

Shadow Remover: Image Shadow Removal Based on Illumination Recovering Optimization

Ling Zhang, Qing Zhang, and Chunxia Xiao, *Member, IEEE*,

Abstract—In this paper, we present a novel shadow removal system for single natural images as well as color aerial images using an illumination recovering optimization method. We first adaptively decompose the input image into overlapped patches according to the shadow distribution. Then, by building the correspondence between the shadow patch and the lit patch based on texture similarity, we construct an optimized illumination recovering operator, which effectively removes the shadows and recovers the texture detail under the shadow patches. Based on coherent optimization processing among the neighboring patches, we finally produce high-quality shadow-free results with consistent illumination. Our shadow removal system is simple and effective, and can process shadow images with rich texture types and nonuniform shadows. The illumination of shadow-free results is consistent with that of surrounding environment. We further present several shadow editing applications to illustrate the versatility of the proposed method.

Index Terms—Shadow detection, shadow removal, shadow matting, patch matching, aerial images.

I. INTRODUCTION

SHADOWS are natural phenomena, which occur when the light is blocked. Although shadows provide important visual cues for object shape perception, illumination position, objects occlusion, etc., shadow-free images can help to improve the performance of the tasks such as object recognition, object tracking and information enhancement [1], [2]. For example, for high spatial resolution remote sense image, shadow removal is very critical for target identification and information recovering. Shadow removal and editing can also improve the visual realism and physical realism in image processing [3], [4]. Shadow removal is now an popular research direction in computer vision and image processing communities. Many shadow removal approaches have been proposed in the last decade (see [2], [5], and [6] for a survey).

As image shadow is formulated in different lighting conditions, surface materials and scene shape, producing

Manuscript received November 3, 2014; revised May 3, 2015 and July 5, 2015; accepted July 28, 2015. Date of publication August 5, 2015; date of current version August 31, 2015. This work was supported in part by the National Basic Research Program of China under Grant 2012CB725303, in part by the National Natural Science Foundation of China under Grant 61472288, New Century Excellent Talents under Grant NCET-13-0441, and in part by the Key Grant Project of Hubei Province under Grant 2013AAA02. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Javier Mateos. (*Corresponding author: Chunxia Xiao.*)

The authors are with the School of Computer, Wuhan University, Wuhan 430072, China (e-mail: lingzhang@whu.edu.cn; zhangqing.whu.cs@gmail.com; cxxiao@whu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2015.2465159

high-quality shadow-free results is a challenging work for both natural images and aerial remote sensing images. The first problem to be addressed is shadow detection. For image with complex shadows, accurately shadow detecting is a difficult problem. For example, sometimes it is even difficult for human to distinguish little dark objects from the scattered shadow points. The second problem is, when the illumination conditions, object materials, and scene shapes are complex, the shadows in the image are usually nonuniform, which makes it difficult to obtain consistent shadow removal results. Finally, as the illumination usually changes dramatically in the boundary regions, effectively recovering the illumination on the shadow boundaries is also a challenging task.

In this paper, we present a novel shadow removal approach using an illumination recovering optimization method. We first detect the shadows in the input image, and compute the shadow alpha for the shadows. Then we adaptively decompose the input image into overlapped patches according to the shadow distribution. Denser patches are put on the shadow boundaries and the regions with dramatically changed illumination. Finally, by building the correspondence between the shadow patch and the lit patch based on illumination independent texture similarity, we develop an optimized illumination recovering operator which can effectively remove the shadows and recover the texture detail under the shadow patches. By using coherent illumination optimization processing among the neighboring patches, we produce high-quality shadow-free results. The texture details under the shadow regions are effectively recovered, and the recovered illumination is consistent with that of surrounding environment. Fig. 1 shows the overview of the proposed shadow removal system.

The proposed method is simple and effective. With the adaptively image patch decomposition and illumination independent patch matching, our method can process nonuniform shadows and shadow regions with rich texture types. While most existing shadow removal methods do not work well in processing images with complex shadows and various texture materials. Furthermore, with the adaptively image decomposition, our shadow removal system also works well on the shadow boundaries. In addition, we present several applications for the proposed method, such as illumination and color transfer, shadow edge softening. In our applications, shadow sharpness, position, and intensity can be freely adjusted, which illustrate the versatility of the proposed method.

The main contributions and advantages of our work are as follows:

- Introduce a novel local illumination recovering optimization, which produces illumination consistent

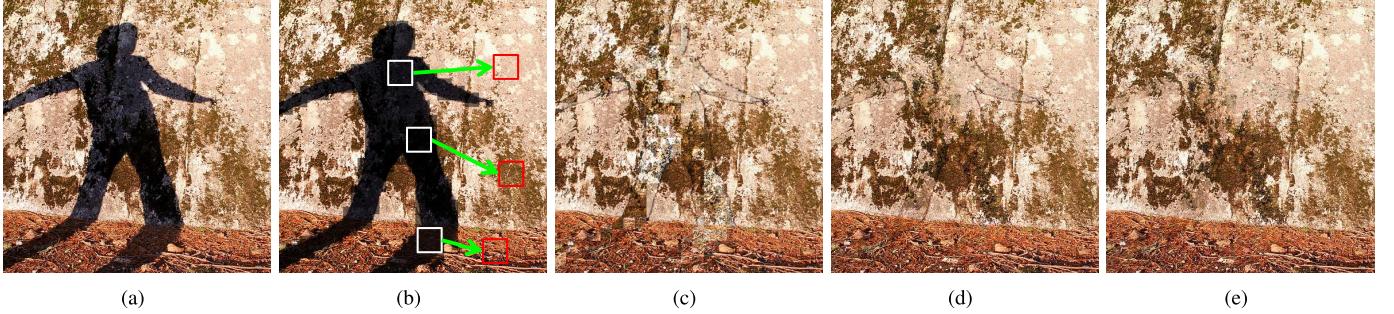


Fig. 1. The overview of our shadow removal system. (a) Input image, (b) nearest patch search for the shadow patches, (c) shadow removal result based on the patch pairs, (d) shadow-free result with coherence optimization, (e) final shadow-free result with boundary processing.

results on images with nonuniform shadows and multi-texture types.

- Develop an adaptive image decomposition strategy for patch matching used in shadow removal, which generates compelling results on both shadow interior and shadow boundaries.
- Present several image editing applications using the proposed shadow removal operator, which illustrates the versatility of the proposed approach.

The remainder of this paper is organized as follows. In Section II, we introduce the related work. In section III, we perform the shadow detection and shadow alpha matting, which is a preprocessing step. Section IV introduces the local illumination recovering operator. Section V presents the shadow removal approach. In Section VI, we present several applications of our method. Section VII shows the experimental results, and Section VIII concludes the paper.

II. RELATED WORK

Shadow removal involves two basic stages: shadow detection and shadow removal. So far, many shadow detection methods have been proposed, including automatic shadow detection methods [1], [5], [7], [8] and user-assisted shadow detection methods [9]–[11]. Many video shadow detection methods also have been proposed [12], [13]. A variety of shadow removal methods have been developed in computer vision, including the shadow removal from single images [1], [2], [5], [6], [14], [15], and shadow removal from multiple images and video sequences [16], [17]. A comprehensive survey is beyond the scope of this paper, we only focus on shadow removal methods from a single image and review the most related work to our paper.

Methods Based on Gradient Domain: Finlayson et al. proposed a series of shadow removal methods [1], [18], [19] based on gradient domain manipulation. The basic idea in these methods is to reconstruct the shadow-free image based on the gradient information in shadow regions by nullifying the gradients on shadow boundaries. Mohan et al. [3] proposed a shadow removal method by fitting a gradient domain shadow edge model. This method also simulates a variety of lighting conditions, such as ambient lighting, for shadow editing. Liu and Gleicher [11] removed shadow by assigning gradients for the shadow and lit area according to illumination

and gradient change in penumbra area. Although impressive results have been presented, the gradient domain based shadow removal methods may fail to produce high-quality shadow-free results for the nonuniform shadows since they only modify the gradients in shadow edges or penumbra regions. To receive satisfied results, the gradient domain based method needs to detect accurate shadow edges [1], or find best function fit for the shadow edge intensity [3], [11]. Since shadow edge detection is also a challenging work, this limits the practicality of these methods.

Methods Based on Illumination Transfer: Inspired by color transfer technique [20], several shadow removal methods have been proposed. The basic idea of color transfer is applying statistical analysis to transfer one image's color characteristics to another. Shor and Lischinski [10] built linear mapping models from the shadow and nonshadow areas based on illumination transfer. However, the shadow and nonshadow samples used for estimating the illuminated recovering parameters should have similar texture materials, which means this method can only handle shadow regions with uniform texture. Xiao et al. [21] proposed an adaptive multi-scale illumination transfer technique considering the material reflectance variation, and improved to recover the texture details of the shadow regions. The global color transfer methods [10], [21] applied a global transformation on the shadow regions to match color statistics of the nonshadow regions. As they do not take texture variety into account, they work well when the shadow and nonshadow sample regions exhibit similar texture.

More recently, Xiao et al. [5] further improved [21] by performing illumination transfer on the matched subregion pairs between the shadow regions and the nonshadow regions. This method can process complex images with different kinds of shadow texture regions and illumination conditions. But it fails to ensure smooth transition between subregions, and may produce illumination inconsistent results. Li et al. [6] applied color transfer results as the predicted shadow-free image used in the fidelity item. They also introduced an adaptive NL (nonlocal) regularized shadow removal method for aerial images by regularizing the shadow scale and the updated shadow-free image. The adaptive behavior regionally smooths the shadow scale, which can preserve the edges and the textures in the shadow regions. The NL Laplacian prior

can reduce noises in the shadow regions, but introduces texture detail blurring artifacts.

Methods Based on Shadow Matting: Chuang *et al.* [17] proposed a shadow matting method which considered the input image as a linear combination of a shadow-free image and a shadow matte image. Although this method can be used to remove shadow, it focuses on shadow extracting and compositing, and does not pay much effort for shadow removal. Instead of considering shadow extraction as the conventional matting equation, Wu *et al.* [9] and Wu and Tang [22] supposed shadow effect as a light attenuation problem. Wu and Tang [22] proposed a Bayesian framework for shadow extraction and produced shadowless image. Later, Wu *et al.* [9] first estimated an approximate shadowless image using color transfer techniques [20], and then defined an optimization function incorporating the approximate shadowless image for shadow extracting. Both [9] and [22] applied user-supplied hints to identify shadow and nonshadow regions. Although [9], [22] tried to preserve the texture appearance under the extracted shadow, to process complex nonuniform shadows, both methods still may not work well in recovering the image detail in the shadow areas.

Other Shadow Removal Methods: Guo *et al.* [8] presented a regions-based approach for shadow detection and removal. By segmenting the input shadow image using mean shift algorithm, Guo *et al.* explored the pairwise relationship between regions, and provided information about illumination condition of regions for shadow detection. They estimated the ratio between direct light and environment light, and recovered the illumination of shadow regions by relighting each pixel. This approach can produce high-quality shadow-free results for simple shadow images. However, for image with complex shadows and nonuniform shadows, this method cannot recover satisfactory texture detail since it does not take the reflectance variation into account. More recently, Xiao *et al.* [23] transformed the shadow removal problem into nonlocal feature matching between unshadowed samples and shadow pixels, and recovered the illumination from a single RGB-D image by applying an energy minimization method. They implemented the feature matching by measuring similarities between nonlocal pixels using normals, chromaticity and spatial locations. However, the performance of this algorithm relies on accurate depth map provided by the depth sensor such as MS Kinect.

Arbel and Hel-Or [2] considered each image channel as an intensity surface, and approximated the surface shape of shadow regions using a smooth thin-plate with the constraint on the shadow edges and the texture anchor points. This method can remove some nonuniform shadows. However, smooth thin-plate approximation is difficult to accurately estimate the shadow scale factors on the textured and highly structured images. Although texture anchor points have been used to alleviate this problem, there is still much room for improvement. Assuming a single flat texture shadow surface, Baba *et al.* [24] presented a shadow removal method based on color and variance adjustment of shadow pixels in RGB space. The method in [25] is based on the estimation of shadow scale factors assuming uniform shadow intensities

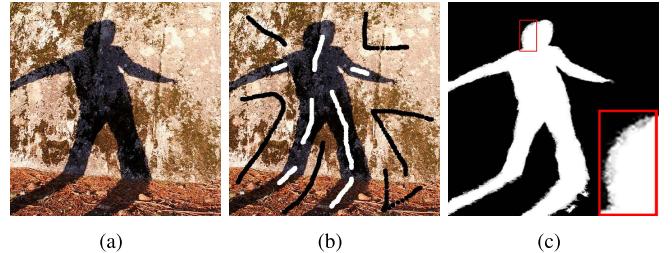


Fig. 2. Shadow matte result. (a) Input image, (b) brushes for sample regions, (c) shadow matte map, where the white regions represent shadow regions, the black regions are lit regions and the rest is the shadow boundary, and the bottom left is close-up for the red box.

and hard shadows. Both methods [24], [25] used inpainting techniques for completion of missing information in shadow boundary regions.

III. SHADOW DETECTION AND ALPHA MATTING

To perform shadow removal, we first have to detect the shadows. The current shadow detection methods can be divided into two categories: automatic shadow detection [26], [27] and interactive shadow detection [6], [10], [11]. As automatic shadow detection is an extremely difficult task, we incorporate user interaction and shadow alpha matting for shadow detection. Similar to interactive image matting, we first specify some shadow samples and lit samples, and construct a trimap for the input image. With the trimap, we can extract the shadow alpha matte using the specified samples as the constraints. More specially, we set $\alpha = 0$ for the specified shadow samples, and $\alpha = 1$ for the specified lit samples. Then we apply the closed-form matting method [28] and minimize the following energy equation for computing the shadow alpha:

$$\alpha = \arg \min \alpha^T L \alpha + \lambda (\alpha^T - b_S^T) D_S (\alpha - b_S) \quad (1)$$

where L is the Laplacian matrix, D_S is a diagonal matrix whose diagonal elements are one for constrained pixels and zero for all other pixels, and the vector b_S contains specified alpha values for the constrained nonshadow pixels and zero for all other pixels. Minimizing above energy equation, we can get the shadow alpha α .

With the computed shadow alpha matte α , we can identify the shadow regions of the input image. Specially, given threshold δ_1 and δ_2 , the pixels with $\alpha < \delta_1$ can be considered as the pixels in the shadow regions (umbra regions). The pixels with $\alpha > \delta_2$ can be considered as pixels in the nonshadow regions. The pixels with $\delta_1 \leq \alpha \leq \delta_2$ can be considered as the pixels in the penumbra regions, for hard shadow, these pixels can be considered as the shadow boundaries. In the shadow boundaries, the α value usually changes dramatically. In our experiments, we set $\delta_1 = 0.2$ and $\delta_2 = 0.9$. Fig. 2 shows a shadow matte result. Note that several methods [6], [10] also used closed-form matting method [28], [29] for shadow detection.

IV. LOCAL ILLUMINATION RECOVERING OPERATOR

In image formation equation [30], an image is the pixel-wise product of illumination and reflectance: $I_x = R_x L_x$, where

I_x is the observed RGB color at pixel x . L_x and R_x are the illumination and the reflectance (albedo) at pixel x , respectively. We assume shadows in the scene are cast due to the single primary source of illumination is blocked. If the pixel x is in the lit regions, the illumination can be described as a sum of the direct illumination L^d and the indirect (ambient) illumination L^a : $L_x = L^d + L^a$. In the umbra regions, since the object occludes the primary light source, it will cast a shadow at pixel x , and the cast illumination is only the ambient illumination: $L_x = L^a$. In the penumbra regions or shadow boundaries, as the occluder blocks some of the direct illumination, the practical illumination including the ambient illumination and some of the direct illumination, which can be described as: $L_x = \alpha_x L^d + L^a$, where α_x is the shadow matting alpha value of pixel x (introduced in Section III), which can be regarded as the attenuation factor of the direction illumination. Thus, the shadow color at pixel x and its shadow-free value I_x^{free} can be expressed as:

$$\begin{cases} I_x = (\alpha_x L^d + L^a)R_x \\ I_x^{free} = (L^d + L^a)R_x \end{cases} \quad (2)$$

Shor and Lischinski [10] claimed that there is a linear relationship between the observed value of pixel in a shadow region and its shadow-free value. The shadow-free value at pixel x can be represented as $I'_x = kI_x + b$, where $k = \frac{\sigma(L)}{\sigma(S)}$ and $b = \mu(L) - k\mu(S)$. $\mu(S)$ is the average value of the shadow sample regions, and $\mu(L)$ is the average value of the lit sample regions which have similar textures to the shadow sample regions. $\sigma(L)$ and $\sigma(S)$ are the standard deviation corresponding to the regions. Note that, this shadow removal model is only used in images with single texture, while can not work well for images with complex textures. The use of local patch in our method is just to meet this demand. We employ the results of Shor et al. to estimate the initial value in our shadow removal system. Specifically, we replace the shadow sample regions and the lit sample regions with the shadow patch and its matched lit patch, and the shadow-free value at pixel x in a shadow patch can be estimated as:

$$I'_x = \frac{\sigma(L)}{\sigma(S)}(I_x - \mu(S)) + \mu(L) \quad (3)$$

Then Equation (2) can be reformulated as:

$$\frac{L^d}{L^a} = \frac{I_x - I'_x}{\alpha_x I'_x - I_x} \quad (4)$$

Let $t = \frac{I_x - I'_x}{\alpha_x I'_x - I_x}$, we have

$$L^d = tL^a \quad (5)$$

Based on the above analysis, the shadow-free result of pixel x in shadow regions (umbra regions or penumbra regions) can be estimated as:

$$I_x^{free} = (t + 1)L^a R_x = \frac{t + 1}{\alpha_x t + 1} I_x \quad (6)$$

Our illumination recovering operator (Equation (6)) is based on the assumption that, in a local patch, the illumination and reflectance variation are small. Using this operator,

we can efficiently remove the shadows in the shadow patch using information of the matched patch with similar texture. To obtain a smooth shadow alpha α on the shadow boundaries, we estimate the alpha α for pixel on shadow boundaries using the average α value of the neighboring pixels.

V. SHADOW REMOVAL

From Equation (6) we know that, with the computed shadow matting alpha and the local reflectance constant assumption, for one patch in the shadow regions, if we find a lit patch in the nearby lit regions with similar material or texture, the shadow of this patch can be removed using our illumination recovering operator. To remove the whole shadows of the input image, we can transform this problem into the following solver: for each patch in the shadow regions, we find a corresponding patch with similar texture in the lit regions, and then by using the illumination recovering method on each corresponding patch pair, the shadows in the image can be removed.

However, to receive satisfactory results, the following several issues need to be addressed. The first issue is that if we remove the shadow of each patch separately, the results may be inconsistent and visually unnatural, thus, we should ensure that the shadow-free results have natural transitions between neighboring patches. The second important issue is that as the dramatically change of illumination may lead to texture detail loss on shadow boundaries, these boundaries should be elaborately processed to recover the texture information. Finally, for each patch in the shadow regions, to make the shadow removal processing efficient, the corresponding matched patch should be found efficiently in the lit regions.

To address above issues, we incorporate the illumination recovering operator into coherent optimization process. We first decompose the input image into adaptive overlapped patches. For each patches in the shadow regions, we find a corresponding patch in the lit regions for shadow removal. The shadow-free value for the pixel in the overlapped region is computed as the weighted average of all the shadow-removed values at its position from patches containing this pixel. By using this technique, we can receive shadow removal result with satisfied transition between patches. Finally, we exploit the image synthesis technology to recover the illumination and texture information on the shadow boundaries.

A. Image Decomposition

We decompose the input image into adaptive patches according to the illumination distribution. Formally, let I denote the input image, S and L are the shadow regions and lit regions of I , respectively. We first decompose the input image I into uniformly overlapped patches with patch size of $w \times w$ ($w > 10$). Instead of moving patch one pixel at a time, we shift patch five pixels at a time. Thus, we set patches that sufficiently overlap with each other which ensures that a given pixels can be affected by multiple patches, as shown in Fig. 3. This makes the shadow removal result exhibit natural and smooth transition between adjacent patches, and also alleviates the artifacts when a certain patch is matched inaccurately.

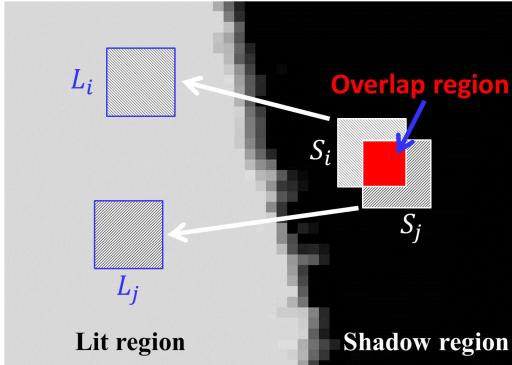


Fig. 3. Overlap region between patches. Both the shadow region and lit regions are decomposed using overlapped patches. For each patch in the shadow regions, we can find its nearest patch in the lit regions for shadow removing.

Since the illumination in the shadow boundaries usually changes dramatically, to receive satisfied shadow removal results, we further subdivide patches on the shadow boundaries with smaller patch size. In our experiments, the pixels with $\delta_1 \leq \alpha \leq \delta_2$ are considered as shadow boundary pixels, where $\delta_1 = 0.2$ and $\delta_2 = 0.9$. We subdivide the patch containing boundary pixels into four smaller patches. Using this method, we can receive adaptive patch decomposition for the input image. Note that we consider the patch containing pixels with $\alpha \leq \delta_2$ as the shadow patch. The remaining patches are lit patches.

Using adaptive decomposition, the content of the input image can be expressed using the shadow patches $\{S_k\}_{k=1,\dots,N_s}$ and the lit patches $\{L_k\}_{k=1,\dots,N_l}$, where N_s is the number of shadow patches and N_l is the number of lit patches. The lit patches present guided samples for shadow patches used in shadow removal. For each patch S_i in $\{S_k\}$, we find a matching patch L_j in $\{L_k\}$ which has similar texture with S_i . For the patch pair (S_i, L_j) , we remove the shadow on patch S_i by applying our local illumination recovering operator.

B. Fast Nearest Patch Matching

For each shadow patch S_i , we develop an effective texture matching metric to find a nearest patch in the lit regions. Our patch matching method is based on the following observation: the texture and color of two patches with the same reflectance are similar only when the illumination is the same; the color and intensity of these two patches may differ when the illumination is different. Thus, the patch matching metric should be illumination independently, that is, it should be robust in the presence of illumination changes. As the illumination difference between shadow regions and lit regions is usually large, we perform the effective patch search from two aspects. For finding the matched patch pair (S_i, L_j) , each aspect acts as a constraint condition in the searching processes.

Covariance Matrices: Region covariance descriptor [31] is an efficient feature description for a region in an image, which provides strong discriminative power in distinguishing local texture and image structures [32]. It represents

a region R with a covariance matrix of the feature points:

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{k=1}^n (\mathbf{z}_k - \mu)(\mathbf{z}_k - \mu)^T \quad (7)$$

where n is the pixel number of region R , \mathbf{z}_k is the d -dimensional feature vector of the k -th pixel, and μ is the mean feature vector of the pixels in region R . The d -dimensional feature vector can be chosen freely depending on different applications. In our method, we choose a 6D vector (intensity, chromaticity, first derivatives of the intensity in x and y directions, second derivatives of the intensity in x and y directions) as the feature vector of each pixel, and \mathbf{C}_R is a 6×6 covariance matrix. As subtracting the mean feature vectors, covariance matrices are less sensitive to illumination, which is preferred for patch matching between shadow regions and lit regions.

Since covariance matrices do not live on Euclidean space, we use the Cholesky decomposition [33] to transform covariance matrices into Euclidean space. We use the following vector [33] to represent the covariance matrix:

$$f(\mathbf{C}_R) = \{\mu, \sqrt{6}\mathbf{L}_1, \dots, \sqrt{6}\mathbf{L}_6, -\sqrt{6}\mathbf{L}_1, \dots, -\sqrt{6}\mathbf{L}_6\}^T \quad (8)$$

where \mathbf{L}_i is the i th column of the lower triangular matrix \mathbf{L} computed from the Cholesky decomposition of \mathbf{C}_R : $\mathbf{C}_R = \mathbf{L}\mathbf{L}^T$. To accelerate this search, we construct a KD-tree for the vector $f(\mathbf{C}_R)$ of all lit patches. We query the KD-tree with the vector $f(\mathbf{C}_R)$ of the shadow patches, and extract some nearest patches as the candidate lit patches for each patch S_i . In fact, a larger number of candidates contribute to more accurate patch matching while it requires higher computational cost. To achieve a good tradeoff between efficiency and accuracy during the patch matching, we choose five candidate lit patches in our experiments.

Spatial Distance: Spatial distance between two patches is also an important cue to find the corresponding patch. As spatially neighboring patches usually have similar illumination information, thus, for shadow removal, finding a lit patch close to shadow patch with similar texture is more appropriate to produce illumination coherent results. In our method, spatial distance metric is used to select the final matched patch from the five candidate lit patches. The spatial distance is computed as the square sum of the difference between the lit patch center and the shadow patch center. For the five candidate lit patches, we choose the patch which has the smallest spatial distance to S_i .

Xiao et al. [5] used Gabor wavelet descriptor to extract the texture feature, however this method is influenced by the illumination. That is, this descriptor is illumination dependently. In contrast, the texture descriptor we use to extract the texture information is illumination independently. In Fig. 4, we compare patch matching results based on our patch matching strategy by using region covariance descriptor and the Gabor wavelet descriptor [5], respectively. The results show that our texture descriptor outperforms Gabor wavelet descriptor using for texture matching.

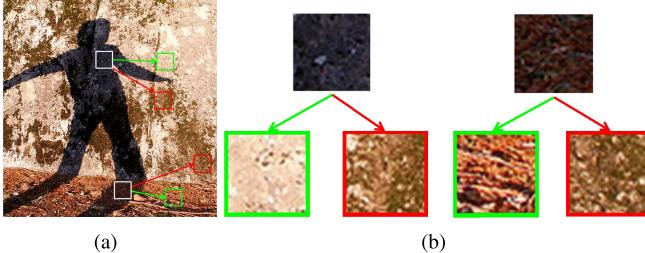


Fig. 4. Evaluation of the proposed texture descriptor. (a) The two source shadow patches are shown in white box, the target patches found by Gabor filter texture descriptor [5] are shown in red box, and our results are shown in green box, (b) close-ups for sample patches in (a).

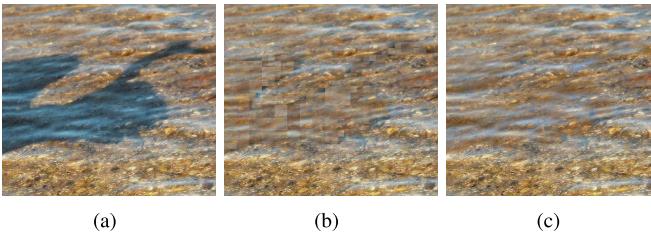


Fig. 5. Local illumination recovering optimization. (a) Input image, (b) shadow removal result without using overlapped patches, (c) shadow removal result using overlapped patches and illumination consistency optimization.

C. Coherence Recovering Optimization

Once we have found the nearest patch in the lit regions for each shadow patch, we can remove the shadow using Equation (6). Because of independent processing for each shadow patch, illumination between patches may be inconsistent, as illustrated in Fig. 5(b). To get coherent results and eliminate the potential blurring artifacts, we introduce illumination consistency optimization and texture detail enhancement in our method. Texture detail enhancement is an optional step which is required when texture blurring exists in shadow regions after the illumination consistency optimization.

1) Illumination Consistency Optimization: Since the sufficient overlap between patches, a pixel may be contained by multiple shadow patches. Thus the overlapped shadow pixel has different shadow-free value for different patch. Intuitively, to receive consistent result between adjacent patches, each pixel should take all possible shadow-free values into consideration.

Let $S(x)$ be a patch set containing pixel x . The weight $w_i = 1 - \frac{dis(x, center_S_i)}{\max_{S_k \in S(x)}[dis(x, center_S_k)]}$ is the optimized weighting factor for pixel x in patch $S_i \in S(x)$, where $dis(x, center_S_k)$ is the spatial distance between pixel x and the patch center of S_i . Let $I_{x \in S_i}^{free}$ be the shadow-free result of pixel x by performing local illumination recovering operator using the matched patch pair (S_i, L_j) . The illumination consistency optimization result for pixel x is computed as weighted average of all the possible shadow-free values $I_{x \in S_i}^{free}$ in each $S_i \in S(x)$:

$$I_x^{free} = \frac{\sum_{S_i \in S(x)} w_i I_{x \in S_i}^{free}}{\sum_{S_i \in S(x)} w_i} \quad (9)$$

After the procedure of illumination consistency optimization, the inconsistent artifacts of the neighboring patches can be avoided or greatly alleviated, and we receive high-quality shadow-free result, as illustrated in Fig. 5(c). In our experiment, we set denser and smaller patches on the shadow boundaries, which helps to receive satisfactory result on the boundaries.

2) Texture Detail Enhancement: The illumination consistency optimization makes consistent transition between shadow removed patches. However, due to the weighted averaging, sometimes it will lead to some texture blurring artifacts, especially for image with weak texture structure, as illustrated in Fig. 6(b). Furthermore, to recover the illumination information of the heavy shadow regions, the texture detail may not be recovered thoroughly.

In order to recover texture details, we apply the image gradient of the original shadow regions as the guidance to enhance the received shadow-free result with the purpose to maintain gradient information of shadow regions. We define a optimization function as follows:

$$\sum_{x \in S} (I_x^{detail} - I_x^{free})^2 + \lambda_1 \sum_{x \in S} (\nabla I_x^{detail} - \nabla I_x)^2 \quad (10)$$

The first term of this energy function is the data term. I_x^{detail} is the target value at pixel x . I_x^{free} is the shadow removed value generated from illumination consistency optimization. The second term is the gradient constraint term, whose purpose is to maintain the gradient information of the shadow regions. ∇ is the gradient operator. λ_1 is a user controlled parameter used for balancing the contribution of the gradient constraint term. A large λ_1 is set when the shadow regions show significant blurring artifacts.

Because of the influence of shadow, especially heavy shadow, the gradient may be weakened in shadow regions. Hence, we set a weight coefficient λ_2 for the original gradient in Equation (10), whose objective is to compensate for the gradient weakening derived from light occluding. The value of λ_2 is controlled by user, which is fixed for one image. But for different images, a large λ_2 can be set when the color differences between adjacent pixels in shadow regions are small. The optimization function is rewritten as:

$$\sum_{x \in S} (I_x^{detail} - I_x^{free})^2 + \lambda_1 \sum_{x \in S} (\nabla I_x^{detail} - \lambda_2 \nabla I_x)^2 \quad (11)$$

We solve the above linear system using gradient decent method. As shown in Fig. 6, with the texture detail recovering enhancement, the texture details in shadow regions can be efficiently recovered. The step for texture detail enhancement is optional, and it is required only when texture blurring occurs after illumination consistency optimization. This blurring is a relatively rare event. In all the results presented in this paper, only Fig. 6(d) and the fourth column in Fig. 15 need such optimization.

D. Shadow Boundary Processing

Our method can effectively recover the illumination around the shadow boundaries where there is smooth transition between shadow regions and lit regions, as illustrated in the

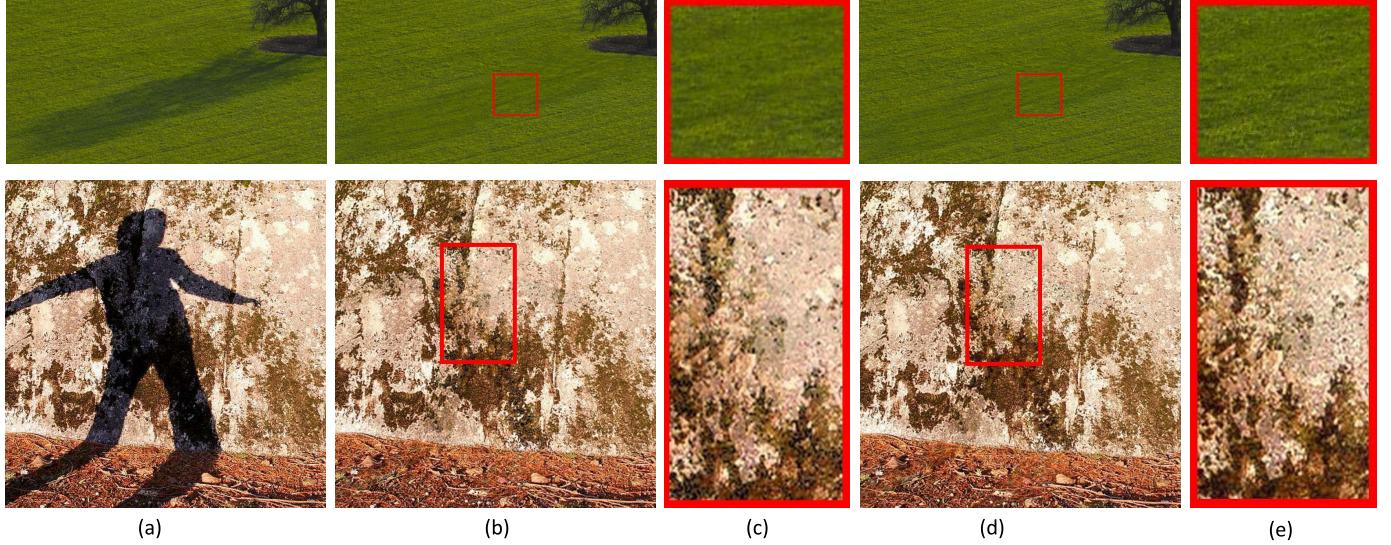


Fig. 6. Texture detail enhancement. (a) Input images, (b) shadow removal results without texture detail covering enhancement, (c) close-ups for the red boxes in (b), (d) shadow removal results with texture detail recovering, where $\lambda_1 = 1$, $\lambda_2 = 2.0$ in the first row and $\lambda_1 = 0.5$, $\lambda_2 = 0.8$ in the second row, (e) close-ups for the red boxes in (d).

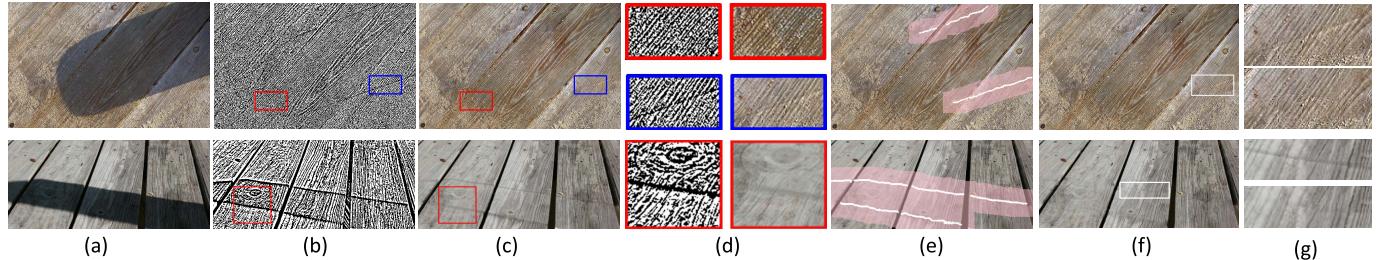


Fig. 7. Shadow boundary processing. (a) Input images with shadows, (b) the image details, (c) shadow removal results without shadow boundary processing, (d) close-ups for the red boxes and the blue boxes, (e) the trimaps for shadow boundary regions and sample regions, white regions are the target shadow boundaries and the pink regions are the sample regions, (f) shadow removal results with shadow boundary processing, (g) close-ups for white box regions, where the bottom and top close-ups correspond to the results with and without shadow boundary processing, respectively.

boxes of the first row in Fig. 7(c). But for some complex and sharp shadow boundaries, as shown in the second row in Fig. 7(c), our current method may not work well. The main difficulty in shadow boundaries is that some of the detail information is lost at shadow boundaries due to the dramatic changes of illumination on shadow boundaries, which leads to some defects for some existing shadow removal methods.

Let I be the input image, and I^{filter} is the filtered map by bilateral filter [34], we compute the image detail by $D = I - I^{filter}$. We can observe from Fig. 7(b) that, the detail information on the shadow boundaries is sometimes seriously destroyed or missing. Thus, to receive satisfied shadow-free results on sharp shadow boundaries, different from the linear interpolation method such as [5], we need to develop more effective shadow boundary processing techniques. Compared with the whole image, the target shadow boundary regions are relatively small. The example-based texture synthesis method can handle it very well. Inspired by the texture synthesis method [35], we present the constrained texture synthesis to recover the texture and illumination information on the shadow boundaries.

Let ϕ be the target shadow boundary regions. For each pixel in ϕ , T_x is a window centered at pixel x with size of $r \times r$. To remove transitional difference between lit regions and shadow removed regions, we minimize the following objective function to recover boundary texture and illumination:

$$E(I_x^{edge}, \{T_x\}) = \sum_{x \in \phi} (||T_x - M|| + \omega \frac{||T_x - M||^2}{d(x)^2 + c(W)^2}) \quad (12)$$

where I_x^{edge} is the target value at pixel x . The first term measures the appearance difference between T_x and M in a L_1 norm fashion. M is a $r \times r$ window in the sample regions. In our experiments, we set $r = 7$. The second term is the proximity term constraining the search space. ω is the balance parameter. $d(x)$ is the distance between pixel x and the boundary of the lit region and $c(W) = \frac{W}{8}$ is the strength parameter for adjusting the proximity constraint. W is the largest image dimension (image width or height).

We constrain the sample regions near the target shadow boundary regions, and obtain the sample regions by dilating the mask of ϕ , such as the pink regions in Fig. 7(e). We apply a two-step iterative method to get the optimized results. For

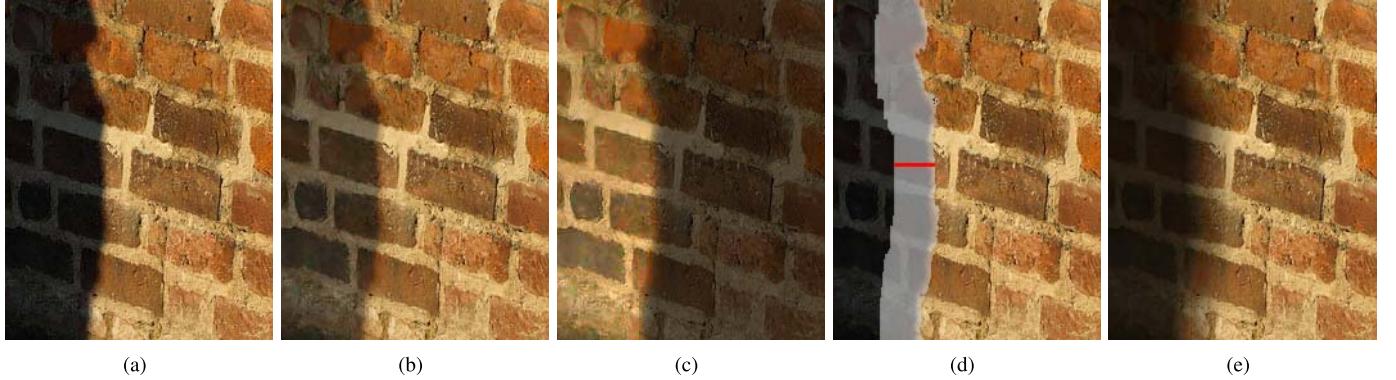


Fig. 8. Shadow editing. (a) Input image, (b) the direction illumination of the shadow areas are tuned by setting $\eta = 1/2$, $v = 1$, (c) shadow editing result, where $\eta = 1$, $v = 1$ in the shadow areas and $\eta = 1/5$, $v = 1$ in the lit areas, (d) the specified transition region for shadow edge softening, (e) shadow edges softening result with the attenuation function $f_x = \sqrt{1 - d_x}$, where d_x is the normalization distance between the pixel I_x and the shadow boundary.

Algorithm 1 Shadow Removal Algorithm

Input: RGB image I

Output: Shadow-free RGB image

- 1: Detect the shadow regions (Section III)
 - 2: Decompose the input image into adaptive patches (Section V-A)
 - 3: **for** each patch $S_i \in S$ **do**
 - 4: Patch matching: Find an optimal matching patch L_j in L (Section V-B)
 - 5: Patch shadow removal: Remove the shadow in patch S_i applying the Local illumination recovering operator on patch pair (S_i, L_j) (Section IV)
 - 6: **end for**
 - 7: Illumination optimization: perform weighted average for the overlapped pixels (Section V-C)
 - 8: Texture details recovering (optional) (Section V-C)
 - 9: Boundary processing (optional) (Section V-D)
-

each iteration, we first find T_x by minimizing Equation (12), then evaluate the value for each pixel x in ϕ . We repeat the two steps iteratively until satisfying the convergence value. Based on these two terms, for each patch in target shadow boundary regions, we can find a nearest patch in sample regions, which will be used for synthesizing the textures in target shadow boundary regions. As illustrated in Fig. 7 (f), with the controllable texture synthesis on shadow boundaries, the illumination and texture details are effectively recovered, and the results are also consistent with the surrounding regions.

In algorithm 1, we outline the main steps of the proposed shadow removal algorithm. Each step has been detailed in the previous sections. In our shadow removal system we only need to manually set three parameters: ω , λ_1 and λ_2 .

VI. APPLICATIONS

Our illumination recovering operator can be easily extended to image editing applications, such as shadow editing and color transfer.

1) *Shadow Editing*: According to the image formation model [30], one pixel in an image can be represented as: $I_x = (\eta L^d + v L^a) R_x$, the parameter η , v depend on the light

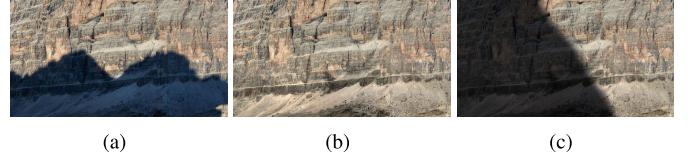


Fig. 9. Image shadow removal and editing based on illumination recovering optimization. (a) Input image, (b) shadow removal result, (c) shadow editing for specific area with $\eta = 0.1$, $v = 0.4$.

conditions and can be considered as the light attenuation factor or the object occluding factor. In the lit areas, η and v are defined as 1. In the umbra areas, η is 0 and v is 1. To remove the shadow, we add the direct illumination to the shadow areas, that is, setting $\eta = 1$ in the shadow areas. We can also modify the value of η and v in specified areas for producing new shadow editing results, which corresponds to change the direct illumination or indirect illumination for the specified area.

We first specify a shadow sample and a nonshadow sample with similar texture. Then we utilize the illumination recovering operator and the relationship between L^d and L^a : $L^d = t L^a$ for shadow editing, where t is computed based on the shadow sample and nonshadow sample in the input image. By setting the direct illumination and the indirect illumination, the intensity of the specified areas can be expressed as: $I_x^{edit} = \frac{\eta t + v}{\eta t + 1} I_x$. In Fig. 8 and Fig. 9, we give several shadow editing results by setting different values for η and v . Compared with the shadow editing method [3] based on Poisson shadow interpolation and the illumination editing method [36], our method is easier to simulate a variety of lighting conditions.

Our illumination recovering operator can also be applied to soften the sharp shadow edges, as illustrated in Fig. 8(e). Softening shadow is generated with illumination transition from the shadow areas to the lit areas. This process can be simulated using our method. As shown in Fig. 8(d), we first specify a transition region around the shadow edges. For the transition region, we define a direct illumination attenuation function f_x which varies depending on the distance between pixel x and the shadow edges. Then the shadow edge softening can be achieved by performing the following illumination

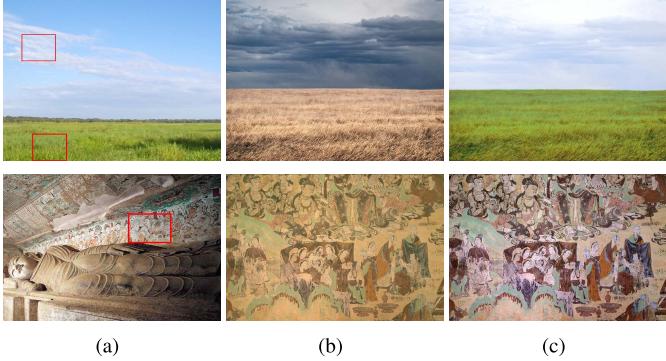


Fig. 10. Color transfer. (a) Sample images, (b) input target images, (c) color transfer results, where color of the red box in the sample image is transferred to the target image.

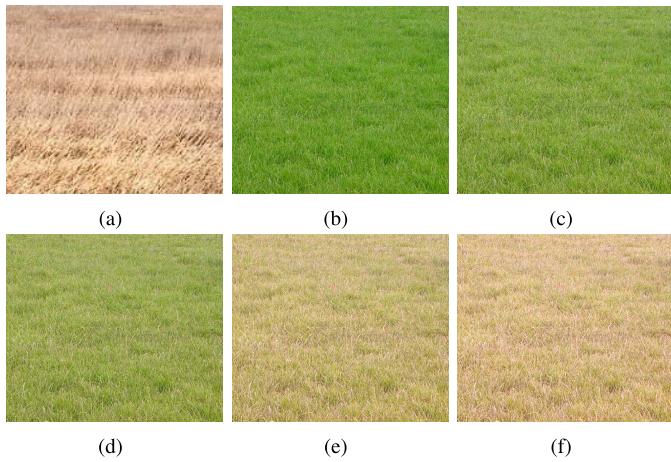


Fig. 11. Color transfer. (a) Target color image, (b) input image, (c) color transfer result with $g_x = 0.2$, (d) color transfer result with $g_x = 0.4$, (e) color transfer result with $g_x = 0.7$, (f) color transfer result with $g_x = 0.9$.

recovering: $I_x^{edit} = (f_x L^d + L^a) R^x = \frac{f_x t + v}{at + 1} I_x$, where f_x is the normalized light attenuation function, and $0 \leq f_x \leq 1$. The user can develop more sophisticated attenuation function to produce more physically realistic results.

2) *Color Transfer*: Color and tone transfer is a popular topic in computer graphics [20], [37]. Our illumination recovering operator can also work on color transfer. Given the sample image, we can transfer its color to the target image. More specially, to transfer the color information of the sample image to the target areas, we apply our illumination recovering operator on these areas. The color transfer result for the target result can be written as: $I_x^{edit} = (t + 1) I_x$, where I_x denotes the RGB components of the original image. Fig. 10 shows two examples for color transfer.

We can also simulate the color transfer processing in animated style. We add a color transfer parameter g_x to the color transfer equation, and g_x is assigned by user according to different applications. It also can be set as a function. Then, the color transfer process can be expressed as: $I_x^{edit} = (g_x t + 1) I_x$. As shown in Fig. 11, there are some color transfer results by setting different values for g_x , which can simulate the process for grass from green to yellow.

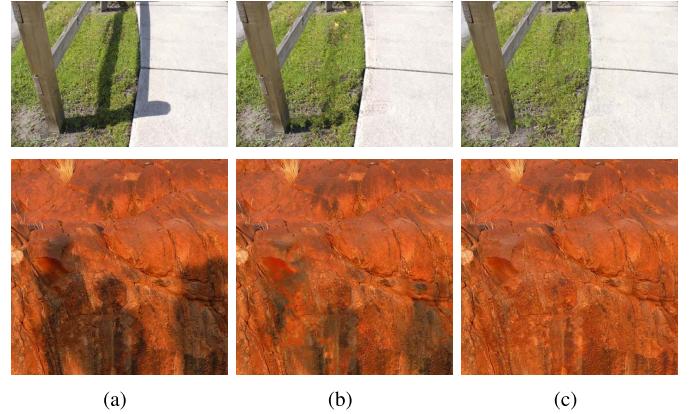


Fig. 12. Shadow removal results. (a) Input images, (b) results of [8], (c) our results.

VII. EXPERIMENTS AND DISCUSSIONS

In this section, we perform our shadow removal method on a variety of shadow images to illustrate the effectiveness of the proposed approach. We also present comparisons with the most related shadow removal methods. The limitations of the proposed method are also given. All our results are implemented using C++ on a machine equipped with Pentium(R) Dual-Core CPU E5200@2.50GHz with 2GB RAM.

In Fig. 12, we compare our method with the approach [8]. Guo *et al.* [8] segment the input image using texton histogram and SVM, but the divided regions are irregular. Large regions may contain several different kinds of colors and textures, which leads to calculation error for the ratio between direct light and environment light. This may result in an unsatisfactory shadow-free result, as shown in Fig. 12(b). Our method introduces adaptive overlapped patches which alleviate the problem of inaccurate matching. Meanwhile, we calculate different ratio values between direct illumination and environment illumination for different patches, but Guo *et al.* [8] uses the paired regions to compute a fixed ratio value for an image. The fixed ratio is inappropriate for the shadow images with complex textures, which makes the shadow-free result unnatural, such as the result in Fig. 12(b).

In Fig. 13, we present shadow removal results for boundary treatment. As illustrated in Fig. 7, the texture details under the shadow boundaries may be blurred. We apply the texture and illumination optimization to recover the information of the shadow boundaries, and also present comparison results with previous methods. Xiao *et al.* [5] processes the shadow boundary using alpha matte interpolation. As the interpolation process depends on the original texture details at the shadow boundary regions, this method cannot process complex shadow boundaries with loss of details. Shor and Lischinski [10] utilizes graph-cut based texture technique to repair the shadow boundaries. This method works well for image with rich texture, otherwise, the results will not be satisfied. The method of [1] needs to locate the boundary precisely, and inaccurate shadow edges lead to fuzzy transition from interior shadow regions to nonshadow regions. Compared with aforementioned methods, our results are more visually natural and consistent with the surrounding content.

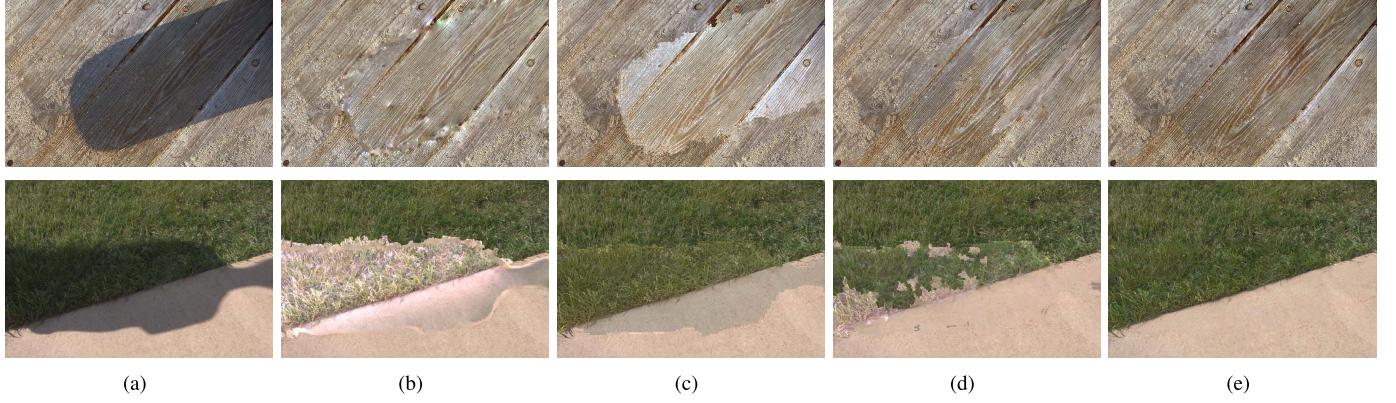


Fig. 13. Boundary processing results. (a) Input images, (b) results of [1], (c) results of [10], (d) results of [5], (e) our results.

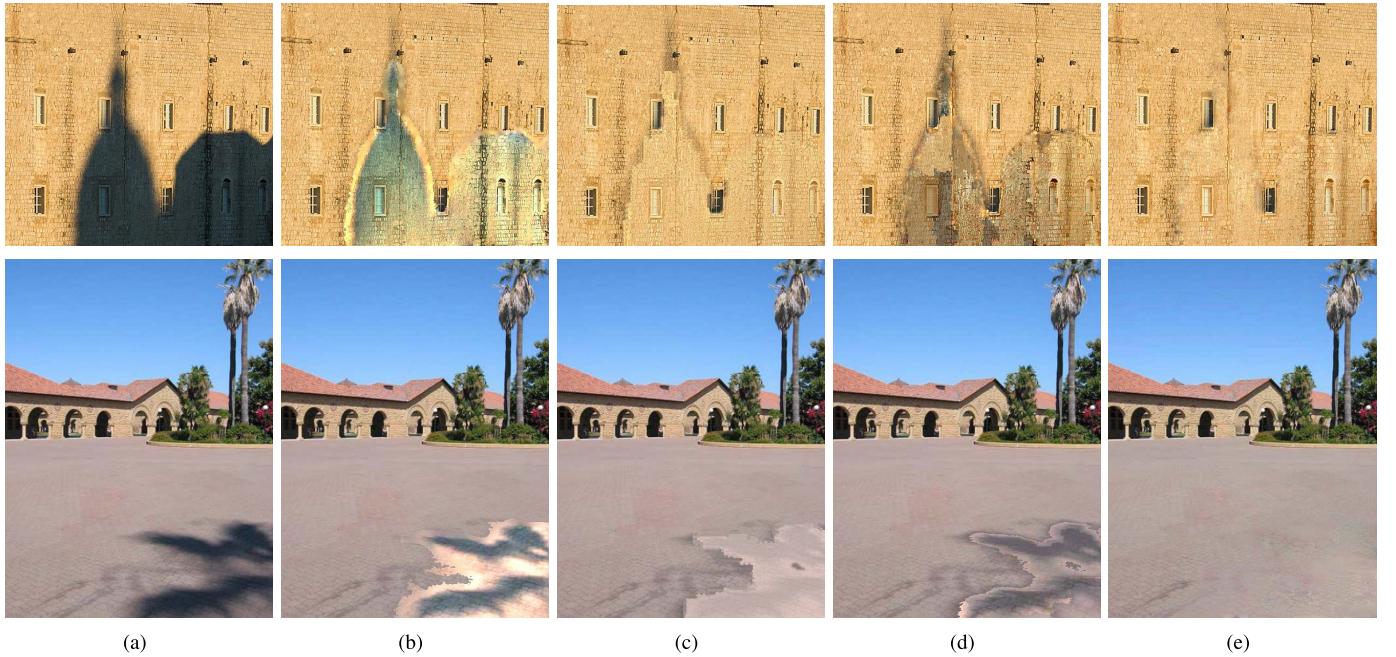


Fig. 14. Nonuniform shadow removal result. (a) Input image, (b) result of [1], (c) result of [10], (d) results of [5], (e) our result.

The results in Fig. 14 demonstrate that our method is capable of dealing with nonuniform shadows. Note that in these images, the shadow intensity vary among the umbra, penumbra, and lit regions. In Fig. 14, we compare our method with the related works [1], [5], [10] for handling nonuniform shadows. Since the relatively wide soft shadow on the shadow boundaries, the accurate shadow edges are difficult to be detected out. This makes Finlayson et al. [1] impossible to recover the illumination in shadow regions. Shor and Lischinski [10] use fixed parameters for whole shadow regions. But the illumination is variable for soft shadows, it make this method failed to deal with nonuniform shadows. The unsatisfied results in Fig. 14(d) demonstrate Xiao et al. [21] also cannot treat nonuniform shadows. By using adaptive texture patch matching, our method processes each small patch individually and remove the nonuniform shadows completely. In addition, the recovered illumination of nonuniform shadow regions is consistent with the surrounding environment.

In Fig. 15, we give shadow removal results for shadow images with complex textures, and compare our method

with [1], [5], [8], [10]. These images contain multiple textures or materials. Shor and Lischinski [10] fails to take texture types into account in shadow regions. This make the wrong recovered color in shadow regions, especially for images with large color variation between differen textures. The shadow removal method [10] for simple texture image is not applicable to complex texture images. Xiao et al. [5] take the texture classification into account and divides the shadow areas into several parts. The independent processing for each part induces inconsistency between subregions, as shown in the fifth row in Fig. 15. Neither [1] nor [8] works well in removing shadows with complex texture or material content. Note that for the results of the sixth row in Fig. 15, even the shadow regions have a variety of texture types, our method can effectively recover the texture information in the shadow regions. In addition, the illumination on the shadow boundaries is also effectively recovered, and the results are consistent with the surrounding regions.

Our method can also be applied to aerial remote sensing shadow images. In Fig. 16, we compare our method to [5] and [6] on aerial remote sensing images. The texture

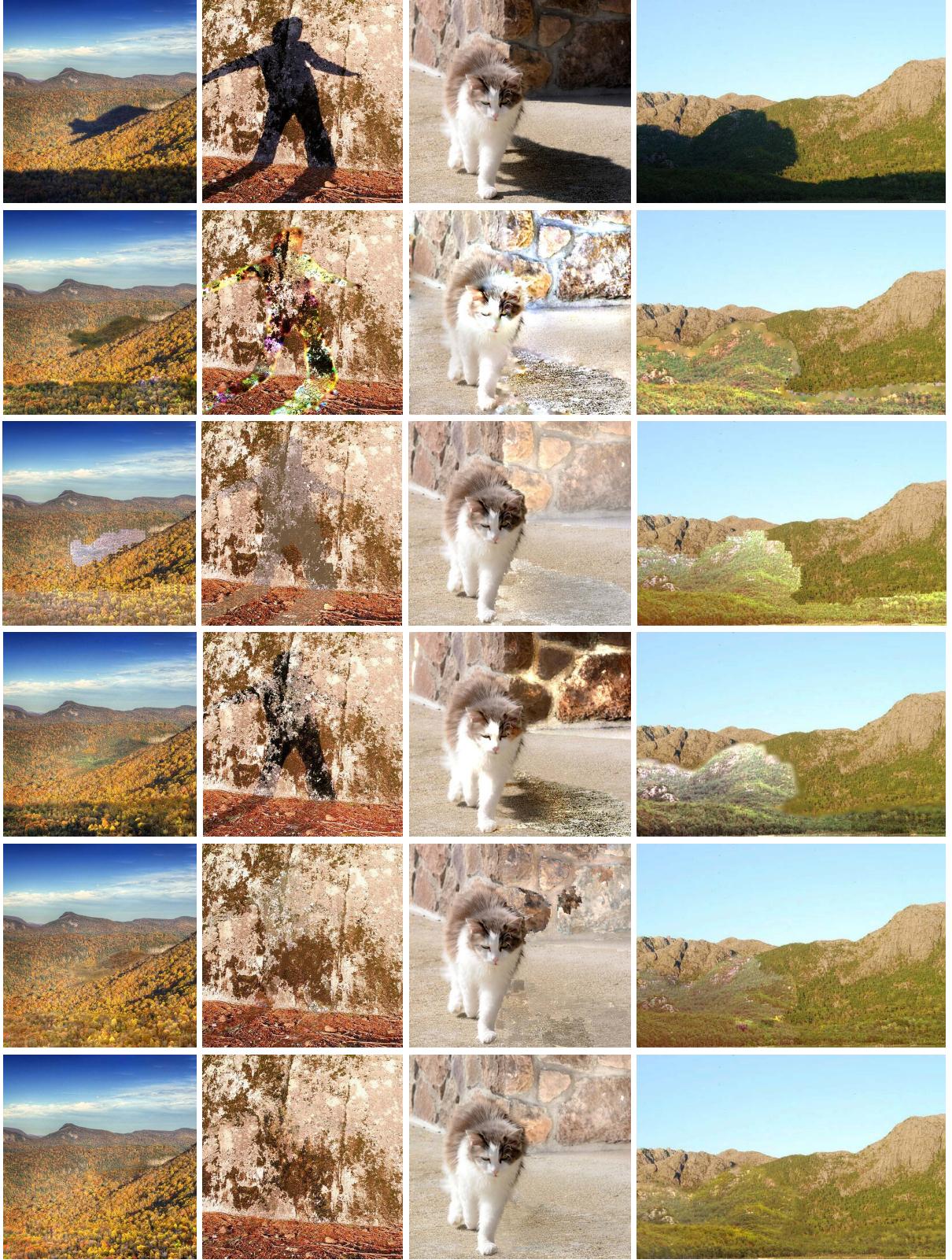


Fig. 15. Shadow removal results for complex textures. The first row: input images, the second row: results of [1], the third row: results of [10], the fourth row: results of [8], the fifth row: results of [5], the sixth row: our results, the patch size is 15×15 , 20×20 , 30×30 , 21×21 , respectively.

descriptor used in [5] is not illumination independent which is not suited for subregion matching between shadow regions and nonshadow regions. Meanwhile, as demonstrated by [38], high resolution aerial images often contain heavy noises.

The linear shadow-free algorithm [5] does not work well on aerial shadow image, as illustrated in Fig 16(b). Li et al. [6] removed the shadows of image using spatially adaptive NL operators. To reduce the influences of abundant texture

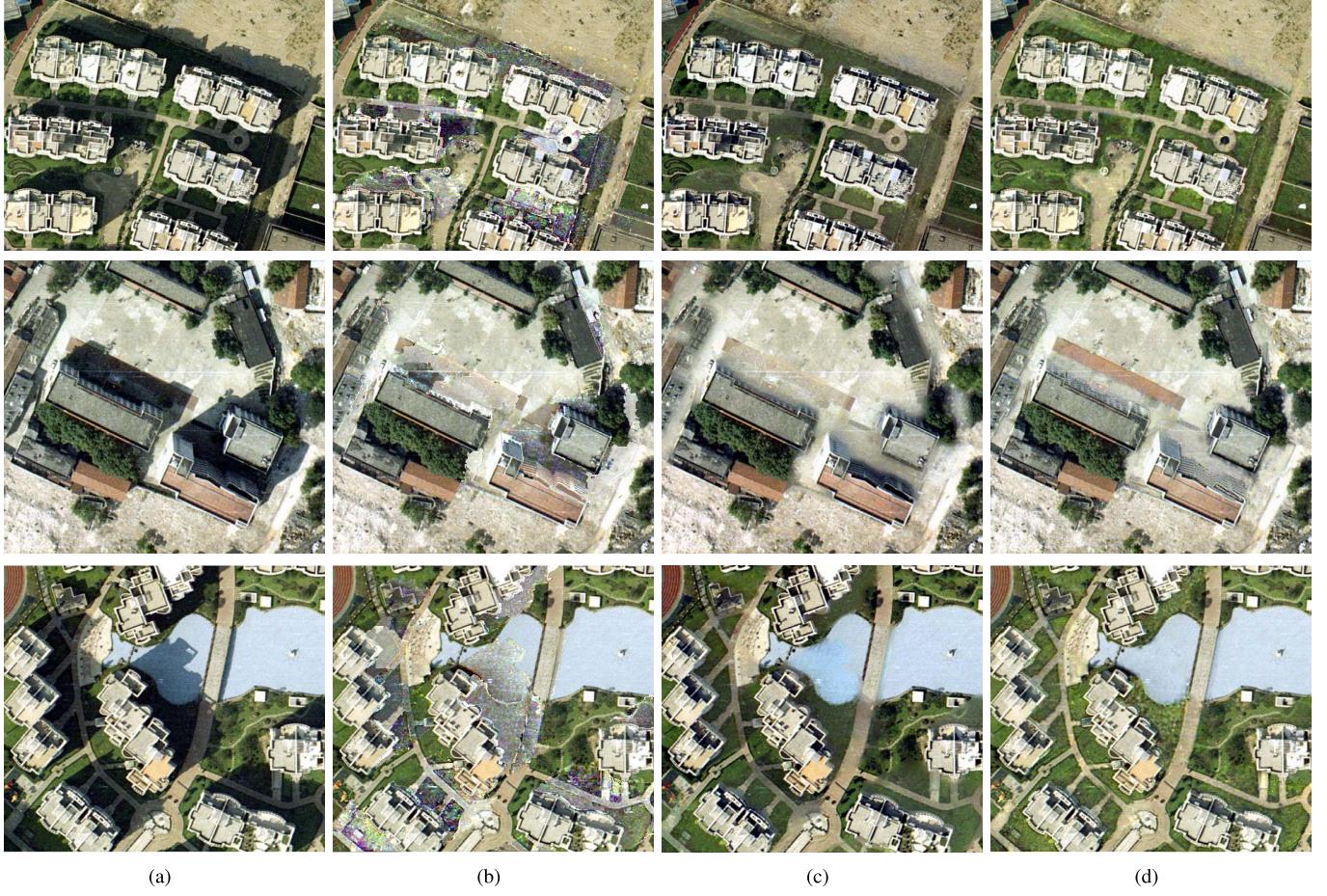


Fig. 16. Aerial remote sensing shadow-free results. (a) input images, (b) results of [5], (c) results of [6], (d) our results.

and noise existing in aerial remote sensing images, they use NL Laplacian prior to the regularization terms in the optimizing function. This introduces texture detail blurring for the shadow-free result. As shown in Fig. 16(c), the results are over smoothed. Furthermore, some shadows are not removed completely. Our method has a better matching accuracy which produces natural shadow free results, and can effectively recover the texture details in shadow regions which benefit the further image analysis.

In Fig. 17, we compare our shadow removal results with ground truth images. We fix the camera on a tripod in the sun. Thereafter, we capture the shadow image and ground truth image with an occluder coming in and out. As illustrated in Fig. 17, both the illumination and texture of our results are very close to those of the ground truth images.

The time consumption of the proposed method depends on the size of the shadow regions and the number of the sample patches in the lit regions. The time is mainly spent in patch matching, and the shadow removal processing. Typically, for an image with size of 704×647 (for example, the third row in Fig. 15), we decompose the shadows into 231 patches, and there are 236 lit patches. It takes about 0.9 seconds for performing the patch matching, and takes about 8.2 seconds for performing the shadow removal.

Limitations: For some special shadow images, the pixels in the shadow patch are zero. In this case, it is difficult to find

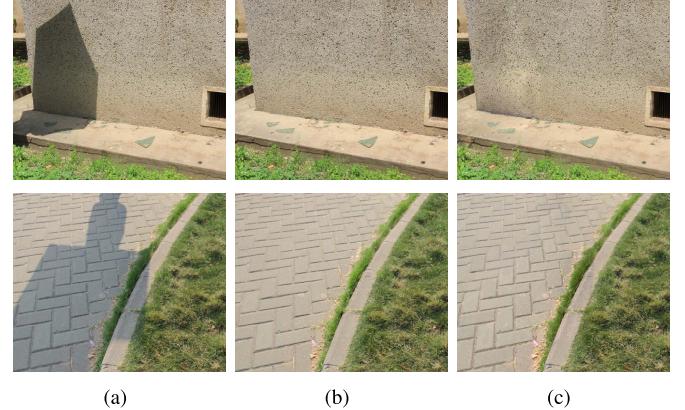


Fig. 17. Shadow removal results. (a) Input images, (b) ground truth, (c) our shadow removal results.

a corresponding patch in the lit regions. Furthermore, for the shadow images with heavy noise, the illumination recovered results may also contain some noises. Another limitation of our method is that when the textures of shadow regions and lit regions are quite different, there is no correctly matched lit patch for shadow patch. In this situation, if we apply the lit patch as the sample to remove shadows, the texture and illumination in the shadow regions cannot be accurately estimated, which induces to unnatural shadow removal results.



Fig. 18. Limitation. (a) Input image, (b) our shadow removal result.

As shown in Fig. 18, the shadow regions are mountain and road, but the lit regions are sky and clouds, in this case, an unsatisfied image is obtained using the inappropriate matched information.

VIII. CONCLUSION

In this paper, we have presented a novel shadow removal approach using an illumination recovering optimization operator. We adaptively decompose the input image into overlapped patches according to the shadow distribution. By building the correspondence between the shadow patch and the lit patch, we construct an illumination recovering operator which effectively remove the shadows and obtain natural transition result between patches through coherence optimization process. Our shadow removal system is simple and effective, and can handle a variety of complex shadows, including images with both sharp shadows and soft shadows.

In the current version, we perform image decomposition using adaptive method. In the future, we would like to apply the Poisson disk [39] to generate adaptive patches, which is edge-aware, thus, our method may produce more accurate and pleasing shadow removal results. When we consider video sequence as video cube, our adaptive image decomposition, local illumination recovering optimization and nearest patch search can be easily extended to video data. In the future, we would like to extend our method to video shadow removal and editing. Another potential research topic is that, we can apply the histogram matching method used in the texture optimization to eliminate the possible texture blurring occurred in our illumination optimization.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions. They also thank H. Li, L. Zhang, and H. Shen for presenting aerial remote sensing shadow images used in Fig. 16. Chunxia Xiao is the corresponding author.

REFERENCES

- [1] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 59–68, Jan. 2006.
- [2] E. Arbel and H. Hel-Or, "Shadow removal using intensity surfaces and texture anchor points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1202–1216, Jun. 2011.
- [3] A. Mohan, J. Tumblin, and P. Choudhury, "Editing soft shadows in a digital photograph," *IEEE Comput. Graph. Appl.*, vol. 27, no. 2, pp. 23–31, Mar./Apr. 2007.
- [4] I. Boyadzhiev, S. Paris, and K. Bala, "User-assisted image compositing for photographic lighting," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 96–96, 2013.
- [5] C. Xiao, D. Xiao, L. Zhang, and L. Chen, "Efficient shadow removal using subregion matching illumination transfer," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 421–430, Oct. 2013.
- [6] H. Li, L. Zhang, and H. Shen, "An adaptive nonlocal regularized shadow removal method for aerial remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 106–120, Jan. 2014.
- [7] J. Zhu, K. G. G. Samuel, S. Z. Masood, and M. F. Tappen, "Learning to recognize shadows in monochromatic natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 223–230.
- [8] R. Guo, Q. Dai, and D. Hoiem, "Single-image shadow detection and removal using paired regions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2033–2040.
- [9] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum, "Natural shadow matting," *ACM Trans. Graph.*, vol. 26, no. 2, pp. 1–8, 2007.
- [10] Y. Shor and D. Lischinski, "The shadow meets the mask: Pyramid-based shadow removal," *Comput. Graph. Forum*, vol. 27, no. 2, pp. 577–586, Apr. 2008.
- [11] F. Liu and M. Gleicher, "Texture-consistent shadow removal," in *Computer Vision—ECCV*. New York, NY, USA: Springer-Verlag, 2008, pp. 437–450.
- [12] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 918–923, Jul. 2003.
- [13] J.-B. Huang and C.-S. Chen, "Moving cast shadow detection using physics-based features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2310–2317.
- [14] Q. Yang, K.-H. Tan, and N. Ahuja, "Shadow removal using bilateral filtering," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4361–4368, Oct. 2012.
- [15] D. Miyazaki, Y. Matsushita, and K. Ikeuchi, "Interactive shadow removal from a single image using hierarchical graph cut," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2010, pp. 234–245.
- [16] Y. Weiss, "Deriving intrinsic images from image sequences," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 68–75.
- [17] Y.-Y. Chuang, D. B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski, "Shadow matting and compositing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 494–500, 2003.
- [18] G. D. Finlayson, S. D. Hordley, and M. S. Drew, "Removing shadows from images," in *Proc. ECCV*, vol. 28, no. 1, 2002, pp. 823–836.
- [19] G. D. Finlayson, M. S. Drew, and C. Lu, "Intrinsic images by entropy minimization," in *Computer Vision—ECCV*. New York, NY, USA: Springer-Verlag, 2004, pp. 582–595.
- [20] E. Reinhard, M. Adhikmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.
- [21] C. Xiao, R. She, D. Xiao, and K.-L. Ma, "Fast shadow removal using adaptive multi-scale illumination transfer," *Comput. Graph. Forum*, vol. 32, no. 8, pp. 207–218, Dec. 2013.
- [22] T.-P. Wu and C.-K. Tang, "A Bayesian approach for shadow extraction from a single image," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 480–487.
- [23] Y. Xiao, E. Tsougenis, and C.-K. Tang, "Shadow removal from single RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3011–3018.
- [24] M. Baba, M. Mukunoki, and N. Asada, "Shadow removal from a real image based on shadow density," in *Proc. ACM SIGGRAPH Posters*, 2004, p. 60.
- [25] C. Fredembach and G. Finlayson, "Simple shadow removal," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, 2006, pp. 832–835.
- [26] V. J. D. Tsai, "A comparative study on shadow compensation of color aerial images in invariant color models," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1661–1671, Jun. 2006.
- [27] A. Makarau, R. Richter, R. Muller, and P. Reinartz, "Adaptive shadow detection using a blackbody radiator model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2049–2059, Jun. 2011.
- [28] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.
- [29] C. Xiao, M. Liu, D. Xiao, Z. Dong, and K.-L. Ma, "Fast closed-form matting using a hierarchical data structure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 49–62, Jan. 2014.
- [30] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images," in *Comput. Vis. Syst.*, A. Hanson and E. Riseman, Eds. New York, NY, USA: Academic, 1978.

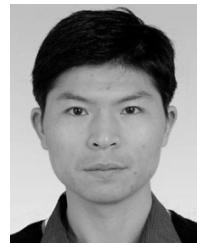
- [31] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Computer Vision—ECCV*. Berlin, Germany: Springer-Verlag, 2006, pp. 589–600.
- [32] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graph.*, vol. 32, no. 6, p. 176, Nov. 2013.
- [33] X. Hong, H. Chang, S. Shan, X. Chen, and W. Gao, "Sigma set: A small second order statistical region descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1802–1809.
- [34] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.
- [35] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 795–802, Jul. 2005.
- [36] X. Mei, H. Ling, and D. W. Jacobs, "Illumination recovery from image with cast shadows via sparse representation," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2366–2377, Aug. 2011.
- [37] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 637–645, 2006.
- [38] Q. Zhang and C. Xiao, "Cloud detection of RGB color aerial photographs by progressive refinement scheme," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7264–7275, Nov. 2014.
- [39] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.



Ling Zhang received the B.Sc. degree in computer science and technology from Wuhan Donghu University, in 2009, and the M.Sc. degree in instructional technology from Central China Normal University, in 2012. She is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University, China. Her research interests include image and video editing, and computational photography.



Qing Zhang received the B.Sc. and M.Sc. degrees in computer science and technology from Wuhan University, Wuhan, China, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with School of Computer Science, Wuhan University. His research interests include image and video processing.



Chunxia Xiao (M'11) received the B.Sc. and M.Sc. degrees from the Mathematics Department, Hunan Normal University, in 1999 and 2002, respectively, and the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, in 2006. He is currently a Professor with the School of Computer Science, Wuhan University, China. From 2006 to 2007, he held a post-doctoral position with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, and from 2012 to 2013, he visited the University of California-Davis for one year. His main interests include image and video processing, digital geometry processing, and computational photography.