it using 10,000 randomly selected images in the ImageNet dataset [16]. Using Eq. (1), we compute the J^{MD} values of all pixels in the images. And the distribution of J^{MD} values is shown in Fig. 4(a). Some example images and their J^{MD} maps are shown in Fig. 4(b). The dynamic range of the images is [0, 255], and the images are in YCbCr color format. From the figures, we can see that about 60% of the pixels have zero J^{MD} values and more than 90% of the pixels have J^{MD} values less than 2. The example images also show that J^{MD} maps are very dark. These give very strong supports to our prior.



(a) Distribution of J^{MD} values. (b) Example images and their J^{MD} maps.

Fig. 4. Verification of the gray-color correspondence prior. J^{MD} measures the maximum color differences between the centered pixels of patches that share similar gray intensities in local regions. As shown, the J^{MD} values in (a), which are obtained using images in the ImageNet dataset [16], are very low and tend to be zero for most pixels. And the J^{MD} maps in (b), whose values are multiplied by 10 for better visualization, are very dark too. For more details, see Section 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In seldom cases, the prior is not correct because it is possible that two patches with different colors are in different lighting conditions, and the difference of the lighting conditions happens to compensate for the difference of the waves' energy of the different colors, leading to the patches having different colors but the same gray intensity. Our evaluation method in Section 5.1 considers this possibility. We will select out these cases to avoid failed colorization.

4. Colorization in the dual-lens system

Based on the gray-color correspondence prior, we propose an efficient colorization algorithm in this section. The challenge is that, due to different light efficiency between monochrome and color cameras, the same scene will have different intensities in the monochrome image and the gray channel of the color image.

Traditional reference-based colorization algorithms, such as [7,8], usually use hand-crafted features of the patch to search for the matching patches. We propose a ResNet [18] based convolutional neural network, called color similarity network (shown in Fig. 5), to extract the deep features of patches and use them for searching similar patches. For each patch in the input gray image, its search range in the reference image accords with the search range of the aforementioned gray-color correspondence prior, i.e. the patches with the same vertical positions and the position differences in the horizontal position is from 0 to $d-1$. It is possible that no patches in the reference image could provide correct colors due to occlusions, so we use the color propagation method [6] to colorize occluded pixels. For further accelerating the colorization, we propose the coarse-to-fine colorization network (shown in Fig. 6). We do the reference-based colorization at the low resolution to obtain the coarse colorization result. Then, the coarse-to-fine colorization network upsamples the coarse result with the original-resolution gray image as guidance.

4.1. Colorization

First, we propose the color similarity network to extract the features of patches for measuring the color similarity of different patches. As shown in Fig. 5, the input to the proposed network is a pair of image patches and the output is a measure of the color similarity between the centered pixels of the two patches. The architecture is a siamese network, i.e. two shared-weight sub-networks joined at the head [15]. The sub-networks are composed of a number of residue blocks [18] with Batch Normalization and ReLu following all layers but the last layer (in this paper, each sub-network has 8 residue blocks. Each residue block has 2 convolutional layers and the residue connection). Both sub-networks output a vector capturing the feature of the input patch. And the resulting two vectors are compared using the cosine similarity measure so as to get the final output of the network. The proposed network is similar with the stereo matching method [15]. The differences are (1) our network is based on ResNet [18] which has proven to be effective for related problems, e.g. super resolution [19], etc. (2) In the training, we label the similarity of each pair of patches by their color similarity while the method of [15] labels the similarity by the ground-truth disparity. As explained in

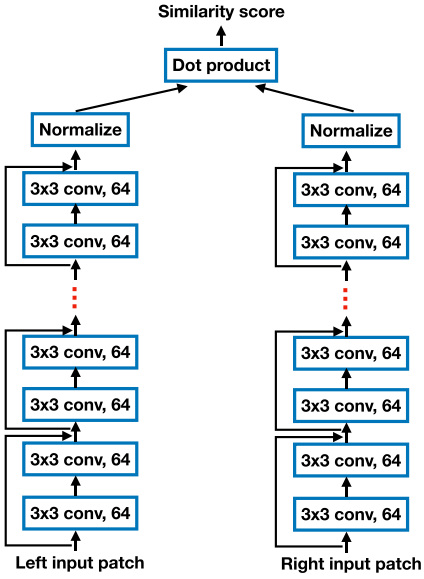


Fig. 5. The architecture of the color similarity network. It is a siamese network for estimating the color similarity score of two patches. The sub-networks are composed of 8 residue blocks [18] with Batch Normalization and ReLu following all layers but the last layer. Each residue block has 2 convolutional layers and the residue connection. The similarity score is obtained by extracting a vector from each of the two input patches and computing the cosine similarity between them. The cosine similarity computation is split in two steps: normalization and dot product for improving computation efficiency.

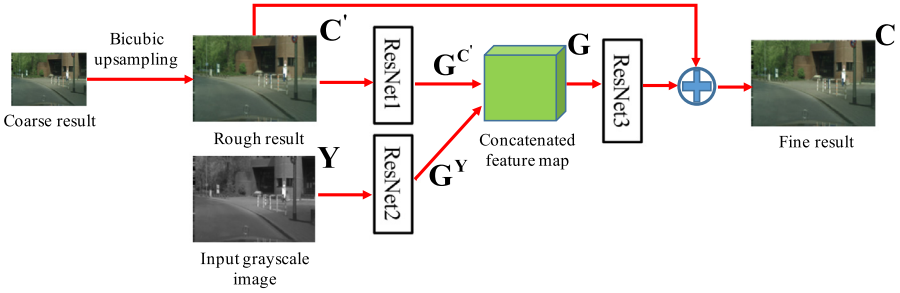


Fig. 6. The architecture of the proposed coarse-to-fine colorization network.

Section 1, color and disparity are different things, and the same pair of patches may have different labeling values in these two cases.

To build the training set, for each patch $\mathbf{x}$ in the input gray image, one positive patch $\mathbf{y_1}$ and one negative patch $\mathbf{y_2}$ in the reference image are used in training. The positive pair $(\mathbf{x}, \mathbf{y_1})$ is labeled as 1, while the negative pair $(\mathbf{x}, \mathbf{y_2})$ is labeled as 0. We judge the positive and negative patches by their color differences with $\mathbf{x}$. Specifically, the color differences between the centered pixels of $\mathbf{x}$ and $\mathbf{y_1}$ are less than m, while the color differences between the centered pixels of $\mathbf{x}$ and $\mathbf{y_2}$ are greater than m (m is set as 2 in this paper). For each patch $\mathbf{x}$, the search range of positive and negative patches is the patches in the reference image that have the same vertical position but the differences in the horizontal position is from 0 to $d-1$. Usually, there exist multiple positive and negative patches and $\mathbf{y_1}$ and $\mathbf{y_2}$ are randomly selected from the positive and negative patches, respectively.

The proposed network is efficient in computation. To search for the best-matching patches, we just need to run the sub-networks once on each image to extract the features of all pairs of patches and run the dot product of the feature vectors d times.

Using the proposed features, for each patch in the input gray image, we search for its best-matching patch in the search range of the reference image. But, occlusion regions usually exist between the image pair. And it is possible that no patches in the reference image could provide correct color, so the best-matching patches still have small similarity score. If the best-matching patch has high similarity score, we see them as located in high confidence regions. Otherwise, the patches belong to low confidence regions. For high confidence regions, we directly colorize the centered pixels of the patches using the color of the best-matching ones in the reference image. For low confidence regions, we adopt the method in [6] to propagate colors using the surrounding colorized regions. The optimization-based interpolation in [6] uses partial color information as seed colors to propagate colors to the complete image. In our work, the colors of the colorized pixels in high confidence regions are used as the seed colors. The reader is referred to [6] for further details. We set a constant threshold value $T = 0.75$ in this paper. If the similarity score of the best-matching patch is lower than T, we will see the patch with low confidence. Otherwise, the patch is with high confidence.

4.2. Coarse-to-fine colorization

For acceleration, we propose the coarse-to-fine colorization network to perform coarse-to-fine colorization. We share similar insights with [6] that neighboring pixels with similar gray intensities should have similar colors, and the input gray image $\mathbf{Y}$ could provide guidance of spatial color consistency. Our method is based on the deep joint filter [20]. Our difference from [20] is that (1) we use ResNet [18] instead of traditional 2-D convolution due to good performances of ResNet in related problems, and (2) we learn the residue between the ground truth color image and the rough colorization result, because learning the residue map has proven to be more effective in related works, e.g. single image super resolution [21].

Formally, we use the input gray image $\mathbf{Y}$ as guidance to correct the rough result $\mathbf{C}'$ by

$$\mathbf{C} = \mathbf{C}' + \Phi(\mathbf{C}', \mathbf{Y}), \tag{2}$$

where Φ denotes the operation of the coarse-to-fine colorization network.

As shown in Fig. 6, the rough colorization result $\mathbf{C}'$ and the input gray image $\mathbf{Y}$ are fed into two ResNets, named ResNet1 and ResNet2, to get their features $\mathbf{G}^{\mathbf{C}'}$ and $\mathbf{G}^{\mathbf{Y}}$, respectively. Then, $\mathbf{G}^{\mathbf{C}'}$ and $\mathbf{G}^{\mathbf{Y}}$ are concatenated to form the feature map $\mathbf{G}$, which is fed into another ResNet, named ResNet3, to get the residue color map $\Phi(\mathbf{C}', \mathbf{Y})$. By adding $\mathbf{C}'$ and the residue color map $\Phi(\mathbf{C}', \mathbf{Y})$, the final colorization result $\mathbf{C}$ is obtained. The coarse-to-fine colorization network can be seen as a high dimension joint filter.

ResNet1 has 16 convolution layers in total. The 16 layers are 8 repeated residue blocks and each residue block consists of 2 convolution layers with 3×3 kernel and a residue connection. BatchNorm layers and ReLu layers are added after each of the convolution layers except the last layer. The 16th layer is a convolution layer with 3×3 kernel and no BatchNorm layer or ReLu layer is added. The filter number the 16 layers of ResNet1 is a hyper-parameter which is set as 32 in this paper. ResNet2 has the same network structure as ResNet1. ResNet3 has similar structure with ResNet1. The difference between ResNet3 and ResNet1 is that in the last layer the filter number is 1 and no BatchNorm layer or ReLu layer is added. The parameters of ResNet1, ResNet2, and ResNet3 are trained separately.

As shown in Fig. 2, we firstly perform the colorization at the coarse level, i.e. the down-sampled resolution, and then up-sample the low resolution colorization result to the fine level, i.e. the original resolution, with the guidance of the original input gray image. In this paper, the downsampling and upsampling ratio is 5×5.

5. Colorization quality evaluation

We judge whether the input color image accords with the gray-color correspondence prior in Section 5.1. In addition, we propose the symmetry colorization based method to evaluate the colorization quality in Section 5.2.

5.1. Evaluation using the gray-color correspondence prior

Since our colorization method is based on the gray-color correspondence prior, the first step of evaluating the colorization quality is to judge whether the input color image accords with the gray-color correspondence prior. We compute the J^{MD} values of all pixels in the input color image according to Eq. (1), and then compute the maximum J^{MD} value of all pixels. When the maximum J^{MD}

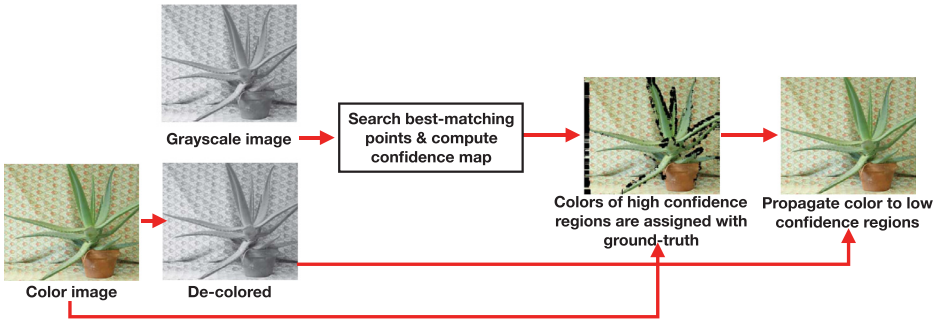


Fig. 7. Pipeline of the symmetry colorization method, which uses the gray image as reference to colorize the de-colored image of the color image. First, we search for best-matching patches and compute the confidence map. Second, for pixels in high confidence regions, we recover their colors using the ground-truth values from the color image. Third, for pixels in low confidence regions, their colors are propagated by surrounding colorized pixels. The first and third steps are the same as the colorization in Section 4.1. See more details in Section 5.2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

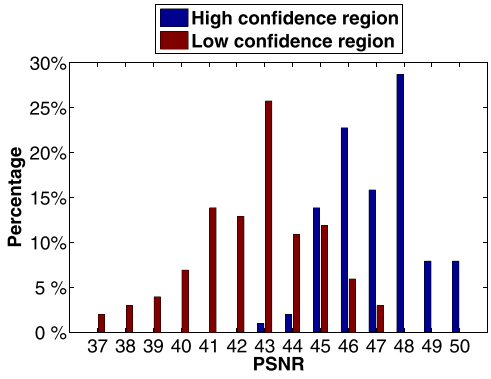


Fig. 8. Analysis of the qualities of colorization results in low and high confidence regions separately. The higher PSNR (Peak Signal to Noise Ratio) values indicate better colorization quality. As shown, the colorization qualities in high confidence regions are always high enough and can be seen as correct colorization results, while the colorization qualities in low confidence regions vary a lot and need to be evaluated. See more details in Section 5.2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

value is higher than a threshold, which is set as 10 in this paper, we see this image as an outlier to the gray-color correspondence prior. For outliers to the prior, the colorization results will not be reliable, so we directly output the input color image, without performing the colorization algorithms.

5.2. Symmetry colorization based evaluation

The challenge of evaluating the quality of the colorization result is the lack of the ground-truth color image. Our insight for solving this problem is the symmetry property of colorization, i.e. the colorization quality should be similar when colorizing the left image with the right one as reference or colorizing the right image with the left one as reference. Based on the symmetry property, we propose to do the colorization at the opposite direction, i.e. decoloring the input color image and colorizing the de-colored image using the input gray image as reference. Thus, the input color image itself can be used as the ground-truth color image to evaluate the colorization quality. This method is called the symmetry colorization method in this paper, and the pipeline is shown in Fig. 7.

The difficulty of the symmetry colorization method is that, since the reference image is gray, for high confidence regions according to the searching results, the corresponding pixels in the reference image do not have color information. To solve this problem, firstly, we analyzed the quality of the colorization results in low and high confidence regions separately in the Cityscapes dataset. As shown in Fig. 8, the PSNR (Peak Signal to Noise Ratio) values of the colorization results' high confidence regions are always higher than 43 dB, but the low confidence regions' values range from 37 dB to 47 dB. This statistic indicates that the colorization qualities in high confidence regions are always high enough while the colorization qualities in low confidence regions vary a lot. So, colorization results in high confidence regions can be seen as correct directly without evaluation, and we only need to evaluate the colorization quality in low confidence regions. To do so, we assume that if the reference image has color information, the colors will be the correct ones for the pixels in the high confidence regions. Thus, we assign the ground-truth colors to high confidence regions directly using the input color image. Then, we propagate colors to low confidence regions using the surrounding colorized pixels, which is the same as the propagation in Section 4.1. After getting the result of the symmetry colorization, we can evaluate the colorization quality using the ground-truth color image.

We test the correlation between the symmetry colorization based evaluation and the ground-truth colorization quality using the metric of PSNR. The linear correlation coefficient can achieve 0.93 on average over all the five datasets in our experiment. This high correlation indicates the accuracy of the proposed evaluation method.

Table 1
Two setups of the colorization benchmark. We simulate the monochrome-color dual-lens system by adding signal dependent Gaussian noise with a given standard deviation where κ represents the noise-free signal intensity [28].

Noise std.	Color camera	Monochrome camera
Setup1	$0.03\sqrt{\kappa}$	$0.01\sqrt{\kappa}$
Setup2	$0.07\sqrt{\kappa}$	$0.01\sqrt{\kappa}$

6. Experimental results

6.1. Datasets and experimental environment

We use five popular stereo datasets in our experiments, namely Cityscapes [23], KITTI [24], Middlebury 2006 [25], Middlebury 2014 [26], and Sintel [27]. These datasets contain pairs of color images captured by the dual-lens system with two color cameras. For realistic simulations, following [1], within each pair of images, we de-color one image and use the de-colored result as the input monochrome image, and the other color image is used as the input color image. In addition, we imitate the light-efficiency differences between color and monochrome cameras by adding different amount of noises to the monochrome input images and color input images. We configured two different setups for this experiment. The details are summarized in Table 1.

The proposed deep convolutional networks are implemented with Torch and the other processing modules are implemented

Table 2
Average PSNR values of different colorization methods in the five datasets under Setup1 in Table 1. The higher PSNR values indicate better colorization quality. 'MB' is short for Middlebury.

PSNR (dB)	Welsh et al. [7]	Ironi et al. [8]	Gupta et al. [9]	Jeon et al. [1]	Furusawa et al. [10]	Zhang et al. [3]	Iizuka et al. [4]	Ours
Cityscapes	37.89	38.45	38.09	39.33	34.74	29.38	31.30	**40.86**
KITTI	33.50	35.80	35.31	36.26	27.88	28.35	27.58	**36.63**
MB2006	31.28	32.98	32.04	36.80	30.86	29.12	29.19	**38.60**
MB2014	30.12	32.24	31.65	32.32	29.44	17.26	22.02	**37.14**
Sintel	34.94	36.06	35.45	36.12	32.13	29.34	33.97	**39.26**

Table 3
Average PSNR values under Setup2 in Table 1.

PSNR (dB)	Welsh et al. [7]	Ironi et al. [8]	Gupta et al. [9]	Jeon et al. [1]	Furusawa et al. [10]	Zhang et al. [3]	Iizuka et al. [4]	Ours
Cityscapes	35.06	35.68	34.53	35.38	32.91	29.57	31.39	**40.21**
KITTI	31.92	32.96	33.04	33.50	28.79	28.32	27.39	**36.48**
MB2006	29.60	25.42	31.72	31.75	29.52	28.41	28.42	**37.30**
MB2014	28.83	27.56	28.73	29.78	28.07	18.56	23.13	**36.16**
Sintel	32.70	32.52	33.31	33.98	32.01	29.44	34.02	**38.71**

Table 4
Average SSIM values of different colorization methods in the five datasets under Setup1 in Table 1. The higher SSIM values indicate better colorization quality.

SSIM	Welsh et al. [7]	Ironi et al. [8]	Gupta et al. [9]	Jeon et al. [1]	Furusawa et al. [10]	Zhang et al. [3]	Iizuka et al. [4]	Ours
Cityscapes	0.8971	0.8976	0.9488	0.9532	0.8417	0.4606	0.7572	**0.9772**
KITTI	0.7941	0.9294	0.9329	0.9688	0.7605	0.7517	0.8084	**0.9776**
MB2006	0.9063	0.9408	0.8969	0.9786	0.8603	0.7468	0.6772	**0.9876**
MB2014	0.8331	0.9008	0.8796	0.9578	0.8055	0.2897	0.4214	**0.9712**
Sintel	0.7959	0.9183	0.9333	0.9437	0.7945	0.6872	0.8526	**0.9709**

Table 5
Average SSIM values under Setup2 in Table 1.

SSIM	Welsh et al. [7]	Ironi et al. [8]	Gupta et al. [9]	Jeon et al. [1]	Furusawa et al. [10]	Zhang et al. [3]	Iizuka et al. [4]	Ours
Cityscapes	0.8492	0.7781	0.9061	0.9142	0.8252	0.4553	0.7517	**0.9645**
KITTI	0.7609	0.8608	0.8776	0.9180	0.7603	0.7511	0.7983	**0.9721**
MB2006	0.8765	0.7151	0.8938	0.9530	0.7827	0.7520	0.6882	**0.9817**
MB2014	0.7890	0.7675	0.7706	0.9225	0.7542	0.3030	0.4344	**0.9592**
Sintel	0.7580	0.8146	0.9059	0.9243	0.7284	0.6882	0.8526	**0.9661**

with Python. The experiments are performed on an Intel I7 2.6 GHz machine with 8GB memory and an NVIDIA Titan-X GPU.

When we train the color similarity network, the input to the network is a batch of 128 pairs of image patches. The loss function we use is the mean squared error between the prediction result and the ground-truth annotation. We minimize the loss using mini-batch gradient descent with the momentum term set to 0.9. We train for 20 epochs with the learning rate set to 0.001. We train the color similarity network on the datasets of Middlebury 2006 and 2014, which contains 38 million examples. When testing the performance on the datasets of Cityscapes, KITTI, and Sintel, we directly use the model trained on Middlebury for cross-validation.

When we train the coarse-to-fine colorization network, the network is optimized with RMSProp and a constant learning rate of 0.001. We train with a batch size of 1 using a 256×512 randomly located crop from the input images. We train the network on the dataset of Sintel, which contains 1064 training and 564 testing images. The random crop operation helps augment the training data to 10,640 images by cropping 10 images randomly from each original training image. The loss function we use is the mean squared error between the prediction results and the ground-truth color maps.

During the training, first, we train the color similarity network, and use the trained network to get coarse colorization results of all images in all datasets. Second, we train the coarse-to-fine colorization network and use it to get the fine colorization results.

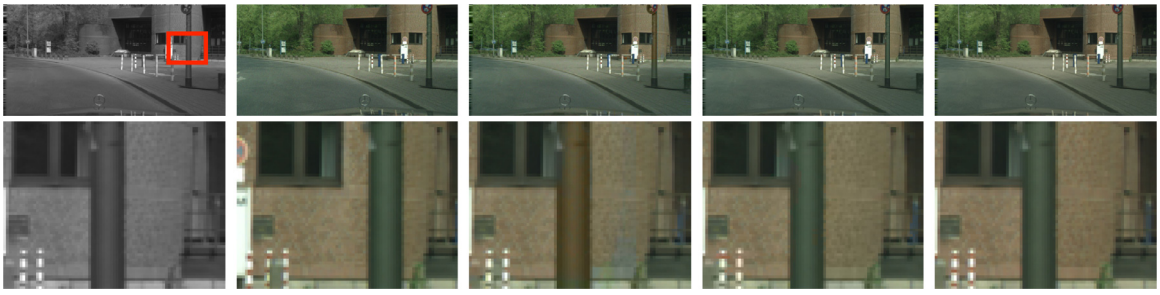
6.2. Experiment I: Colorization

Comparison algorithms: First, we compare with five state-of-the-art reference-based colorization algorithms, i.e. the methods of Welsh et al. [7], Ironi et al. [8], Gupta et al. [9], Jeon et al. [1] and Furusawa et al. [10]. In addition, we compare with two state-of-the-art deep learning based automatic colorization algorithms, i.e. the methods of Zhang et al. [3] and Iizuka et al. [4], which could automatically colorize monochrome images without any reference images. The methods of Welsh et al. [7], Ironi et al. [8], and Gupta et al. [9] do not assume short-baseline between the pair of images. So, for each pixel in the monochrome image, the search region is the whole reference image. For fair comparison, we re-implement the methods and make the search range the same as our method, which is defined in Section 3. The method of Furusawa et al. is designed for colorizing manga images while we aim at general images. When performing the method of Furusawa et al., the panel is set as the whole reference image.

Results: We show the quantitative results in Tables 2–5. As shown, our method largely outperforms the comparison methods. And some qualitative colorization results are shown in Figs. 9–11. As shown in Fig. 9, Welsh's et al. method does not have good performance, because their assumption of the correspondence from gray intensity to color value of all pixels is not true for many images. So, some regions are wrongly colorized. Ironi's et al. method has problems for edges and small objects because many unoccluded pixels are wrongly marked as occluded pixels, and thus the colorized pixels of unoccluded pixels are not enough for color

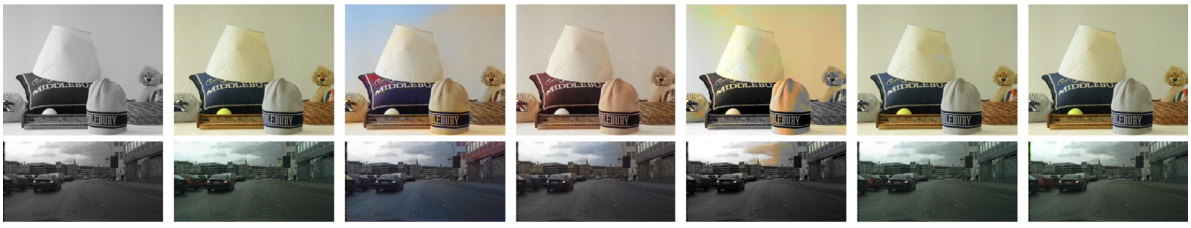
(a) Input gray and color images. (b) Welsh et al. (c) Ironi et al. (d) Gupta et al. (e) Our result. (f) Ground truth.

Fig. 9. An example to compare the colorization results of Welsh's et al. method [7], Ironi's et al. method [8], Gupta's et al. method [9], and our colorization method. The region marked with the red box is shown in the second row. As shown, the comparison methods fail to recover correct colors in the marked region. This example is under Setup1 in Table 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Input gray and color images. (b) Jeon et al. (c) Our result. (d) Ground truth.

Fig. 10. An example to compare the colorization results of Jeon et al.'s method [1] and our colorization method. The region marked with the red box is shown in the second row. As shown, Jeon's et al. method fails to recover correct colors in the marked region. This example is under Setup2 in Table 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)Input gray and color images. (b)Zhang et al. (c)Iizuka et al. (d)Furusawa et al. (e)Our result. (f)Ground truth.

Fig. 11. Examples to compare deep learning based automatic colorization algorithms, i.e. Zhang et al. [3] and Iizuka et al. [4], manga image colorization algorithm, i.e. Furusawa et al. [10], and our algorithm. As shown, due to not using the reference images as guidance, the recovered colors of Zhang et al. and Iizuka et al. are not correct in most regions. The method of Furusawa et al. fails in most regions too, because the assumptions of manga images are not true for general real-world images. The top and bottom examples are from Setup1 and Setup2 in Table 1, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

propagation. Gupta's et al. method does not perform well, especially for objects with complicated textures. It is because the features of each superpixel are obtained by averaging the feature values of all pixels in the superpixel, which will decrease the accuracy of correspondence searching for our problem. Jeon's et al. method has comparable results with ours as shown in the quantitative results. But they do not deal with the occlusion regions well. As shown in Fig. 10, there are occlusions in the red box region, and the results of their method are not correct. Furusawa's et al. result, as shown in Fig. 11, is not good enough because the method assumes that the images are manga images but in our problem the images are general images. The colorization qualities of the state-of-the-art CNN-based automatic colorization methods [3,4] are worse than most of the reference-based mathods and ours. As shown in Fig. 11, their results have wrong colors in most regions. It is because they are solving different problems. The input image in these methods is only single gray image. The reference color image, which could provide much useful color information during the colorization, is not utilized at all.

We also show the processing time of different methods in Table 6. From Table 6, we can find that ours is much faster than the other colorization methods, only costing 0.37 s per image on average. It is because most of our computation is performed on GPU, e.g. the ResNet feature extraction, the coarse-to-fine colorization, and only a small part of our method needs to be performed on CPU, i.e. searching for best-matching patches, and color propagation in low confidence regions. The GPU device is much faster than CPU, because the computation is paralleled, and thus our method is efficient in computation.

The performances of all the algorithms vary among different datasets. It is because the images in different datasets have different levels of occlusions. More occlusions usually lead to lower performance.

In the step of color propagation in low confidence regions, we also perform an ablation study to compare the method of Levin et al. [6] with Zhang et al. [5], a recently proposed CNN based edit propagation method. The PSNR values (dB) of the methods of Levin et al. and Zhang et al. over the five datasets are 38.09 and 36.94, respectively. The method of Levin et al. has better performance because in our case the colorized regions are very dense and only a small number of parts in the image need color propagation. Our results accord with the claims in [5] that the CNN based method

Table 6
Processing time of different colorization methods.

Time (s)	Welsh et al. [7]	Ironi et al. [8]	Gupta et al. [9]	Jeon et al. [1]	Furusawa et al. [10]	Zhang et al. [3]	Iizuka et al. [4]	Ours
2048×1024	1.6	11.7	114.5	227.1	21.3	5.1	3.8	0.37

Table 7
Average PSNR values of different upsampling methods in the five datasets.

PSNR (dB)	Bicubic	Dong et al. [19]	Huang et al. [11]	Ledig et al. [22]	Ours
Cityscapes	47.79	47.53	46.47	45.14	**48.61**
KITTI	46.76	46.51	45.32	45.93	**47.68**
MB2006	46.44	46.34	45.28	44.84	**47.01**
MB2014	47.44	47.94	47.57	46.02	**48.19**
Sintel	47.99	47.96	46.78	46.96	**49.81**

is competing when the scribble is sparse but their advantage is not obvious when the scribble is dense.

We also use the dual-lens system of Huawei P9, consisting of one monochrome camera and one color camera, to capture several pairs of images. The results are reported in Fig. 1, which have better quality in the monochrome channel and correct colors.

6.3. Experiment II: Coarse-to-fine colorization

(1) First, we compare our upsampling method with different upsampling methods in the literature. Our method uses the ground-truth low resolution color image as the input, and the input high resolution gray image as guidance.

Comparison algorithms: The state-of-the-art upsampling methods are compared, including Bicubic, Huang's et al. method [11], Dong's et al. method [19], and Ledig's et al. method [22].

Results: The quantitative and qualitative up-sampling results are shown in Table 7 and Fig. 12, respectively. As shown, our method gets the highest performance for all datasets. And the qualitative results in Fig. 12 show that we can do well in both textureless regions and texture regions. The methods of Bicubic, Dong et al., Huang et al., and Ledig et al. have limited performances, because no guidance of the original gray image is used in the upsampling. The methods of Dong et al., Huang et al. and Ledig et al. have even lower performances than Bicubic, because they are designed to super resolve in both monochrome and color channels and tend to add more texture and details into the result. In our case, only the colors need upsampling but their results are usually over-enhanced. The processing time of the comparison methods is shown in Table 8. As shown, our method is efficient but most of the comparison algorithms are costly in computation.

Table 8
Processing time of different upsampling methods for the coarse-to-fine operation, with upsampling ratio 5×5.

Time (s)	Bicubic	Dong et al. [19]	Huang et al. [11]	Ledig et al. [22]	Ours
5×5	0.05	23.4	145.1	62.7	0.09

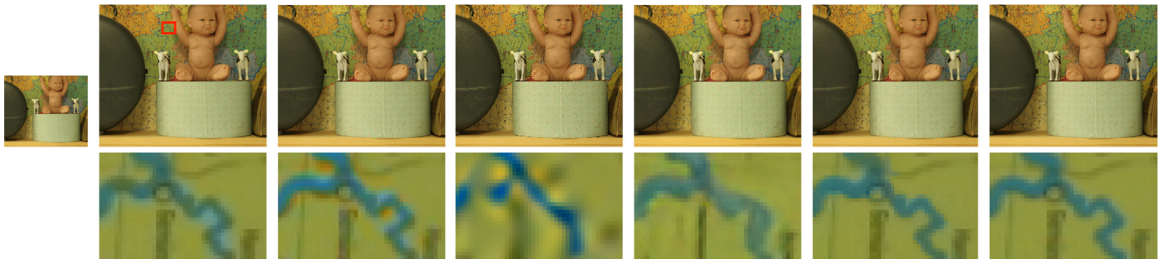
Table 9
Quantitative results of our method without the coarse-to-fine (CTF) operation.

Ours without CTF	PSNR in Setup1	PSNR in Setup2	SSIM in Setup1	SSIM in Setup2
Cityscapes	39.37	36.27	0.9669	0.9271
KITTI	35.97	35.65	0.9671	0.9376
MB2006	37.07	33.80	0.9757	0.9597
MB2014	35.18	32.22	0.9618	0.9223
Sintel	37.49	35.11	0.9624	0.9348

(2) Second, we also conduct an ablation study, where our method without the coarse-to-fine operation is compared.

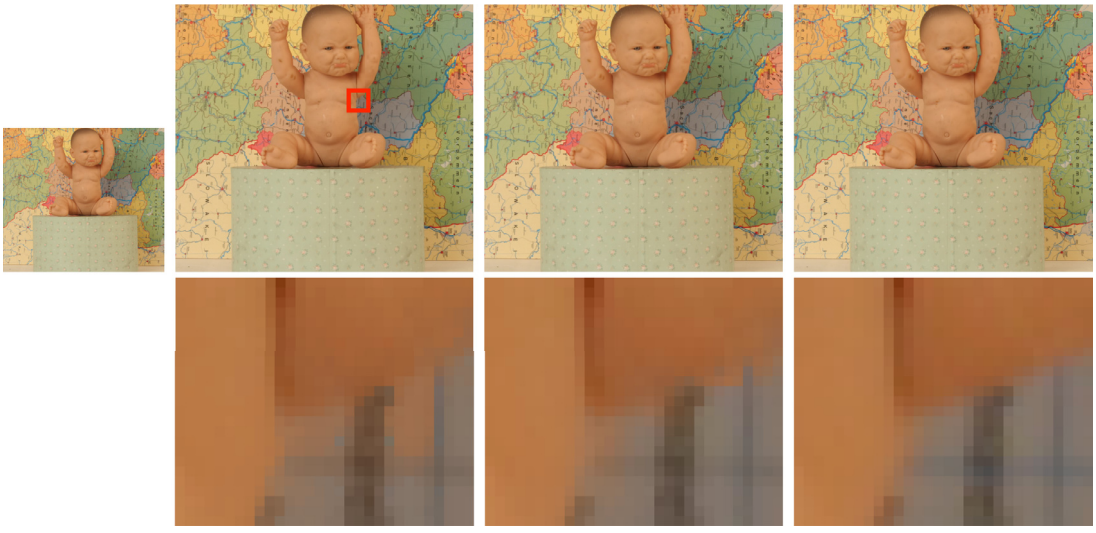
Results: The quantitative results of our method without the coarse-to-fine operation are shown in Table 9. As shown, our coarse-to-fine colorization method (whose results are presented in the 'Ours' column of Tables 2–5) could have higher accuracy than our method without the coarse-to-fine operation. It is because, with the guidance of the high resolution input gray image, some colorization errors at the coarse level could be corrected during the joint upsampling process. From Table 8, we can also find that the coarse-to-fine operation is efficient in computation, only costing 0.09 s per image on average. And performing the colorization with the coarse-to-fine operation costs 0.37 seconds in total, as shown in Table 6. In comparison, if we directly perform the colorization at the fine level, the computation time is 2.1 seconds per image on average. This shows the benefits of the coarse-to-fine operation in computation. The reason is that the colorization part (introduced in Section 4.1) is more computational costly than the joint upsampling part (introduced in Section 4.2). Colorization with the coarse-to-fine operation could reduce the computation in the colorization part, and thus reduce the computation cost in total. In short, the coarse-to-fine operation has benefits for both colorization quality and computation efficiency.

(3) Third, we conduct an ablation study where we try traditional convolutional layers instead of ResNet blocks in the color



(a)LR image. (b) Bicubic. (c) Dong et al. (d) Huang et al. (e) Ledig et al. (f) Our result. (g) Ground truth.

Fig. 12. An example to compare the upsampling results of Bicubic, Dong et al. [19], Huang et al. [11], Ledig et al. [22] and our upsampling method for the low resolution (LR) color image in (a). The region marked with the red box is shown in the second row. As shown, the comparison methods fail to upsample the color channels (Cb and CR) correctly in the marked region while our upsampling method has much better result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) LR. (b) Upsampled by Levin. (c) Our result. (d) Ground truth.

Fig. 13. An example to compare the upsampling results of Levin's et al. method [6] and our upsampling method for the low resolution (LR) color image in (a). The region marked with the red box is shown in the second row. As shown, Levin's et al. method has wrong result in the marked region while our upsampling method has much better result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(a) Input gray and color images. (b) Our result. (c) Ground truth.

Fig. 14. Examples of the outliers of our colorization results according to the symmetry colorization based evaluation method. The region marked with the red box is shown in the second row. As shown, our method fails to recover the color of the blue river in the marked region because it is completely occluded in the reference color image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 10
PSNR results (dB) of our method by varying the number of layers of ResNet blocks and using traditional convolutional layers (Conv) in the color similarity network.

	Number of layers	Cityscapes	KITTI	MB2006	MB2014	Sintel
ResNet	2*2	40.62	36.41	38.36	36.92	39.00
	2*4	40.81	36.59	38.56	37.11	39.22
	2*8	**40.86**	**36.63**	**38.60**	**37.14**	**39.26**
	2*16	40.86	36.62	38.59	37.13	39.26
Conv	4	38.81	34.56	36.38	35.16	37.15
	8	38.54	34.23	36.06	34.80	36.79
	16	38.42	34.16	35.91	34.67	36.65

Table 11
Results of colorization quality evaluation in all the five datasets. 'Outlier II' indicates the outlier images to the gray-color correspondence prior according to our evaluation method in Section 5.1. 'Inliers' and 'Outlier I' indicate the inlier and outlier colorization results respectively, according to the symmetry colorization based evaluation method in Section 5.2.

	Inlier	Outlier I	Outlier II
Percentage	87.27%	8.39%	4.34%
Average PSNR(dB)	44.26	36.07	33.15

similarity network, and we also conduct a series of experiments where we vary the number of convolutional layers. The results are shown in Table 10. From this table, we can find that using ResNet blocks could largely increase the quality than using traditional convolutional layers, because residue blocks are easier to train. And these experiments indicate that it is not 'the deeper the better' in this deep model for colorization. It may be caused by the difficulty of training when the network is deeper. And using 8 residue blocks is a proper choice for our colorization problem (Fig. 13).

6.4. Experiment III: Colorization quality evaluation

The quantitative results are shown in Table 11. As shown, the outliers to the gray-color correspondence prior only occupy 4.34%, which verifies that our prior is effective for most images. According to the symmetry colorization based evaluation method (the threshold we set is 42 dB), 87.27% of the colorization results are judged to be inliers while the rest 8.39% are outliers, caused mostly by complicated occlusions. The average PSNR of the inliers can achieve 44.26 dB, which is a very high PSNR value and can

indicate the high quality of the colorization. An example of the outlier results are shown in Fig. 14.

7. Conclusions

We introduce a deep learning based algorithm to shoot high-quality color images using the dual-lens system with monochrome and color cameras. Two problems, i.e. colorization in the dual-lens system, and the colorization quality evaluation, are solved in our paper. For colorization in the dual-lens system, we propose the gray-color correspondence prior. Based on the prior, a deep learning based and coarse-to-fine colorization method is proposed. For colorization quality evaluation, a symmetry colorization based evaluation method is proposed. Quantitative and qualitative experimental results verify our observations, and demonstrate the effectiveness of the proposed algorithm. The proposed algorithm is also efficient in computation.

Conflict of interest

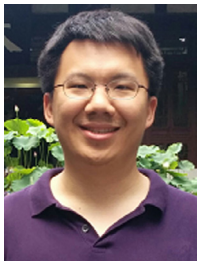
None.

Acknowledgments

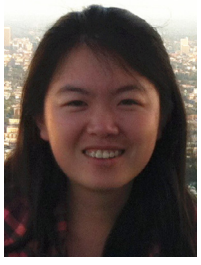
This work is funded by the National Natural Science Foundation of China (No. 61802026 and 61806016). We thank the anonymous reviewers for helping us to improve this paper.

References

[1] H.G. Jeon, J.Y. Lee, S. Im, H. Ha, I.S. Kweon, Stereo matching with color and monochrome cameras in low-light conditions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4086–4094.
[2] A. Chakrabarti, W. Freeman, T. Zickler, Rethinking color cameras, in: Proceedings of the IEEE International Conference on Computational Photography, 2014.
[3] R. Zhang, P. Isola, A. Efros, Colorful image colorization, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 649–666.
[4] 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 Trans. Graph. 35 (4) (2016) 110:1–110:11.
[5] R. Zhang, J. Zhu, P. Isola, X. Geng, A. Lin, T. Yu, A. Efros, Real-time user-guided image colorization with learned deep priors, ACM Trans. Graph. 36 (4) (2017) 119:1–119:11.
[6] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, ACM Trans. Graph. 23 (3) (2004) 689–694.
[7] T. Welsh, M. Ashikhmin, K. Mueller, Transferring color to greyscale images, ACM Trans. Graph. 21 (3) (2002) 277–280.
[8] R. Ironi, D. Cohen-Or, D. Lischinski, Colorization by example, in: Proceedings of the Rendering Techniques, 2005, pp. 201–210.
[9] R.K. Gupta, A.Y.S. Chia, D. Rajan, E.S. Ng, H. Zhiyong, Image colorization using similar images, in: Proceedings of the ACM international conference on Multimedia, 2012, pp. 369–378.
[10] C. Furusawa, K. Hiroshiba, K. Ogaki, Y. Odagiri, Comicolorization: semi-automatic manga colorization, in: Proceedings of the SIGGRAPH Asia, 2017.
[11] J. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exempla, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.
[12] A. Buades, B. Coll, J. Morel, A non-local algorithm for image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 60–65.
[13] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[14] M. Barnsley, Fractals Everywhere, Academic Press Professional, Inc., 1988.
[15] J. Zbontar, Y. LeCun, Stereo matching by training a convolutional neural network to compare image patches, J. Mach. Learn. Res. 17 (1) (2016) 2287–2318.
[16] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[17] G. Wyszecki, W. Stiles, Colour Science: Concepts and Methods, Quantitative Data and Formulae, in: Wiley Series in Pure and Applied Optics, New York, 1982.
[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[19] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 184–199.
[20] Y. Li, J. Huang, N. Ahuja, M. Yang, Deep joint image filtering, in: Proceedings of the European Conference on Computer Vision, 2016.
[21] J. Kim, J. Lee, K. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[22] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
[24] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The Kitti vision benchmark suite, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.
[25] D. Scharstein, C. Pal, Learning conditional random fields for stereo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[26] D. Scharstein, H. Hirschmuller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, P. Westling, High-resolution stereo datasets with subpixel-accurate ground truth, in: Proceedings of the German Conference on Pattern Recognition, 2014, pp. 31–42.
[27] D.J. Butler, J. Wulff, G.B. Stanley, M.J. Black, A naturalistic open source movie for optical flow evaluation, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 611–625.
[28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, Multiplexing for optimal lighting, IEEE Trans. Pattern Anal. Mach. Intel. 29 (8) (2007) 1339–1354.



Xuan Dong received the Ph.D. degree in Computer Science from Tsinghua University in 2015, and the B.E. degree in Computer Science from Beihang University in 2010. He is currently an assistant professor in the School of Computer Science, Beijing University of Posts and Telecommunications, China. His research interests include computer vision and computational photography.



Weixin Li received the Ph.D. degree in computer science from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 2017. She is currently an Associate Researcher at the School of Computer Science and Engineering (SCSE) and Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), Beihang University, Beijing, China. Her research interests include computer vision, image processing, and big data analytics.