

Shadow Removal Method for Single Image Based on Instant Learning

Tianjun Zhu,^{1,2} Zhiliang Zou,^{1*} Tunlung Wu,² Jianying Li,² Bin Li,³
Rui Song,⁴ Qianyu Song,⁴ Sanmiao Du,¹ and Hegao Wan¹

¹College of Mechanical and Equipment Engineering, Hebei University of Engineering,
19 Taiji Road, Handan 056021, China

²Department of Mechanical and Automotive Engineering, Zhaoqing University,
Yingbin Avenue, Zhaoqing 526061, China

³Aptiv PLC, 5275 Delphi Drive, Troy, Michigan 48098-2815, USA

⁴CATARC Automotive Test Center Guangzhou Co., Ltd., 39 Xiangshan Avenue, Guangzhou 511340, China

(Received March 24, 2021; accepted June 11, 2021)

Keywords: shadow removal, instant learning, offset correction, illumination estimation, enhancement

In this paper, a simple user-aided, robust, and high-quality fast shadow removal method is proposed. Firstly, an instant learning method is used to remove the shadow of an image, guided by input from the user of two rough areas of shadow and lighting pixels from which the initial shadow mask for shadow removal is derived. Secondly, the detected shadow image is offset and corrected to eliminate the influence of ambient light by a simple input of a color sampling line indicating the rate of change of illumination of the shadow region, penumbra region, and luminance region pixels. Clustering and illumination estimation are then performed for the corrected images. Meanwhile, the shadow is removed by calculating the statistical distribution of the shadow-free regions, and the lighting is restored by estimating the illumination scaling factor. Finally, the energy function used for illumination optimization can perform smooth estimation and image enhancement of the area after shadow removal. Experiments show that our proposed algorithm can generate high-quality shadow-free images and deal with difficult scenes with various texture types and uneven shadows.

1. Introduction

Shadows are a common natural phenomenon and can be used to predict the shape and location of the object that produces the shadow, as well as the source and direction of light. However, shadows can cause many problems in the field of computer vision. For example, in intelligent vehicle systems, they greatly affect image segmentation and the accurate identification of vehicles, pedestrians, and road markings by vision sensors (cameras) mounted on a car. Shadow removal from images is an ill-posed mathematical problem, and it is difficult to remove shadows completely from images. As a hotspot in the field of computer vision, shadow removal is mainly divided into two sub-tasks: shadow detection and shadow removal. One of the

*Corresponding author: e-mail: 991954002@qq.com
<https://doi.org/10.18494/SAM.2021.3397>

difficulties of light restoration in the textured area of a shadow surface is the variability of diverse shadow types.⁽¹⁾ In particular, the following four common attributes of shadows greatly increase the difficulty of shadow removal: rich texture features, high shadow softness, changeable shadow colors, and extreme broken shadows.

In response to the above-mentioned problems and challenges in shadow removal, methods to obtain high-quality shadow-free images have been proposed. However, it is always difficult to remove shadows completely, so most algorithms have been designed for one or more specific problems to improve the accuracy of shadow-free images. Similarly, by observing images with shadows, it has been found that the pixels of objects with the same material will form a color line in the RGB color space with a change of illumination.⁽²⁾ By using this characteristic of the shadow image, a simple scribble color sampling line is designed to indicate the relationship between the illumination gradient of pixels in the shadow region and those in the lighting region for image offset correction, so as to eliminate possible artifacts in the penumbra region. Then, the shadow is removed and the lighting is restored using the statistics ratio of the shadow area. In addition, we use an energy function for illumination optimization to perform smooth estimation and enhancement of the image region after removing shadows. Through standard evaluation data, it is demonstrated quantitatively and qualitatively that this method is better than some state-of-the-art methods. In this paper, the proposed shadow removal algorithm is mainly applied to the processing of a single image.

2. Related Work

Shadow removal methods can be divided into automatic^(3–8) and user-aided^(1,9–15) methods. The difference between them is related to the initial detection of shadows, and the removal of image shadows after detection is an automatic task. In any case, shadow detection and removal are ill-posed problems and are difficult to implement reliably.

Early methods of automatic shadow removal included the use of internal images, a popular solution to this problem.⁽⁶⁾ The decomposition of the internal image provides shadow and reflectance information, but such methods are unreliable and may result in excessive processing. Other methods detect shadows by learning the features of shadow regions.^(7,8) However, shadow detection is often limited by the scope of training data, the edge detection, and the segmentation quality of image shadows. In addition, in recent years, shadow removal has mainly relied on the deep learning method of a convolutional neural network with end-to-end training. Learning mappings from paired shadows and the ground truth image is a new approach in image shadow removal.^(3–5) Qu *et al.* developed three sub-networks to extract features from multiple views and embedded these sub-networks in the algorithm framework to remove shadows.⁽³⁾ In Ref. 4, one conditional generation antagonistic network (CGAN) was used to detect shadows and another CGAN was used to remove shadows. However, such methods require a large number of high-quality, paired data sets to be trained. Thus, a shortage of high-quality data sets is a major limitation in the development of in-depth learning methods. Later, Hu *et al.* examined the use of unpaired shadow image data sets to perform mask-shadow generation antagonistic network

(Mask-Shadow GAN) in-depth learning, which can achieve good shadow removal of visual effects.⁽⁵⁾ Nevertheless, such antagonistic training still requires a large number of scene data sets for learning and training. Moreover, this algorithm is complex and time-consuming.

User-aided methods usually require different degrees of manual assistance and can obtain more precise shadow removal at the cost of requiring the user to detect some areas of shadow. Wu *et al.* asked users to confirm shadows and lighting pixels, excluding other uncertain areas.⁽⁹⁾ Other methods required users to define the boundary of a shadow.^(10,11) Shor and Lischinski only required a shadow pixel as the input, but there were limitations on other areas of the shadow surface, which were not detected in the image.⁽¹²⁾ Arbel and Hel-Or required users to define many texture anchors.⁽¹³⁾ In addition, the use of natural coatings for penumbra restoration will lead to penumbra artifacts in the final result of treatment.⁽¹⁴⁾ Zhang *et al.* performed a simple user-aided method to indicate that there is a similar texture feature relationship between lighting pixels and shadow pixels.⁽¹⁵⁾ However, while the user used the illumination factor to perform image extinction, the image processing results are sensitive to the user's initial setting.

Until now, most shadow removal methods^(7,11,12) have visually compared some specific shadow images, and only a few studies have performed a quantitative evaluation. These methods of image shadow removal lack high-quality ground truth data for comparison. Quantitative tests were carried out in Ref. 10, but the data used to obtain the test results were not disclosed. Guo *et al.* first disclosed the real data set of shadowless images they collected and then tested the data. In addition, this type of image data set is very difficult to collect and will be affected by many uncontrollable changes outdoors.⁽⁸⁾ Because these changes affect shadow removal results, quantitative testing of such data may be difficult. Therefore, Han and Cosker extended the aforementioned data set according to the diversity of scenes (including four different categories: texture, softness, brokenness, and colorfulness), and presented new division rules.⁽¹⁾ This partition rule will be used to evaluate multiple shadow removal algorithms for selected images in this paper.

Traditional image features and models were used for shadow removal in early studies. In recent years, machine learning methods have been used to improve the accuracy of shadow removal. Our method combines both image features and machine learning methods for fast shadow removal. With the previous literature taken into account, our algorithm has three important contributions:

- (1) Powerful offset correction. The effect of using tone and contrast to restore brightness may cause inconsistencies in shadow correction areas. A robust and simple user-indicated color sampling line offset correction is used to correct artifacts.
- (2) Simple and robust image detection based on user inputs. The shadow detection in this paper only requires two simple marks made by the user to provide samples of lighting and shadow pixels. The method in this paper is different from previous methods that require more complex user inputs^(9–11) or simpler inputs,⁽¹²⁾ which often compromised the robustness and quality of shadow removal results.
- (3) Efficient shadow removal. The statistical distribution of the shadow-free area is introduced, and then the shadows are removed and lighting is restored. This method does not require

prior training, and can efficiently estimate lighting changes without complicated image training. Compared with some existing methods such as image models and deep learning, our method has certain advantages.

These contributions are expected to be very significant for the visual recognition field. Moreover, the validity of the proposed shadow removal algorithm was demonstrated in extensive repeated experiments.

3. Simple Interactive Shadow Removal Algorithm

In this section, we first briefly introduce the proposed algorithm, and then expand on the technical details of each part in the following sections (see Fig. 1).

(1) Shadow detection (Sect. 3.2). The shadow image is first detected using the K-nearest neighbors (KNN) classifier, which is guided by two simple user scribbles [Fig. 1(b)]. Next, the detection image is binarized and classified. Then, Gaussian filtering is carried out on the binarized image.

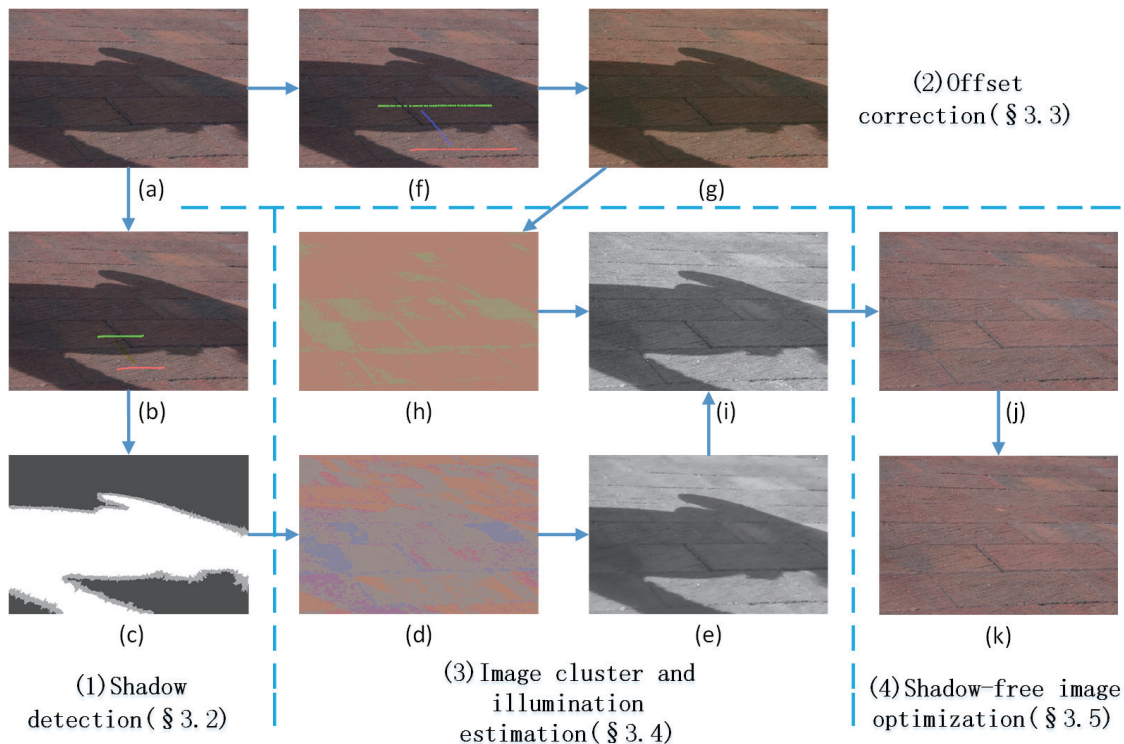


Fig. 1. (Color online) Shadow removal pipeline. (a) Original image; (b) shadow image and user scribbles indicating illuminated (red line) and shadowed (green line) pixels; (c) detected shadow mask; (d) clustering of original image; (e) initial luma estimation image; (f) shadow image and user-defined illumination gradient region (blue line); (g) corrected image; (h) image corrected by clustering; (i) image corrected by luma estimation; (j) initial shadowless image; (k) illumination-optimized shadowless image.

Finally, an initial shadow mask is obtained by binarizing the filtered image [threshold size of 0.5, Fig. 1(c)].

- (2) Offset correction (Sect. 3.3). After obtaining the rough shadow mask, the gradient of the illumination factor is sampled by using the color sampling line perpendicular to the shadow boundary [Fig. 1(f)], and then the image is offset-corrected to eliminate the influence of ambient light [Fig. 1(g)].
- (3) Image clustering and illumination estimation (Sect. 3.4). Uniform sphere sampling is performed on the detected image and the offset-corrected image to construct a k-dimensional tree (KD-Tree) for clustering color sampling lines [Figs. 1(d) and 1(h)]. Then, the shadowless image [Fig. 1(j)] is obtained by removing the umbra and restoring the illumination of the image by estimating the illumination scaling factor [Figs. 1(e) and 1(i)].
- (4) Illumination optimization (Sect. 3.5). In the case without the introduction of artifacts, the shadow ratio is improved by using an energy function for illumination optimization so that most regions in the shadow area correspond to homologous material regions in the luminance area. Finally, the pixel scale factor used for restoring the illumination is used to perform local smoothing on the shadow-free image to achieve a consistent texture and illumination [Fig. 1(k)].

3.1 Shadow image model

According to Ref. 16, an image $I_c(x)$ can be expressed as the dot product of the illumination $L_c(x)$ and reflectance $R_c(x)$:

$$I_c(x) = L_c(x)R_c(x), \quad c \in \{R, G, B\}. \quad (1)$$

$L_c(x)$ can be represented by the direct illumination $L_c^d(x)$ and ambient illumination $L_c^a(x)$, where c is the color channel of the color image. Shadows are cast when the direct illumination $L_c^d(x)$ in the scene is blocked by an object, and the illumination can be rewritten as

$$\begin{cases} L_c(x) = \tilde{S}(x)L_c^d(x) + L_c^a(x), \\ \tilde{S}(x) \in [0, 1]. \end{cases} \quad (2)$$

The attenuation factors of direct illumination $\tilde{S}(x)$ in the above formula can be expressed as

$$\tilde{S}(x) = \begin{cases} 0, & x \in \text{shadow pixels}, \\ 1, & x \in \text{other pixels}. \end{cases} \quad (3)$$

According to Ref. 17, the ambient illumination $L_c^a(x)$ in Eq. (2) can be expressed as an affine function of the offset. Thus, the illumination function $L_c(x)$ can be transformed into the following affine function:

$$L_c(x) = [\tilde{S}(x) + \alpha] L_c^d(x) + [L_c^a(x) - \alpha L_c^d(x)], \quad \alpha \in \mathbb{R}, \quad (4)$$

where α is the offset. The former term $[\tilde{S}(x) + \alpha] L_c^d(x)$ in Eq. (4) represents the illumination change along the direction of direct illumination $L_c^d(x)$, whereas the latter term $[L_c^a(x) - \alpha L_c^d(x)]$ describes a certain offset α . Thus, the offset of the reflectance $O_c^\alpha(x)$ can be expressed as

$$O_c^\alpha(x) = [L_c^a(x) - \alpha L_c^d(x)] R_c(x). \quad (5)$$

Therefore, the shadow image can be simplified as follows using Eqs. (1), (2), and (5):

$$\begin{cases} I_c(x) = (1 + \alpha)S(x)R_c(x) + O_c^\alpha(x), \\ S(x) \in \left[\frac{\alpha}{\alpha + 1}, 1 \right]. \end{cases} \quad (6)$$

where $S(x)$ is the new illumination attenuation factor.

Three important results for the shadow image in the RGB color space can be obtained from Eq. (6). Firstly, if the reflectance $R_c(x)$ is given, Eq. (6) is an affine function, and the image pixel will form a color line based on the change in direct illumination [see Fig. 2(a)]. Secondly, when the attenuation factor $S(x) \in [-\infty, +\infty]$ is infinitely extended, the color sampling line will not

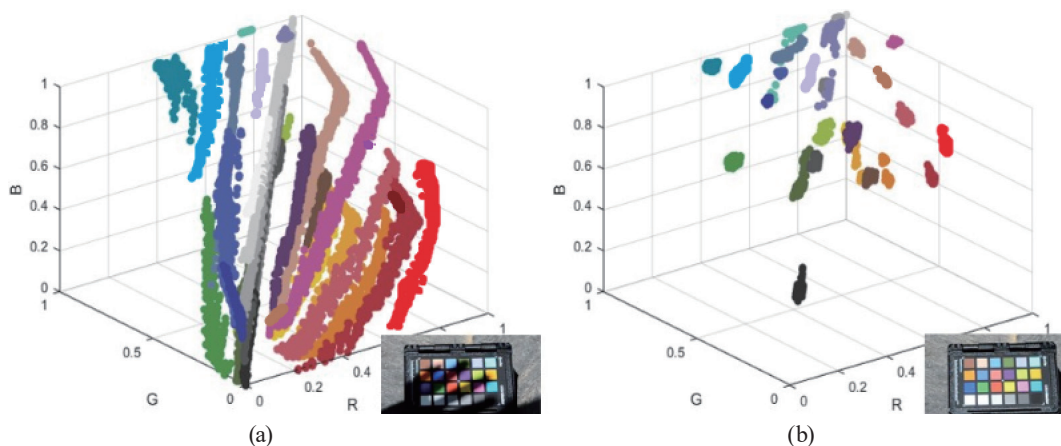


Fig. 2. (Color online) Shadow image (a) and shadowless image (b) and their corresponding RGB space distributions.

intersect the origin unless the direct illumination $L_c^d(x)$ and ambient illumination $L_c^a(x)$ are in the same direction. Finally, because the color sampling lines are related to the reflectivity, when α is a constant value, no deviation can result in all the color sampling lines intersecting with the origin. Similar conclusions are drawn in Refs. 17 and 18.

From the shadow image model above, it can be seen that the color sampling lines provide some important clues about the illumination changes of objects of the same material. In addition, when the offset of all different color sampling lines is eliminated, the shadow ratio can be restored by changing the gradient area of the image to remove the shadow.

3.2 Shadow detection

To remove the shadows, the shadows in the image should first be detected. Since we focus on the removal of shadows, we adopt the method in Ref. 1 to detect shadows. First, we need two rough user inputs to indicate the light and shadow pixels of the image [Fig. 1(b)]. The shadow pixels in the RGB color space in the logarithmic domain are used to train features and construct a KNN classifier ($K = 3$). The distance metric of the classifier is the Euclidean distance, and the classification metric is based on the principle of large numbers for closest-point tie-breaking. We choose the Gaussian kernel for spatial filtering (size 5, standard deviation 3) and apply it to the posterior probability image. Finally, a threshold of 0.5 is used to binarize the filtered image (as shown in Fig. 3) to obtain a rough shadow mask.

After obtaining the rough shadow mask, we use the gradient of the lighting factor $S(x)$ of the sampling line perpendicular to the shadow boundary instead of dealing with time-consuming alignment samples and selected samples.⁽¹⁾ Similarly to in Ref. 19, we can obtain the initial penumbra area by sampling. Finally, as shown in Fig. 3, the scene is divided into the lit area (N), penumbra area (P), and umbra area (U).

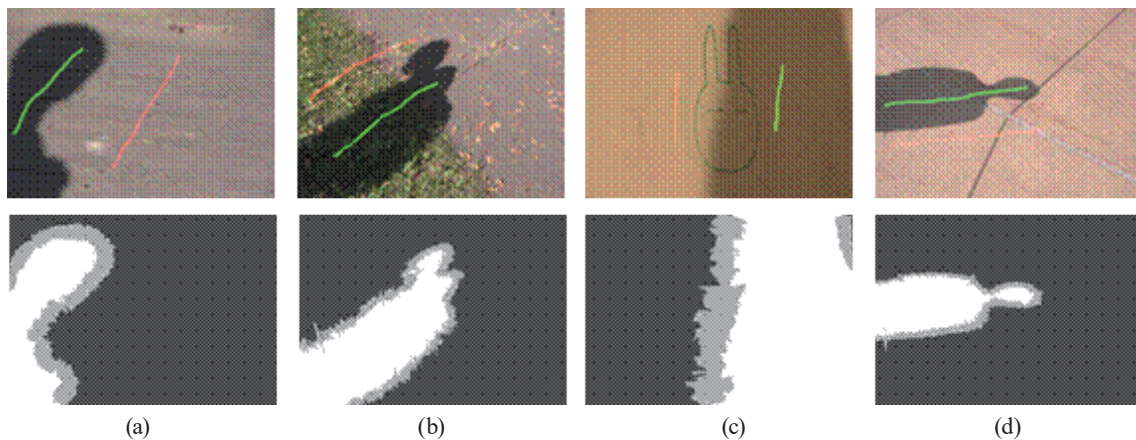


Fig. 3. (Color online) First row: shadow images with user inputs. Second row: shadow masks obtained from image detection.

3.3 Offset correction

In Refs. 20 and 21, fog lines were used for image defogging, and it was found that the pixels in the fog-free image formed color lines in the RGB space. These lines through the origin were derived from the objects within the shadow. When a haze scene appeared in an image, the color sampling line in the haze image no longer passed through the origin, i.e., the offset phenomenon occurred. Inspired by this, in Ref. 21, a line was set for each of these patches for use as a line relative to the origin of the deviation to estimate patches of reflectance. We propose a new algorithm for clearer image processing based on the nonlocal prior. Similarly, we found that the pixels of objects of the same material in the shadow image formed a color sampling line in the RGB spatial distribution with a change in illumination (Fig. 2). Therefore, an instant learning method that uses color sampling lines to indicate changes in shadow pixel illumination is proposed to perform offset correction.

First, the user manually selects an area with similar reflectance R^{g_u} and a significant illumination gradient variation g_u [e.g., Fig. 4(a)]. Then, principal component analysis is used to obtain the direction of the sampling line v of g_u and compute the mean value p_c of the area with the illumination gradient variation. The area of the sampling line $I_c(g_u)$ can be expressed as

$$I_c(g_u) \approx t(g_u)v_c + p_c, \quad (7)$$

where $t(g_u)$ is the independent variable of the sampling line.

As described by Eq. (6), different offsets α will cause different offsets. Therefore, when we eliminate the smallest offset, the change in the image pixel values will be minimal or the error is likely to be less. Therefore, the minimum offset $O_c^{\tilde{\alpha}}(g_u)$ perpendicular to the color sampling line is selected in this paper [e.g., Fig. 5(a)]:

$$O_c^{\tilde{\alpha}}(g_u) = [L_c^a(x) - \tilde{\alpha}L_c^d(x)]R_c^{g_u} \approx p_c - \frac{\langle p|c \rangle}{\|v_c\|}v_c, \quad (8)$$

where $\langle | \rangle$ represents the dot product operation and $\| \|$ is the length operation of the vector.

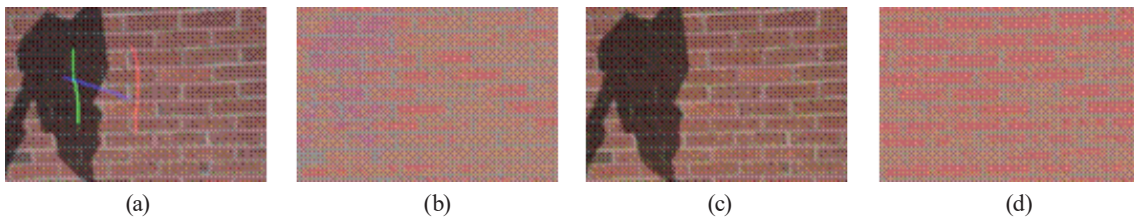


Fig. 4. (Color online) (a) Original image and user-defined illumination gradient area (blue line); (b) clustering of original image; (c) corrected image; (d) clustering of corrected image.

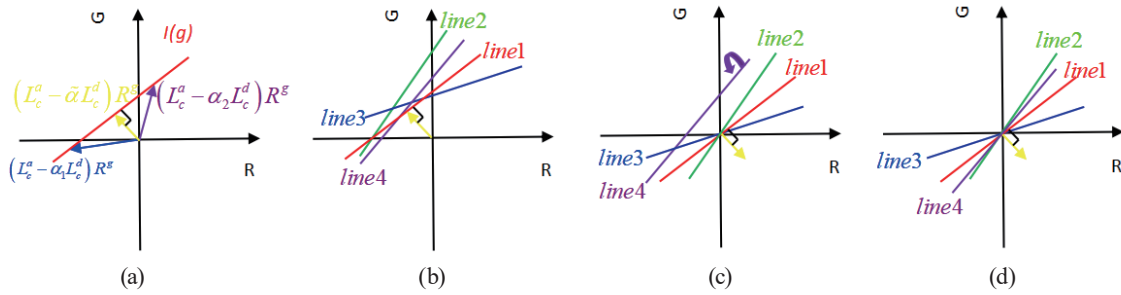


Fig. 5. (Color online) (a) Color sample line g with different offsets; (b) corrected different sample lines in RGB space; (c) corrected sampling lines 1–3; (d) eliminated offset in different sample lines.

Next, a rough offset-corrected image $I'_c(x)$ is obtained by eliminating the minimum offset $O_c^{\tilde{\alpha}}(g_u)$ [Fig. 5(c)]:

$$I'_c(x) = I_c(x) - O_c^{\tilde{\alpha}}(g_u). \tag{9}$$

As mentioned above, through this correction of the sampling lines, the direct illumination is fixed globally. Thus, the direction between different pixels is determined by the pixel reflectance. Therefore, the minimum offset $O_c^{\tilde{\alpha}}(x)$ can be estimated as

$$O_c^{\tilde{\alpha}}(x) = O_c^{\tilde{\alpha}}(g_u) \frac{R_c(x)}{R_c^{g_u}} \approx O_c^{\tilde{\alpha}}(g_u) \frac{I'_c(x)}{\|I'(x)\|} \Big/ \frac{v}{\|v_c\|}. \tag{10}$$

The final offset-corrected image [Figs. 4(c) and 5(d)] can be expressed as

$$\tilde{I}_c(x) = I_c(x) - O_c^{\tilde{\alpha}}(x) = (1 + \tilde{\alpha})S(x)L_c^d(x)R_c(x). \tag{11}$$

3.4 Image clustering and illumination estimation

In this section, all sample lines intersect at the origin after correction. The corrected image $\tilde{I}_c(x)$ can be represented in spherical coordinates as

$$\tilde{I}_c(x) = [d(x), \theta(x), \varphi(x)], \tag{12}$$

where $d(x)$ is the distance $\|\tilde{I}_c(x)\|$ from the origin, and $\theta(x)$ and $\varphi(x)$ are the longitude and latitude in spherical coordinates, respectively.

For areas of the image surface with similar reflectance, a change in the attenuation factor $S(x)$ of illumination only affects the distance $d(x)$ and does not change the longitude and latitude in

spherical coordinates. This implies that our algorithm can obtain different sampling lines by clustering similar spherical coordinates. Next, uniform sphere sampling is used to construct a KD-Tree [Fig. 4(d)] to cluster sampling lines. Different clusters represent different surface materials.

For each given sampling line $l(g)$, $d(x)$ can be calculated using the pixel intensity as follows:

$$d(x) = (1 + \tilde{\alpha})S(x) \left\| L_c^d(x) \otimes R^g(x) \right\|, \quad (13)$$

where \otimes represents per-pixel multiplication.

In addition, when it is assumed that there are no shadow pixels at the most distant sampling lines, the value of the illuminance attenuation factor is 1. Then, the maximum distance $d_{max}(x)$ of the sampling lines is

$$d_{max}(x) = (1 + \tilde{\alpha}) \left\| L_c^d(x) \otimes R^g(x) \right\|. \quad (14)$$

By combining Eqs. (13) and (14), the illumination attenuation factor $S(x)$ can be obtained as follows:

$$S(x) = d(x) / d_{max}(x). \quad (15)$$

From this, we can estimate the change in illumination of the image after correction (see Fig. 6).

3.5 Shadow removal and illumination optimization

According to the offset correction and illumination estimation of the shadow image, our algorithm performs a shadow removal method to ensure global uniform illumination, which has better robustness in complex scenes. The formula used for shadow removal is as follows:

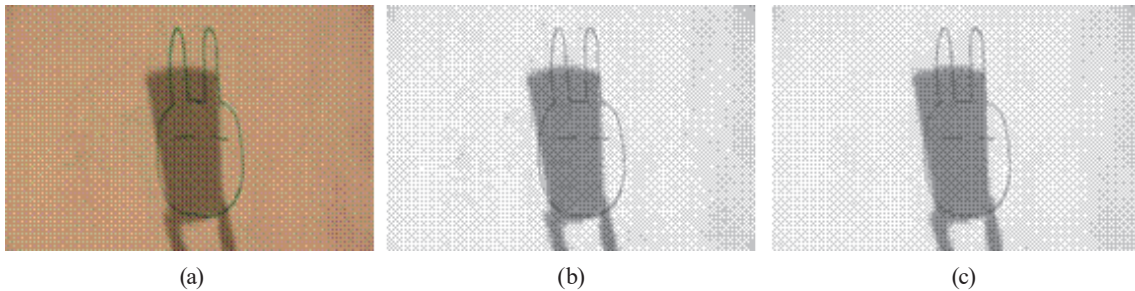


Fig. 6. (Color online) (a) Shadow image; (b) initial luma estimation image; (c) image corrected by the luma estimation.

$$I_c^{\tilde{f}}(x) = \tilde{I}_c(x)/S(x) + O_c^{\tilde{\alpha}}(x). \quad (16)$$

We remove the shadows according to the pixel illuminance ratio of each region of the more refined image. First, the umbra is removed by calculating the distribution of the umbra in the bright region:

$$I_c^{\tilde{f}}(x) = \left\{ \tilde{I}_c(U^g) - E[\tilde{I}_c(U^g)] \frac{SD[\tilde{I}_c(N^g)]}{SD[\tilde{I}_c(U^g)]} + E[\tilde{I}_c(N^g)] \right\}, \quad (17)$$

where g is the color sampling line index, $E[\cdot]$ is the average operator, $SD[\cdot]$ is the standard deviation operator, U represents the umbra area, and N represents the lighting area.

Equation (17) only calculates the illumination distribution of the umbra region and lighting region, ignoring the existence of the penumbra. Moreover, with the gradual change of lighting intensity, the penumbra mask cannot be restored in the same way as in the removal of the umbra area. Inspired by Refs. 12 and 22, we calculate the pixel value and illumination attenuation factor $S(x)$ of the shadow-free region $I_c^f(U^g)$, and then estimate the proportion of shadow pixels to restore the illumination of the penumbra. First, we calculate the mean umbra ratio k_c^U :

$$k_c^U = E[I_c(U^g)/\tilde{I}_c^f(U^g)]. \quad (18)$$

Then, the penumbra ratio is estimated by amplifying the umbra ratio through the illumination attenuation factor:

$$k_c^P = k_c^U + (1 - k_c^U) \frac{S(P^g) - E[S(U^g)]}{E[S(N^g)] - E[S(U^g)]}. \quad (19)$$

Finally, using Eq. (19), we use the scale in the penumbra k_c^P to slowly transform the umbra ratio k_c^U to 1 when there is no shadow ratio. As a result, we obtain the following image with the penumbra removed:

$$\tilde{I}_c^f(P^g) = \tilde{I}_c(P^g)/k_c(P^g). \quad (20)$$

After removing the shadow regions from the original shadow image, we run the above formula in MATLAB [Fig. 7(b)]. The figure clearly shows that artifacts appear in the shadow of the image, and the illumination of the penumbra area is not well recovered. The obtained shadow-free image is illuminated for optimization. Since most of the pixels in the shadow region of the image correspond to the pixels of similar materials in the bright region, most of the

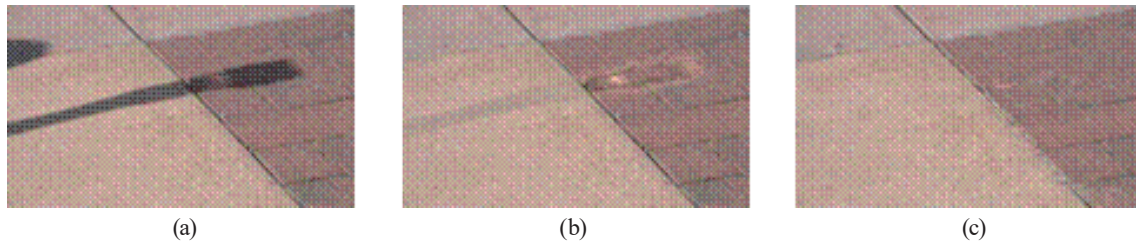


Fig. 7. (Color online) (a) Original shadow image; (b) initial shadowless image with obvious artifact; (c) optimized shadowless image.

shadow area can be removed using the above algorithm. At the same time, local smoothing is carried out for the area with changed illumination, so as to estimate the illumination in the other areas from the re-illuminated regions. Therefore, we refine the shadow ratio $k_c(x)$ by minimizing the following energy function:⁽²³⁾

$$\arg \min_{k'_c(x)} \sum_x \left[\frac{k_c(x) - k'_c(x)}{\sigma^S(x)} \right]^2 + \lambda \sum_x \sum_{y \in N_x} \left[\frac{k'_c(x) - k'_c(y)}{\|I(x) - I(y)\|} \right]^2, \quad (21)$$

where $\sigma^S(x)$ is the standard deviation of $S(x)$, which is computed from each sampling line, λ is used to balance the hand smoothing factor [the first term in Eq. (21)] and the smoothness term (second term) in the control data, and N_x is the size of each pixel x among its four nearest neighbors. After many experiments, we set $\lambda = 0.001$.

The final shadow-free image $I_c^f(x)$ can be represented as follows [see, e.g., Fig. 7(c)]:

$$I_c^f(x) = \tilde{I}_c(x) / k'_c(x) + O_c^{\tilde{\alpha}}(x). \quad (22)$$

We summarize the main framework of the shadow removal method described in the previous section in Algorithm 1. The user inputs a portion of the region that indicates the lighting characteristics of the scene and a region (sampling line) with significant lighting changes. We use simple user scribbles to obtain high-quality shadowless images.

Algorithm 1
Shadow removal.

Input: original image;
Output: shadowless image;
Shadow detection: use user scribbles and KNN classifier to detect shadow image;
Perform offset correction;
Calculate image clustering and illumination estimation;
Shadow removal:
for each color sample line **do**
Umbra removal: calculate the distribution of shadow pixels and nonshadow pixels to remove the umbra;
Penumbra removal: combine the umbra scale factors to remove the penumbra band;
end
Optimize the shadow through the energy function to obtain the final shadowless image.

4. Experiment and Evaluation

4.1 Quantitative assessment

In accordance with previous work,^(1,12) we calculate the root mean square error (RMSE) per pixel between the image processed by our algorithm and the real shadow-free image. Then the RMSE between the real situation of the unshaded ground truth image and the original shadow image is computed. Finally, the two RMSE values are divided to quantitatively compare the effects of the algorithm proposed in this paper. The normalized calculation can clearly show the robustness of the method in this paper.

Table 1 gives quantitative results. We evaluate our method by comparing shadow removal from individual images in the public data sets provided by Refs. 1 and 24. The best score is given in bold. As shown in Table 1 and Fig. 8, our approach is superior to previous methods in most cases,^(1,8,15) especially for testing the complex texture and softness of shadows.

4.2 Visual comparison

A visual comparison of a typical example of using the proposed shadow removal algorithm is shown in Fig. 8. In general, our approach yields higher quality and more aesthetically pleasing shadow removal results than methods that specifically target shadow boundary restoration. However, small artifacts sometimes occur when the shadow region is highly similar to the background. Figure 9 shows some of the difficult special cases, where there is a broken shadow or shadows similar to the background. Further work is required to solve these special cases.

Table 1
Shadow removal errors for test cases according to four attributes.

	Degree	All pixel errors				Shadow pixel errors			
		Guo <i>et al.</i> ⁽⁸⁾	Zhang <i>et al.</i> ⁽¹⁵⁾	Han and Cosker ⁽¹⁾	Ours	Guo <i>et al.</i> ⁽⁸⁾	Zhang <i>et al.</i> ⁽¹⁵⁾	Han and Cosker ⁽¹⁾	Ours
Texture	Weak	0.53(0.50)	0.35(0.17)	0.27(0.18)	0.24(0.17)	0.42(0.57)	0.15(0.19)	0.11(0.09)	0.09(0.10)
	Medium	0.59(1.09)	0.39(0.25)	0.28(0.14)	0.23(0.09)	0.47(1.15)	0.26(0.24)	0.13(0.10)	0.10(0.08)
	Strong	0.71(0.60)	0.58(0.38)	0.48(0.41)	0.41(0.27)	0.64(1.03)	0.37(0.48)	0.34(0.42)	0.22(0.19)
	Mean	0.61(0.73)	0.44(0.27)	0.35(0.22)	0.29(0.18)	0.51(0.92)	0.26(0.37)	0.19(0.22)	0.14(0.12)
Softness	Weak	0.52(1.08)	0.37(0.24)	0.24(0.11)	0.22(0.10)	0.39(1.13)	0.24(0.42)	0.12(0.09)	0.08(0.07)
	Medium	0.70(0.36)	0.40(0.20)	0.33(0.17)	0.31(0.17)	0.64(0.43)	0.25(0.26)	0.16(0.11)	0.13(0.11)
	Strong	1.09(0.75)	0.69(0.49)	0.59(0.26)	0.47(0.25)	1.01(0.97)	0.49(0.62)	0.41(0.24)	0.26(0.18)
	Mean	0.77(0.73)	0.48(0.31)	0.38(0.18)	0.33(0.13)	0.68(0.84)	0.33(0.43)	0.23(0.16)	0.16(0.12)
Brokenness	Weak	0.59(0.98)	0.37(0.23)	0.25(0.14)	0.24(0.12)	0.48(1.04)	0.24(0.40)	0.12(0.10)	0.10(0.09)
	Medium	0.42(0.29)	0.43(0.22)	0.28(0.16)	0.27(0.15)	0.27(0.35)	0.27(0.27)	0.14(0.11)	0.14(0.11)
	Strong	1.42(1.06)	1.07(0.47)	0.68(0.31)	0.65(0.27)	1.55(1.84)	0.88(0.72)	0.52(0.32)	0.36(0.26)
	Mean	0.81(0.78)	0.63(0.31)	0.40(0.20)	0.39(0.18)	0.76(1.08)	0.46(0.46)	0.24(0.16)	0.19(0.15)
Colorfulness	Weak	0.48(0.64)	0.36(0.18)	0.24(0.11)	0.22(0.12)	0.36(0.78)	0.21(0.24)	0.11(0.09)	0.09(0.09)
	Medium	1.67(2.29)	0.60(0.50)	0.48(0.18)	0.42(0.16)	1.56(2.07)	0.57(1.06)	0.24(0.14)	0.21(0.13)
	Strong	1.20(0.99)	0.78(0.57)	0.56(0.31)	0.57(0.32)	1.34(2.33)	0.72(1.00)	0.46(0.48)	0.37(0.35)
	Mean	1.12(1.31)	0.58(0.41)	0.43(0.20)	0.40(0.20)	1.09(1.73)	0.50(0.77)	0.27(0.23)	0.22(0.19)
Other		0.38(0.52)	0.35(0.16)	0.19(0.06)	0.19(0.08)	0.25(0.58)	0.16(0.22)	0.06(0.02)	0.07(0.04)

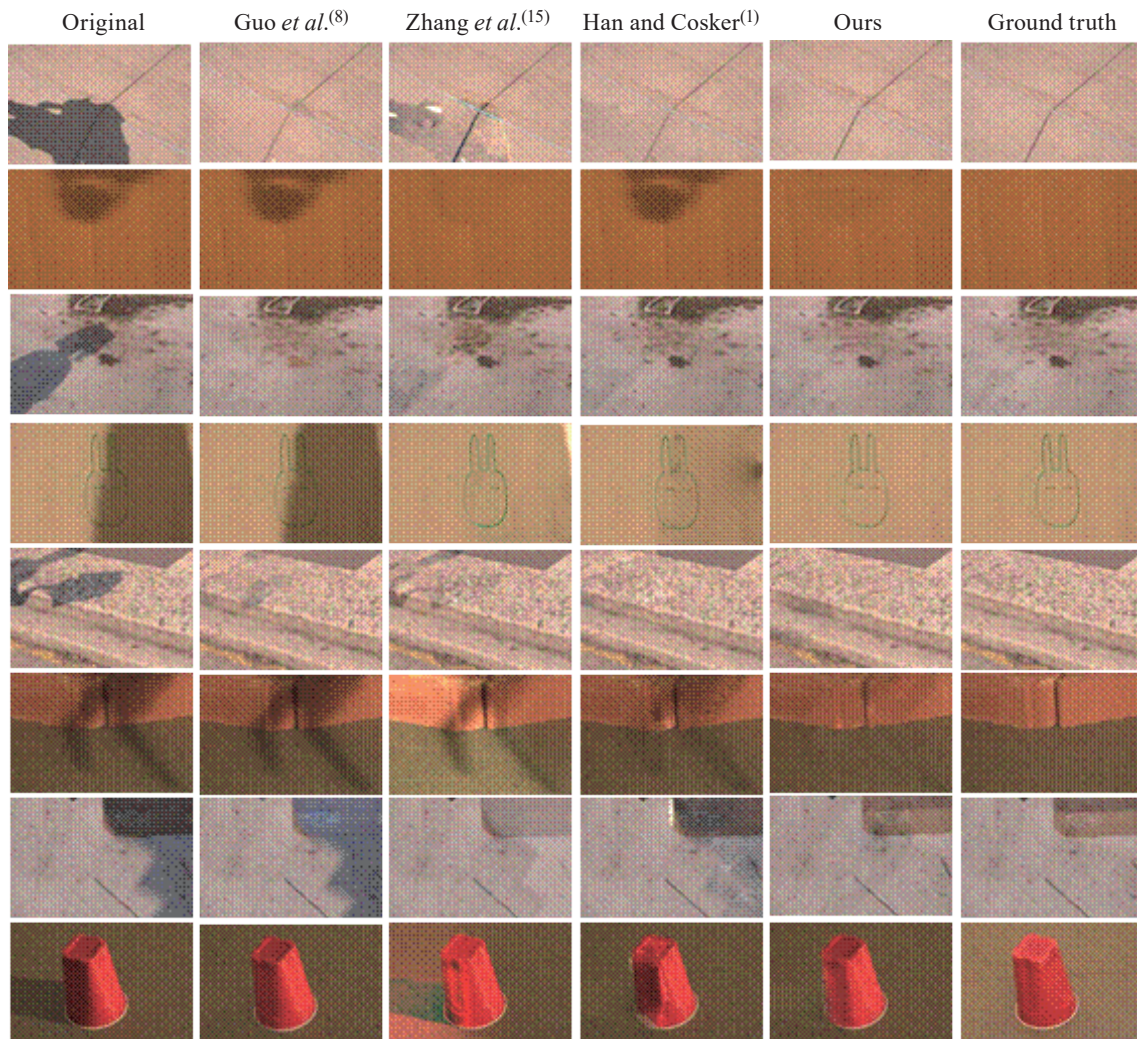


Fig. 8. (Color online) Comparisons of our method with existing state-of-the-art techniques using data set from Ref. 1.

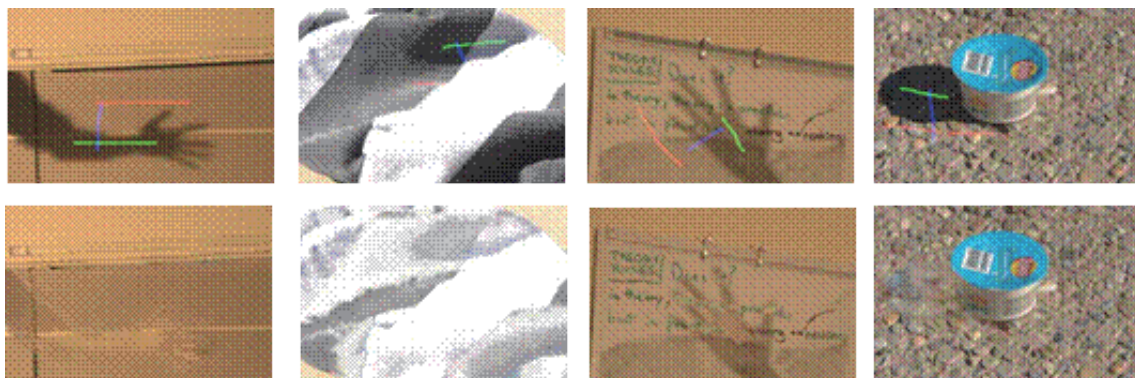


Fig. 9. (Color online) Special cases where the proposed method fails to remove the shadow.

5. Conclusion

In this paper, we propose a user-friendly shadow removal method that involves simple and powerful shadow detection based on user inputs. The strong offset correction method is adopted to realize fast shadow removal. After removing the shadow, the image is enhanced and filtered to obtain a high-quality shadowless image.

This algorithm provides some innovations in the field of shadow image processing and recognition as follows. (1) Simple user inputs and intelligent image pixel intensity sampling are used. (2) A novel robust and simple user-indicated color sampling line offset correction is used to correct artifacts. (3) By introducing the statistical distribution of shadow-free areas, changes in illumination can be estimated robustly and effectively for shadow removal and re-illumination without prior training. To evaluate the proposed method, we use the standard data sets in the field. We also visually compare the results of partial shadow removal in the evaluation data set. Through the analysis in this article, some specific situations where it is difficult to remove shadows are identified, for example, soft broken shadows and shadows with strong texture. Future work may be to combine user scribbles and machine learning methods to remove shadows in images.

Acknowledgments

This work was supported in part by the Teaching Quality and Reform of Higher Vocational Education Project of Guangdong Province (GDJG2019463), Hebei Provincial High Level Talents Foundation of Hebei Province (A2016002025), Characteristic Innovation Project of Guangdong Provincial Department of Education (2019KTSCX201), Natural Science Foundation of Characteristic Innovation Project of Guangdong Ordinary University and the Science and Technology Research Project of Hebei Higher Education (ZD2017213), subsidized projects for overseas students (CL201705), Zhaoqing Research and Development Technology and Application of Energy Conservation and Environmental Protection Ecological Governance (2020SN004), Youth Innovative Talents Project of Guangdong Provincial Department of Education (2018KQNCX290), and Key-Area Research and Development Program of Guangdong Province (2020B0909050003).

References

- 1 G. Han and D. Cosker: *J. Opt. Soc. Am. A Opt. Image. Sci. Vis.* **33** (2016) 1798. <https://doi.org/10.1364/JOSAA.33.001798>
- 2 I. Omer and M. Werman: *Proc. 2004 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2004)* 946–953.
- 3 L. Q. Qu, J. D. Tian, S. F. He, Y. D. Tang, and R. W. H. Lau : *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2017)* 4067–4075.
- 4 J. F. Wang, X. Li, and J. Yang: *Proc. 2018 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2018)* 1788–1797.
- 5 X. Hu, Y. Jiang, C. Fu, and P. Heng: *Proc. 2019 IEEE Int. Conf. Computer Vision (IEEE, 2019)* 2472–2481.
- 6 G. D. Finlayson, M. S. Drew, and C. Lu: *Int. J. Comput. Vision* **85** (2009) 35. <https://doi.org/10.1007/s11263-009-0243-z>

- 7 J. Zhu, K. G. G. Samuel, S. Masood, and M. F. Tappen: Proc. 2010 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2010) 223–230.
- 8 R. Guo, Q. Dai, and D. Hoiem: IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013) 2956. <https://doi.org/10.1109/TPAMI.2012.214>
- 9 T. P. Wu, C. K. Tang, M. S. Brown, and H. Y. Shum: ACM Trans. Graph. **26** (2007) 7. <https://doi.org/10.1145/1243980.1243982>
- 10 C. Fredembach and G. Finlayson: Proc. 2006 IEEE Computer Society (IEEE, 2006) 832–835.
- 11 M. Drew, C. Lu, and G. Finlayson: Proc. 2006 Int. Conf. Multimedia and Expo. (ICME, 2006) 257–260.
- 12 Y. Shor and D. Lischinski: Comput. Graph. Forum. **27** (2008) 577. <https://doi.org/10.1111/j.1467-8659.2008.01155.x>
- 13 E. Arbel and H. Hel-Or: IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 1202. <https://doi.org/10.1109/TPAMI.2010.157>
- 14 F. Liu and M. Gleicher: Proc. 2008 Euro. Conf. Computer Vision (ICCV, 2008) 437–450.
- 15 L. Zhang, Q. Zhang, and C. Xiao: IEEE Trans. Image. Process. **24** (2015) 4623.
- 16 H. G. Barrow and J. M. Tenenbaum: Proc. 1978 Int. Conf. Computer Vision Systems (ICVS, 1978) 3–26.
- 17 B. A. Maxwell, R. M. Friedhoff, and C. A. Smith: Proc. 2008 IEEE Conf. Computer Vision and Pattern Recognition. (IEEE, 2008) 1–8.
- 18 X. Yu, G. Li, Z. Ying, and X. Guo: Proc. 2017 17th Int. Conf. Computer Analysis of Images and Patterns (CAIP, 2017) 307–319.
- 19 S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri: IEEE Trans. Pattern Anal. Mach. Intell. **38** (2016) 431. <https://doi.org/10.1109/TPAMI.2015.2462355>
- 20 R. Fattal: ACM Trans. Graph. **34** (2014) 13. <https://doi.org/10.1145/2651362>
- 21 D. Berman, T. Treibitz, and S. Avidan: Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 1674–1682.
- 22 C. Xiao, R. She, D. Xiao, and K. Ma: Comput. Graph. Forum **32** (2013) 207. <https://doi.org/10.1111/cgf.12198>
- 23 F. Zeev, F. Raanan, L. Dani, and S. Richard: ACM Trans. Graph. **27** (2008) 1. <https://doi.org/10.1145/1360612.1360666>
- 24 G. Han and D. Cosker: Image Vision Comput. **62** (2017) 19. <https://doi.org/10.1016/j.imavis.2017.04.001>