Texture-Consistent Shadow Removal

Feng Liu and Michael Gleicher

Computer Sciences Department, University of Wisconsin-Madison
{fliu,gleicher}@cs.wisc.edu

Abstract. This paper presents an approach to shadow removal that preserves texture consistency between the original shadow and lit area. Illumination reduction in the shadow area not only darkens that area, but also changes the texture characteristics there. We achieve texture-consistent shadow removal by constructing a shadow-free and texture-consistent gradient field. First, we estimate an illumination change surface which causes the shadow and remove the gradients it induces. We approximate the illumination change surface with illumination change splines across the shadow boundary. We formulate estimating these splines as an optimization problem which balances the smoothness between the neighboring splines and their fitness to the image data. Second, we sample the shadow effect on the texture characteristics in the umbra and lit area near the shadow boundary, and remove it by transforming the gradients inside the shadow area to be compatible with the lit area. Experiments on photos from *Flickr* demonstrate the effectiveness of our method.

1 Introduction

Shadow removal is often required in digital photography as well as in many vision applications. For clarity, we define the problem of shadow removal at the very beginning. Following previous work [1,2,3], an image $\mathcal{I}$ can be represented as the composition of the reflectance field $\mathcal{R}$ and the illumination field $\mathcal{L}$ as follows:

$$\mathcal{I}(x,y) = \mathcal{R}(x,y) \cdot \mathcal{L}(x,y)$$

A shadow image can be formulated by applying an illumination change surface $\mathcal{C}(x,y)$ to the illumination field as follows:

$$\tilde{\mathcal{I}}(x,y) = \mathcal{I}(x,y) \cdot \mathcal{C}(x,y) \tag{1}$$

$$\tilde{I}(x,y) = I(x,y) + C(x,y) \tag{2}$$

where Equation 2 is Equation 1' counterpart in the log domain. $\tilde{I}$, I and C are the logarithms of $\tilde{\mathcal{I}}$, $\mathcal{I}$ and $\mathcal{C}$ respectively. $\mathcal{C}$ is usually assumed to be 1 in the lit area, a constant $c(\in (0,1))$ in the umbra area, and changing from c to 1 in the penumbra area. This paper works in the log domain.

The normal goal of the shadow removal is to estimate the illumination change surface $\mathcal{C}$ from $\tilde{\mathcal{I}}$ and recover the shadow free image $\mathcal{I}$ or I. It is important to

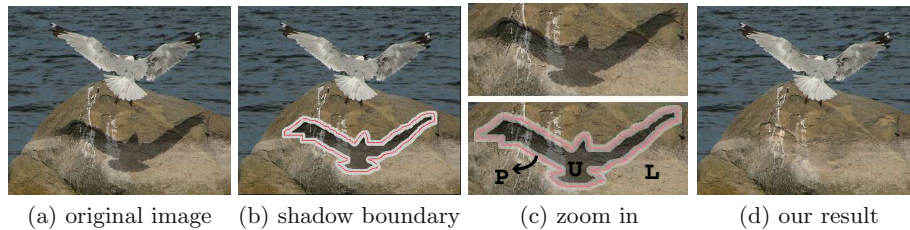| (a) original image | (b) shadow boundary | (c) zoom in | (d) our result |

Fig. 1. Given a rough shadow boundary 'P' provided by users (b and c), our algorithm removes the shadow (d). The red curve inside the brush stroke is the trajectory of the brush center. Users do not need to provide a precise shadow boundary as shown in (c) (Notice the eagle's right wing.). The brush strokes divide the image into 3 areas: definite umbra areas,'U', definite lit areas, 'L', and boundary, 'P', which contains the penumbra area as well as parts of the umbra and lit area.

examine how the illumination change surface $\mathcal{C}$ affects an image. Since an image can be reconstructed from its gradient field with proper boundary conditions, we focus on how $\mathcal{C}$ affects the gradient field in the log domain.

1. $\mathcal{C}$ will affect the gradients in the penumbra area where it is not uniform. Ideally, $\mathcal{C}$ will not affect the gradients in the umbra and lit area since it is uniform in these 2 areas, and is canceled off in calculating the gradients. However, this is not often true in practice as explained in the following.
2. In practice, the imaging process suffers from noise and quantization errors. Usually the signal to noise/quantization error ratio in the shadow area is lower than in the lit area. In this way, $\mathcal{C}$ makes the effect of noise/quantization error on the gradients in the shadow area more significant than in the lit area.
3. Normally, the poor lighting in shadow areas can weaken the texture, and even diminish the details. However, this is not always true for many images containing highly specular surfaces. If the illumination is strong in the scene, texture details in the lit area disappear; while in the shadow area, the reduction of the illumination can keep the textures there.
4. If the surface response curve has a different shape in the shadow and lit area, scaling up the shadow region to cancel $\mathcal{C}$ will change the texture characteristics.

From the above observations, we can see that applying the illumination change surface $\mathcal{C}$ not only affects the gradients in the penumbra area, it also affects the characteristics of the gradient fields in the whole shadow area. We call the former the shadow effect on the penumbra gradients and the latter the shadow effect on the gradient characteristics in the shadow area.

1.1   Previous Work

This paper focuses on removing shadows from a single image. Many methods have been presented to address this problem. Shadow removal is usually achieved
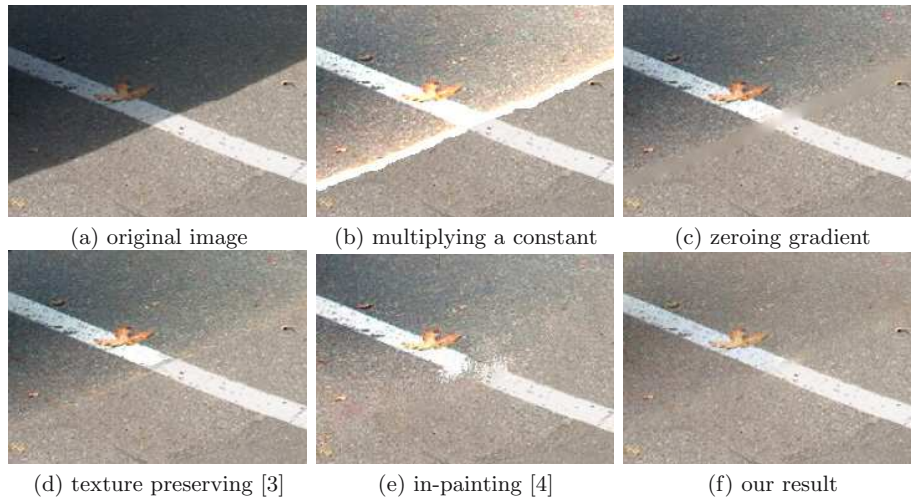
(a) original image   (b) multiplying a constant   (c) zeroing gradient

(d) texture preserving [3]   (e) in-painting [4]   (f) our result

Fig. 2. Motivating example. (b): multiplying constant to the image intensities inside the shadow region. (c): zeroing gradients inside the shadow boundary. (d): texture-preserving shadow removal [3]. (e): in-painting the shadow boundary region [4].

in two steps: shadow detection and image reconstruction. (Approaches to removing shadow using information from multiple images have also been presented(c.f. [2,5]).)

Many methods have been presented to automatically detect shadow regions. Finlayson *et al.* estimate an illumination invariant image based on an invariant color model, and use this invariant image together with the original image to locate the shadow region [6]. Similarly, Salvador *et al.* use invariant color features to segment cast shadows [7]. Levine and Bhattacharyya [8] study properties of color ratios across boundaries between regions in a segmented image, and use a support vector machine to identify shadow regions based on these color ratios. In digital photography, shadow boundaries are often specified through user interactions [9,10]. Our algorithm relies on users to roughly specify the shadow boundary, and refines it automatically.

Once shadow areas are located, they can be removed by multiplying a suitable scalar to the shadow pixels to cancel the effect of the illumination change surface $\mathcal{C}$. This easy method can create a noticeable over-saturated band in the penumbra area as illustrated in Fig. 2(b). To solve this problem, Baba *et al.* adapt the multiplicative scalars $\mathcal{C}$ based on shadow densities [11]. Recently, Arbel and Hel-Or estimate $\mathcal{C}$ by considering the surface geometry, and effectively remove the shadow while preserving texture in both umbra and penumbra areas [3], as illustrated in Fig. 2(d).

Alternatively, shadows can be removed by zeroing gradients in the penumbra area and applying 2D integration to reconstruct the shadow-free image [2,6].

These methods usually work in the log image domain. As shown in Fig. 2(c), zeroing gradients in the penumbra area nullifies the texture there, however. To solve this problem, in-painting techniques are applied to fill in the missing texture [12,4]. However, in-painting sometimes introduces inconsistent textures as illustrated in Fig. 2(e). Alternatively, Mohan *et al.* [10] estimate a soft shadow model in the penumbra area, and remove shadow effect in the gradient domain accordingly.

Although previous methods vary in estimating the illumination change surface $\mathcal{C}$, they share common ideas to reconstruct the shadow-free image in the umbra area: multiplying a constant scalar to cancel the effect of $\mathcal{C}$. Applying 2D integration in the log domain with proper boundary conditions is equivalent to multiplying a constant in the image domain. This scheme can effectively match the overall illumination in the umbra area to that in the lit area. And using proper scalar constants to the penumbra area can also cancel the shadow effect on the penumbra area. However, these methods can not remove the shadow effect on the texture characteristics of the shadow area. Multiplying a constant can magnify the noise and quantization error in the original shadow region. For particular images with strong specular surface and strong lighting, the details in the shadow area, which disappear in the lit area, will be enhanced. All these lead to inconsistent texture between the shadow area and lit area. For example, the texture in the shadow area in Fig. 2(c), (d) and (e) is not compatible with that in the lit area.

1.2   Our Contribution

In this paper, we present a shadow removal method that preserves texture consistency. Since textures manifest themselves by image gradients, our algorithm works in the gradient domain. Specifically, we construct a new image gradient field that removes the shadow effects on both the gradients in the penumbra area and the characteristics on the gradients in the whole shadow area. From this new image gradient field, we can reconstruct the shadow-free image by solving a Poisson equation.

Our major contribution is a method for constructing a shadow-free and texture-consistent gradient field by removing the two-fold shadow effects on the gradient field as mentioned previously. First, we simultaneously locate the penumbra area and estimate the illumination change curves across the shadow boundary by estimating and sampling the illumination change surface using line segments. With the illumination change curves, we can cancel the effect of shadow on the gradient field in the penumbra area. Second, we estimate the shadow effect on the gradient distribution in the shadow area, and transform the gradient field there to cancel the effect to be consistent with that in the lit area. In this way, we obtain the texture-consistent gradient field. These two key algorithms are detailed in § 2. Experiments on photos from *Flickr* demonstrate the effectiveness of our algorithm as detailed in § 3.

2   Texture-Consistent Shadow Removal

In this paper, we provide a brush tool for users to mark the shadow boundary. As illustrated in Fig. 1(c), users can select a brush with much larger size than the boundary, and do not need to delineate the boundary precisely. The brush strokes divide an image into three areas: definite umbra area, definite lit area, and boundary, which consists of penumbra area as well as parts of the umbra and lit area. Our algorithm precisely locates the penumbra area from the user specified boundary, and removes the shadow seamlessly. A working example of our algorithm is illustrated in Fig. 1.

This paper aims to remove shadow effects such that the resulting shadow-free image has consistent texture between the shadow and lit area. We first construct a new image gradient field that removes the gradients induced by the shadow effect and has consistent gradient characteristics between the shadow and lit area. Then we can reconstruct the shadow-free image from the new gradient field through 2D integration by solving a Poisson equation similar to previous work (c.f. [2,6,13]). The major challenge is to construct the new image gradient field G^n given only the rough shadow boundary from users. In § 2.1, we describe a novel algorithm to estimate the illumination change curves across the shadow boundary and cancel the effect of illumination change on the gradient field in the penumbra area. In the § 2.2, we describe a method to estimate the shadow effect on the texture characteristics in the shadow area and transform the characteristics of gradients there to be compatible with that in the lit area.

2.1   Estimate Illumination Change in Penumbra Area

Properly handling the shadow boundary or the penumbra area is a challenge for shadow removal. The ambiguity of the shadow boundary often makes automatic shadow boundary detection methods fail. Relying on users to provide the precise shadow boundary casts a heavy burden on them. To relieve users' burden, Mohan *et al.* [10] presented a piece-wise model where users only need to specify connected line segments to delineate the boundary. However, when dealing with complex shadow boundaries like the eagle's right wing in Fig. 1(c), their method will still require users to specify a large number of key points. To further reduce users' burden, we only require a rough specification of the shadow boundary from users using brush tools as illustrated in Fig. 1(c).

Given an inaccurate shadow boundary specification, our method simultaneously locates the shadow boundary precisely and estimates the illumination change $C(x, y)$ in Equation 2 in the penumbra area. The complex shape of the shadow boundary makes devising a parametric model of $C(x, y)$ difficult. However, we observe that any line segment crossing the boundary has an easily parameterizable illumination profile. Therefore, we model $C(x, y)$ by sampling line segments across the boundary and estimating a parametric model for each as illustrated in Fig. 3(a). Since the user provided-boundary usually is not accurate enough, unlike [3], we do not sample $C(x, y)$ using line segments perpendicular to the boundary. Instead, like [10], we use a vertical/horizontal sampling line per
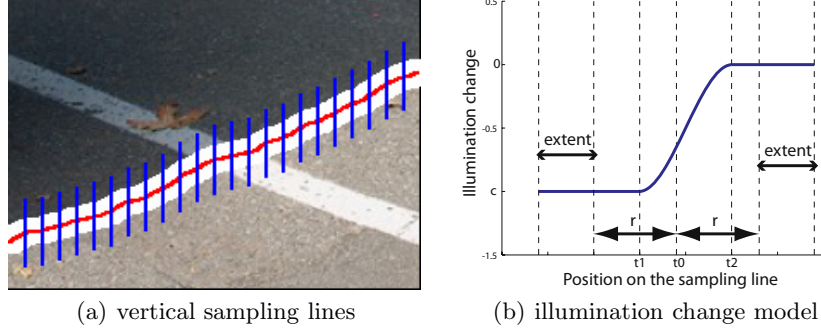
(a) vertical sampling lines   (b) illumination change model

Fig. 3. Sampling illumination change surface using line segments. (a): vertical sampling lines. (b): t_0 and r are the brush center and brush radius. $[t_1, t_2]$ is the penumbra area. *extent* is the range in the umbra and lit area, used to estimate the gradient characteristics.

pixel along the boundary and use the estimated illumination change to cancel the shadow effect on the gradient in Y/X direction. We estimate horizontal and vertical illumination change sampling lines independently.

We model the illumination change along each line segment as the following C^1 continuous piece-wise polynomial as illustrated in Fig. 3(b):

$$C_l(t) = \begin{cases} c, & t < t_1; \\ f(t), & t_1 \leq t \leq t_2; \\ 0, & \text{else.} \end{cases} \tag{3}$$

This piece-wise polynomial model can be parameterized by 3 parameters, denoted as $M_l(c, t_1, t_2)$. Here t_1 and t_2 define the penumbra area along the sampling line. (Without losing generality, we assume $t < t_1$ lies in the umbra area and $t > t_2$ lies in the lit area.) $c(\leq 0)$ is the reduction of the illumination in the umbra area. $f(t)$ is a cubic curve determined by the two boundary points, (t_1, c) and $(t_2, 0)$, and the derivatives at these two points, $f'(t_1) = 0$ and $f'(t_2) = 0$. This illumination change model is determined by both the location of the penumbra area and the characteristics how the illumination changes from c in the umbra area to 0 in the lit area. Due to these combined properties, our method estimates the penumbra area location and the illumination change simultaneously by estimating the above piece-wise polynomial model.

Because we assume that the illumination change surface is smooth, neighboring illumination change models along the shadow boundary should be similar to each other. So we solve for all these models simultaneously instead of fitting each model separately. We formulate the problem of finding illumination change models as an optimization problem, aiming to balance the fitness of the models to the shadow image and the smoothness between neighboring models.

$$E = \sum_{li} E_{fit}(M_{li}, \tilde{I}) + \lambda \sum_{li} \sum_{lj \in N(li)} E_{sm}(M_{li}, M_{lj}) \tag{4}$$

where $E_{fit}(M_{li}, \tilde{I})$ measures the fitness error of the illumination change model M_{li} to the original shadow image $\tilde{I}$, $E_{sm}(M_{li}, M_{lj})$ measures the similarity between M_{li} and M_{lj}, and $N(li)$ denotes the neighborhood of sampling line li. λ is a parameter, with a default value 10.

We measure $E_{fit}(M_{li}, \tilde{I})$, the fitness error of the model M_{li} to the shadow image $\tilde{I}$, as how well the gradient in the penumbra area fits into its neighborhood along the sampling line after shadow effect compensation according to M_{li}.

$$E_{fit}(M_{li}, \tilde{I}) = -\Pi_{t \in [t_{i0}-r_i, t_{i0}+r_i]} \varphi(\hat{G}_{li}(t), T_{li}^{tex}) \tag{5}$$

$$\hat{G}_{li}(t) = \tilde{G}_{li}(t) - C'_{li}(t) \tag{6}$$

where C_{li} is the illumination change curve of M_{li} as defined in Equation 3, C'_{li} is its first derivative, $\tilde{G}_{li}$ is the gradient along li, and $\hat{G}_{li}(t)$ is the gradient after canceling the shadow effect. T_{li}^{tex} is the texture distribution along li. $\varphi(,)$ measures the fitness of the gradient to the distribution T_{li}^{tex}. We model the texture distribution along li as a normal distribution $N(\mu_i, \sigma_i^2)$ of the gradients, which can be estimated explicitly from the umbra and lit extension along li as illustrated in Fig. 3(b). Accordingly, we define the fitness measure as follows:

$$\varphi(G_{li}(t), T_{li}^{tex}) = \frac{exp(-(G_{li}(t) - \mu_i)^2 / 2\sigma_i^2)}{\sqrt{2\pi\sigma_i^2}} \tag{7}$$

We define $E_{sm}(M_{li}, M_{lj})$, the smoothness cost between neighboring illumination change models as follows:

$$E_{sm}(M_{li}, M_{lj}) = \gamma(c_i - c_j)^2 + (1 - \gamma)((t_{1i} - t_{1j})^2 + (t_{2i} - t_{2j})^2)$$

where the first term measures the difference between the illumination steps from the umbra to lit area, and the second term measures the difference between the location of the penumbra area along sampling lines. We emphasize the fact that the illumination change inside the umbra area is mostly uniform by weighting the first term significantly. The default value for γ is 0.9.

Directly solving the minimization problem in Equation 4 is time-consuming. We approximate the optimal solution in two steps:

1. For each sampling line li, we find an optimal illumination change model M_{li}^o which fits the shadow image most by minimizing the fitness error defined in Equation 5. Since the extent of the penumbra area is small, we use a brute-force search method.
2. With the optimal illumination change model M_{li}^o of each sampling line, we approximate the fitness error term in Equation 4 using the difference between the illumination change model M_{li} and M_{li}^o as follows:

$$E = \sum_{li} E_{sm}(M_{li}, M_{li}^o) + \lambda \sum_{li} \sum_{lj \in N(li)} E_{sm}(M_{li}, M_{lj})$$

The above energy minimization is a quadratic minimization problem. We solve it using a Preconditioned Conjugate Gradient method [14].
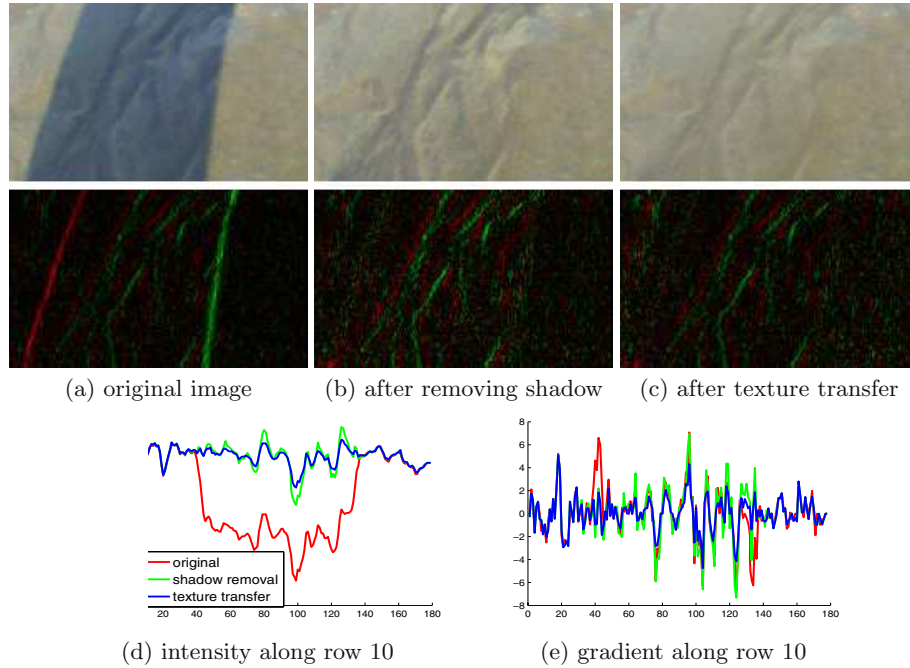
(a) original image   (b) after removing shadow   (c) after texture transfer

(d) intensity along row 10   (e) gradient along row 10

Fig. 4. Reconstruct the gradient field for shadow removal. (a) shows the original image and its gradient field along X direction. For the sake of illustration, we encode the negative and positive gradient values using the GREEN and RED channels respectively. From the original gradient field, we can see the shadow effect on the gradient field by noticing the strong edges along the shadow boundary. By estimating the illumination change across the penumbra area, the shadow effect on the gradient field is canceled as illustrated in (b) and (d). However, as we can see in (b) and (e) right, the shadow area is more contrasty than the lit area, causing inconsistent texture characteristics. This inconsistency is removed after gradient transformation as shown in (c) and (e).

After obtaining the illumination change model along each sampling line, we apply it to the gradient field to cancel the shadow effect according to Equation 6. An example of canceling the shadow effect on the gradients in the penumbra area is shown in Fig. 4(a) and (b).

2.2 Estimate Shadow Effect on Texture Characteristics

Canceling the shadow effect on the gradients in the penumbra area can effectively match the illumination in the shadow area (including penumbra and umbra area) to that in the lit area. However, as illustrated in Fig. 4(b) and (c), it cannot guarantee the texture consistency between the shadow and lit area since the shadow can also affect the texture characteristics in the whole shadow area (§ 1). Our method estimates the shadow effect on the gradient characteristics and transfers the shadow-effect free gradient characteristics to the shadow area to make it compatible with the lit area.

Like transferring color between images [15], where the global color characteristics of an image is parameterized using its sampling mean and deviation, we model the texture characteristics using the sampling mean and deviation of the gradient field. So if given the target mean and deviation, we transform the gradient field in the shadow area as follows:

$$G^s(x,y) = \hat{\mu}^t + \frac{(\hat{G}^s(x,y) - \hat{\mu}^s) * \hat{\sigma}^t}{\hat{\sigma}^s} \tag{8}$$

where $\hat{G}^s$ and G^s are the gradients in the shadow area before and after transformation respectively, and $\hat{\mu}^s$ and $\hat{\sigma}^s$ are the mean and deviation of $\hat{G}^s$. $\hat{\mu}^t$ and $\hat{\sigma}^t$ are the target mean and deviation.

Like transferring color [15], using the characteristics parameters of the lit area as the target parameters can achieve consistent texture characteristics between the shadow and lit area. However, this scheme works well only if the texture distribution is globally homogeneous in the image. Otherwise it can destroy local textures in the shadow area. We calculate the target characteristics parameters by estimating the shadow effect on the gradient distribution and canceling this effect from the original gradient field. Assuming the gradient distribution around the shadow boundary is homogenous and the shadow effect is independent of the shadow-free image, we estimate the shadow effect parameters from gradients around the boundary as follows:

$$\begin{cases} \mu_{se} = \mu_b^s - \mu_b^l \\ \sigma_{se}^2 = \sigma_b^{s\,2} - \sigma_b^{l\,2} \end{cases} \tag{9}$$

where μ_{se} and σ_{se} are the mean and deviation of the shadow effect on gradients in the shadow area. μ_b^s and σ_b^s are the mean and deviation of the gradients in the umbra side along the shadow boundary(the *extent* parts as illustrated in Fig. 3(b)) , and μ_b^l and σ_b^l are those in the lit area side. Accordingly, the target mean and deviation can be calculated by canceling the shadow effect as follows:

$$\begin{cases} \hat{\mu}^t = \hat{\mu}^s - \mu_{se} \\ \hat{\sigma}^t = \sqrt{\hat{\sigma}^{s\,2} - \sigma_{se}^2} \end{cases} \tag{10}$$

Fig. 4(b) and (c) shows that the gradient field transformation leads to consistent texture characteristics between the shadow and lit area. Please refer to the whole image in Fig. 6(a) to examine the consistency of the texture.

3   Results

We have experimented with our method on photos with shadows from *Flickr*. These photos have different texture characteristics. We report some representative ones together with the results in Fig. 1, Fig. 2, Fig. 6, Fig. 7 and Fig. 8, as well as comparison to many representative works [2,6,4,3,10]. (**Please refer to**

(a) original   (b) zeroing-gradient

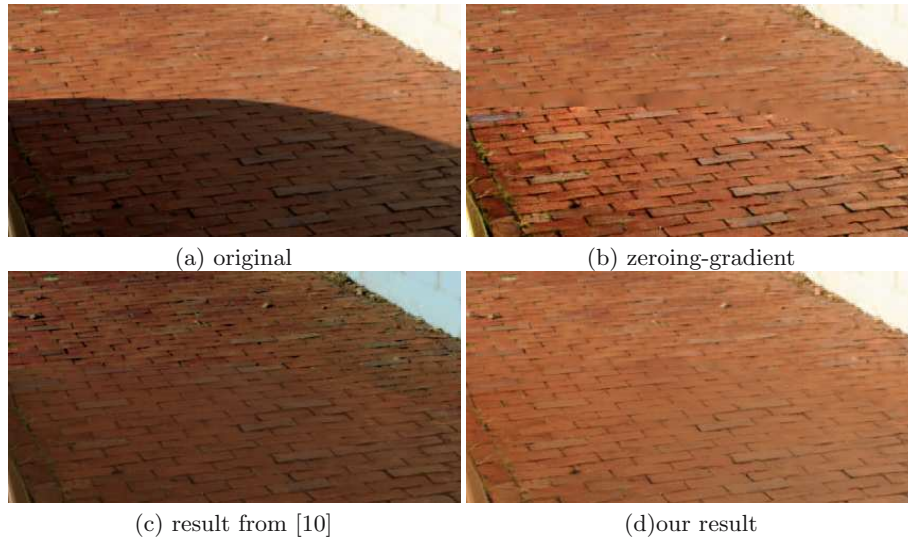(c) result from [10]   (d)our result

Fig. 5. Images in (a) and (c) are from [10]. (b) shadow removed by nullifying the gradients in the boundary [2,6]. (c) shadow removed using the method from [10]. There, not only the illuminance level in the lit area is changed, but also the shadow area is not as contrasty as the lit area. Our method creates a texture-consistent result.

the electronic version of this paper to examine the results. Zooming in on the images will be helpful for the examination.)

For all the experiments, users specify the shadow boundaries with a brush tool. Users do not need to delineate the boundary precisely as shown in Fig. 1(c) (notice the eagle's right wing). They can pick a brush with much larger size than the real shadow boundary area to cover the boundary as shown in the second column of Fig. 6. Given the user specified shadow boundary, our system can automatically perform shadow removal efficiently. The majority of the time is spent on solving the Poisson equation, whose complexity is dependent on the number of pixels in the shadow region. It takes about 3 seconds to remove a shadow region with about 60,000 colored pixels on a 2.2GHz Athlon machine.

Fig. 2 and Fig. 5 compare our method to other representative methods. Methods [2,6] cancel the shadow effect by zeroing the gradients in the boundary area. In this way, the textures there are nullified as shown in Fig. 2(c). While in-painting [4] can partially solve this problem, it sometimes destroys the continuity of the texture as shown in Fig. 2(e). The recent method from [3] can effectively remove shadow, however the texture in the original shadow area is not consistent with that in the lit area as shown in Fig. 2(d). Our method can not only remove the shadows, but also keep the texture consistency between the shadow and lit area as shown in Fig. 2(f). Fig. 5 compares our method to the recent work from [10]. While the illuminance between the lit and the original shadow area is balanced in the result from [10], the illuminance level in the lit area is changed. More overall, the lit and the original shadow area have different

(a) sandy beach

(b) pavement

(c) rock cliff

Fig. 6. experiments results. Left: original images; Middle: shadow boundaries; Right: our results.

(a) tree in hill

(b) sandy beach

(c) desert sand dawn break
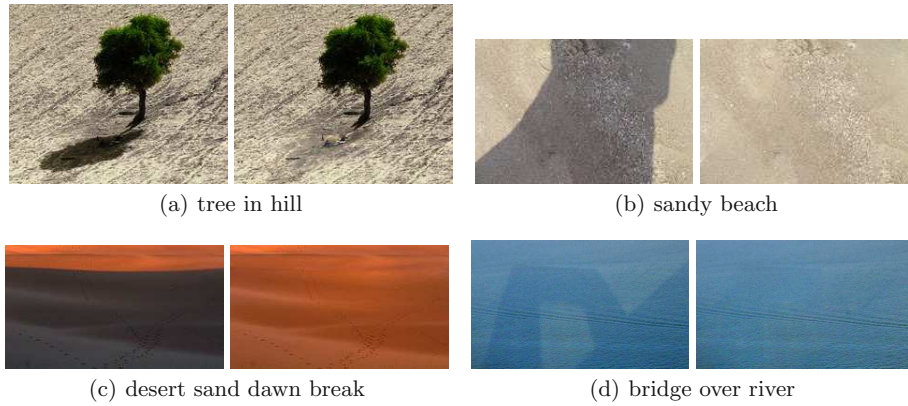

(d) bridge over river

Fig. 7. experiments results. Left: original images; Right: our results.

contrasty levels as shown in Fig. 5(c). Our method effectively removes the shadow as well as keeps the consistent texture characteristics across the whole image as shown in Fig. 5(d) and other examples. For instance, in the Fig. 7(b), the texture of small shell grains in the shadow area and in the lit area is consistent. For the desert example in Fig. 7(c), the highlights across the original shadow boundary are consistent between the shadow and lit area. For the river surface example in Fig. 7(d), the ripples in the shadow area are consistent with that in the lit area. Particularly, the wavefront in the middle is continuous across the original shadow boundaries. For the tree example in Fig. 7(a), the soil inside the shadow region is consistent with the lit area surrounding it. The hill example in Fig. 8(a) is similar.
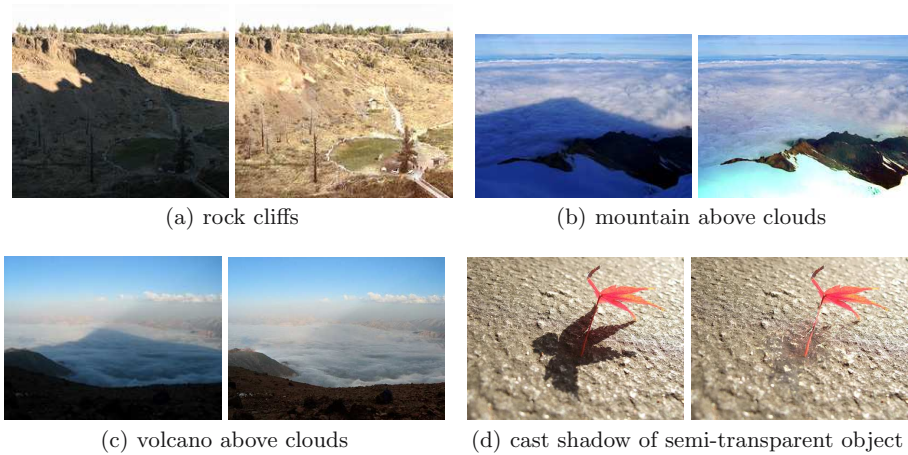

(a) rock cliffs

(b) mountain above clouds

(c) volcano above clouds

(d) cast shadow of semi-transparent object

Fig. 8. experiments results. Left: original images; Right: our results.

From the results in Fig. 6, 7 and 8, we can see that the proposed algorithm can seamlessly remove shadows in images with various texture characteristics. For example, the shadows are on the beach (Fig. 6(a)), on the road surfaces (Fig. 6(b)), on the sands (Fig. 7(b)), on the desert (Fig. 7(c)), on the river surface (Fig. 7(d)), on the hills (Fig. 7(a) and Fig. 8(a)), etc. Our method works well on specular surfaces such as Fig. 6(a), as well as Lambertian surfaces, such as examples in Fig. 7.

Examples in Fig. 8(b) and (c) are very interesting. Noticing the mountains in these examples, shadow removal reveals the beautiful texture details in the original dark shadow areas, which are concealed in the original shadow images. What is particularly interesting is that shadow removal recovers the *blue glacier ice* phenomenon[1] in the Fig. 8(b) (Notice the blue-cyan area of the snow in the left bottom.).

We found from the experiments that our method does not work well on some images. Taking Fig. 8(d) as an example, the shadow area in the original image looks more reddish than its surrounding lit area. This is because when the lighting is blocked by the semi-transparent red leaf, its red component can still pass through. For this kind of cast shadow, the general shadow model in Equation 2 used in previous work (including ours) does not hold. Noticing the original shadow region in the resulting image, we can still sense the reddish component there. In future, analyzing the caustics of shadow from its context may help solve this problem. However, our current method is effective for many images.

4   Conclusion

In this paper, we presented a texture-consistent shadow removal method. Specifically, we construct a shadow-effect free and texture-consistent gradient field between the shadow and lit area and recover the shadow-free image from it by solving a Poisson equation. The experiments on shadow images from *Flickr* demonstrate the effectiveness of the proposed method.

Currently, our method provides users with a brush tool to specify the shadow boundary. The brush tool is very popular in digital photography software. As illustrated in the examples in previous sections, our method does not require a precise shadow boundary. We envision our method a convenient tool for interactive photo editing. Of course, integrating an automatic shadow detection algorithm can make our method even easier to use.

We characterize texture characteristics using the sampling mean and deviation of the gradient field. Based on our current experiments on photos from *Flickr*, this global model works well. An important reason for its success is that a global transformation on an image or its various representations usually preserves important properties of the original image. In fact, similar models work pretty well in other applications like color transfer [15] as well.

---

[1] `http://www.northstar.k12.ak.us/schools/joy/denali/OConnor/colorblue.html`

References

1. Barrow, H., Tenenbaum, J.: Recovering intrinsic scene characteristics from images. In: Computer Vision Systems. Academic Press, London (1978)
2. Weiss, Y.: Deriving intrinsic images from image sequences. In: IEEE ICCV, pp. 68–75 (2001)
3. Arbel, E., Hel-Or, H.: Texture-preserving shadow removal in color images containing curved surfaces. In: IEEE CVPR (2007)
4. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the removal of shadows from images. IEEE Trans. Pattern Anal. Mach. Intell. 28(1), 59–68 (2006)
5. Liu, Z., Huang, K., Tan, T., Wang, L.: Cast shadow removal combining local and global features. In: The 7th International Workshop on Visual Surveillance (2007)
6. Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing shadows from images. In: 7th European Conference on Computer Vision, pp. 823–836 (2002)
7. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Comput. Vis. Image Underst. 95(2), 238–259 (2004)
8. Levine, M.D., Bhattacharyya, J.: Removing shadows. Pattern Recognition Letters 26(3), 251–265 (2005)
9. Wu, T.P., Tang, C.K., Brown, M.S., Shum, H.Y.: Natural shadow matting. ACM Trans. Graph. 26(2), 8 (2007)
10. Mohan, A., Tumblin, J., Choudhury, P.: Editing soft shadows in a digital photograph. IEEE Comput. Graph. Appl. 27(2), 23–31 (2007)
11. Baba, M., Mukunoki, M., Asada, N.: Shadow removal from a real image based on shadow density. ACM SIGGRAPH 2004 Posters, 60 (2004)
12. Fredembach, C., Finlayson, G.D.: Hamiltonian path based shadow removal. In: BMVC, pp. 970–980 (2005)
13. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans. Graph. 22(3), 313–318 (2003)
14. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. SIAM, Philadelphia (1994)
15. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. 21(5), 34–41 (2001)