

USER-AIDED SINGLE IMAGE SHADOW REMOVAL

Han Gong, Darren Cosker, Chuan Li, and Matthew Brown

University of Bath, Department of Computer Science
Bath, BA2 7AY, United Kingdom
{h.gong, d.p.cosker, c.li, m.brown}@bath.ac.uk

ABSTRACT

This paper presents a novel user-aided method for texture-preserving shadow removal from single images which only requires simple user input. We show better uneven shadow boundary processing and umbra recovery over the state-of-art. We first detect an initial shadow boundary by growing a user specified shadow outline on an illumination-sensitive image. Interval-variable intensity sampling is introduced to avoid artefacts raised from uneven boundaries. We then get the initial scale field by applying local group intensity spline fittings around the shadow boundary. Bad intensity samples are replaced by their nearest alternatives based on a log-normal probability distribution of fitting errors. Finally, we use a gradual colour transfer to correct post-processing artefacts such as gamma correction and lossy compression. Compared with the state-of-art methods, we offer highly user-friendly interaction, produce improved umbra recovery and improved processing given uneven shadow boundaries.

Index Terms— shadow removal, single image, user-aided

1. INTRODUCTION

Shadows are ubiquitous in natural scenes, and their removal is an interesting and important area of research. As well as attempting to solve this problem for image editing by e.g. artists, shadows affect many computer vision algorithms, e.g., unwanted shadow boundary’s causing artefacts in image segmentation and contributing to drift issues when tracking given moving scenes.

There are several automatic methods for shadow detection and removal. For example, intrinsic image based methods [1, 2] and features, e.g. intensity and texture, learning based methods [3]. However, automatic shadow detection is ill-posed and currently unreliable to the point that even humans can have difficulty in recognising shadowed areas [4].

This paper focuses on user-aided single image shadow removal. User-aided methods generally achieve better shadow detection and removal at the cost of user input. Many of

them [4, 5, 6, 7] are texture-preserving. Most of the past work require precise input defining the shadow boundary and have issues in properly relighting the umbra.

According to past work [1], shadow effects can be represented as an additive scale field \mathcal{S}_c in the log domain. In the log domain, an image $\tilde{\mathcal{I}}_c$ with shadow effects added to an original image \mathcal{I}_c can be represented as follows:

$$\tilde{\mathcal{I}}_c(x, y) = \mathcal{I}_c(x, y) + \mathcal{S}_c(x, y) \quad (1)$$

where $c \in \{R, G, B\}$ indicates each RGB colour space channel, x and y are the coordinates of pixels. The scales of lit area pixels are 0 and the other pixel scales are negative numbers.

The umbra is the darkest part of the shadow while the penumbra is the wide outer boundary with a nonlinear intensity change between the umbra and lit area [8]. The penumbra scale is non-uniform and shadowed surface textures generally become weaker [6]. Contrast artefacts can also appear in shadow areas due to image post-processing [7, 6, 8]. Arbel *et al.* [7, 8] developed curved surface shadow removal by formulating scale field estimation as thin-plate surface fitting that requires given shadow feature points. Wu *et al.* [4] proposed a shadow removal method that requires a given quadmap and applied a Bayesian optimisation framework. Mohan *et al.* [5] apply a curve-fitting model and remove shadows in the gradient domain. Given precise shadow boundaries, Liu *et al.* [6] apply spline-fitting and texture transfer to remove shadow. Shor *et al.* [9] proposed a shadow removal method that requires one shadow pixel but does not preserve texture shadows. Although some past work preserves penumbra texture, umbra removal and uneven boundary processing are still problematic. Assuming a uniform umbra, Liu *et al.* [6] can often introduce over-saturation artefacts in some cases. Uneven shadow boundaries may affect penumbra detection and scale estimation. Most user-aided methods to assist boundary detection [4, 6, 8] require careful highlighting of the boundary. We propose a method that requires one rough stroke to mark an umbra sample. It can process uneven shadow boundaries and achieves better umbra removal compared with past work. Our major contributions are as follows:

Easy user input. Past work, e.g. [4, 6, 7, 8], requires precise user-inputs defining the shadow boundary. Our

This work is funded by the China Scholarship Council and the University of Bath.

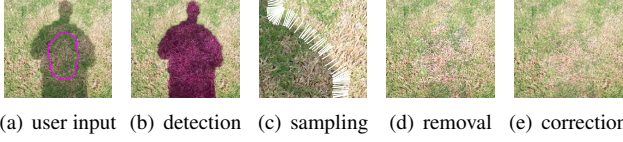


Fig. 1. An overview of our approach. (a): user input indicated by a purple curve, (b): initial shadow boundary detection indicated by the purple mask (§2.1), (c): interval-variable sampling lines indicated by the white lines (§2.2.1), (d): shadow removal by estimating the scale field of the shadow area (§2.2.2). Note that the area’s contrast still appears different from its surroundings. (e): image correction by our gradual colour transfer (§2.3). The input image is taken from [8].

method only requires an umbra segment highlighted by one rough stroke and grows it on an illumination-sensitive image to obtain initial shadow boundaries.

Interval-variable sampling. Past work, e.g. [6, 8], applies interval-fixed sampling around penumbra that causes artefacts near uneven shadow boundaries. To address this, we develop interval-variable sampling according to shadow boundary curvature.

Local group optimisation for picked samples. Inspired by Arbel’s [7, 8] and Liu’s [6] spline-fitting idea, we propose a local group optimisation that balances curve fitness and local group similarity. Unlike all past work, we filter inferior samples that are replaced with their closest alternatives according to a log-normal probability distribution. This reduces shadow removal artefacts.

Gradual colour transfer. Post-processing effects cause inconsistent shadow removal compared with the lit areas both in tone and in contrast. We make use of statistics in thin penumbra boundaries and the shadow scale field to correct these issues.

2. PROPOSED METHOD

Given an input image and a user specified umbra segment (Fig. 1(a)), we detect the initial shadow boundaries by expanding the given umbra segment on an illumination image using an active contour (Fig. 1(b), §2.1). To keep boundary details, we sample pixel intensities along the sampling lines perpendicular to the shadow boundary (Fig. 1(c), §2.2.1) at variable intervals. We perform a local group optimisation to estimate the illumination change which refines shadow boundary detection and provides an initial scale field. According to an adaptive sample quality threshold, sampling lines with bad samples are replaced by their nearest neighbours and a later local group optimisation is applied for them. Finally, we relight the shadow area using our scale field

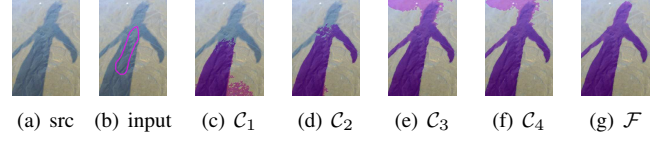


Fig. 2. An illumination-sensitive image \mathcal{F} is fused from the four colour channels of the image (a) (taken from [6]) using the intensity statistics of the given umbra region (b) inside the purple boundary. We grow the region using the image \mathcal{F} to the entire shadow area as indicated in (g) by the purple mask. Sub-figures (c-f) are the region growing results on the corresponding four single channels.

(Fig. 1(d), §2.2.2) and correct post-processing artefacts using our gradual colour transfer (Fig. 1(e), §2.3).

2.1. Initial Shadow Boundary Detection

Determining the initial shadow boundary is the first step of penumbra detection and is required in many previous methods including [7, 8, 6]. In this subsection, we explain how to derive an initial shadow boundary from a given rough umbra sample segment.

Inspired by [10], we fuse four normalised candidate illumination-sensitive channels from different colour spaces into an illumination image. The chosen channels are: the V channel (\mathcal{C}_1) of HSV space, the V channel (\mathcal{C}_4) of LUV space, and the Y channel (\mathcal{C}_2) and Cb channel (\mathcal{C}_3) of YCbCr space. We measure the confidence values of each candidate channel using an exponential incentive function φ representing the texture of each of their umbra sample segments:

$$\varphi(x) = x^{-\lambda} (\lambda > 0) \quad (2)$$

where x is the pixel intensity, λ (default value 5) determines the steepness of the incentive function. Lower texture is preferred as it means higher intensity uniformity of umbra segment. The texture is measured by standard derivation of intensities. The fused image \mathcal{F} is computed as a weighted sum of each normalised candidate channel \mathcal{C}_l as follows:

$$\mathcal{F} = \left(\sum_{l=1}^4 \mathcal{C}_l \varphi(\sigma_l) \right) / \left(\sum_{l=1}^4 \varphi(\sigma_l) \right) \quad (3)$$

where l is the channel index, σ_l is the standard derivation of the umbra sample intensities of \mathcal{C}_l . To avoid texture noise, we apply a bilateral filter [11] to \mathcal{F} first. We grow a sparse-field active contour [12] on the fused image to detect the initial shadow boundary. As shown in Fig. 2, region-growing using the fused image is more robust than relying on one channel.

2.2. Scale Field Estimation

This subsection describes our scale field estimation from initial shadow detection. Shadow affects are represented by



Fig. 3. (a) is the original patch. (b) is the result of sampling every shadow boundary pixel, which results in shadow removal noise near the boundary. (c) is the noise-free result with a larger sampling interval.

varying (or different scale) intensity values. Using a scale field better represents the penumbra and umbra variations, and is used to relight the shadow area using Eq. 1. In § 2.2.1, we first sample the log domain pixel intensity along the sampling lines perpendicular to the shadow boundary. In § 2.2.2, we adopt a local group spline fitting optimisation through the measured sampling line pixel intensities to estimate sparse scales from the initial intensity samples. We replace bad intensity samples with their nearest alternatives and re-optimize for them. We spread the sparse scales to dense scales followed by a gradual colour transfer in § 2.3 that adjusts the colour and texture of initial shadow removal.

2.2.1. Interval-variable Sampling

According to Eq. 1, the logarithms of the original image is supplied for sampling. We sample pixel intensities along the lines perpendicular to the initial shadow boundary as shown in Fig. 4(c). Uneven boundaries can result in non-smooth vector normal estimations along the shadow boundary. To overcome this, we apply cubic spline smoothing to the initial boundary points before we compute their normals and curvatures. Under-sampling along the boundary neglects sharp details and causes artefacts as shown in Fig. 4 while over-sampling incurs penumbra removal noise due to texture as shown in Fig. 3. More sparse pixel scales are computed for curvy boundary parts for precise in-painting. To avoid texture artefacts, we apply a bilateral filter [11] to the input image before sampling. Unlike past work [5, 6, 8, 7], we do not adopt a fixed sampling line interval, e.g., one sampling line per boundary pixel. Our method adjusts the sampling interval according to the curvature of the smoothed boundary. The interval is the same for all RGB channels. We set a curvature accumulator for shadow boundary points and accumulate along the boundary. We place sampling marks and reset the accumulator when the curvature sum reaches a threshold ξ (default value 0.05). We achieve this by adopting Eq. 4. We limit the absolute curvature of each boundary point with an upper limit ξ and compute a cumulative sum array Q of the saturated absolute curvature array. To determine the sampling interval, we choose boundary points where the sampling lines pass through as follows:

$$\begin{cases} \tilde{Q}_m = \lfloor Q_m / \xi \rfloor & (m \leq N, m \in \mathbb{N}) \\ D_n = \tilde{Q}_{n+1} - \tilde{Q}_n & (n \leq N - 1, n \in \mathbb{N}) \end{cases} \quad (4)$$

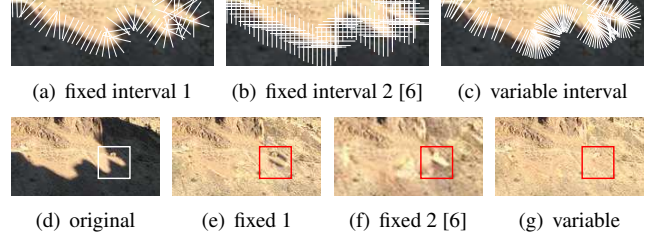


Fig. 4. The white lines in (a), (b), and (c) are the sampling lines of the fixed interval using boundary-perpendicular sampling, the fixed interval using horizontal/vertical sampling in [6], and our boundary-perpendicular variable interval sampling respectively. (d) is the original image. (e), (f), and (g) are the corresponding shadow removal results of the three sampling methods respectively.

where N is the number of boundary points, m and n specify the index of boundary points, \tilde{Q} is the array of the quantized and normalised cumulative sum Q , D is the adjacent element differences array of \tilde{Q} . To get the interval of sampling marks, we set the marks for the first and last boundary points and all the points in D with non-zero values. If the boundary is a straight line, the initial interval is fixed up to a maxima (five boundary points). As shown in Fig. 4, our variable sampling interval avoids penumbra removal artefacts around sharp boundary parts.

2.2.2. Illumination Variance Estimation

Having obtained sparse intensity samples at different positions along the boundary, our goal is to find illumination scaling values inside the umbra, penumbra and lit area. We model the illumination scale change S_i for each i^{th} intensity sample of each RGB channel as follows (see also Fig. 5):

$$S_i(x) = \begin{cases} K & x \leq x_1 \\ f(x) & x_1 < x \leq x_2 \\ 0 & x > x_2 \end{cases} \quad (5)$$

where x is a pixel location within the sampling line, x_1 and x_2 determine the start and end of the penumbra area respectively, and K is a negative scale constant for sample points within the umbra area ($x < x_1$). The constant 0 is assumed for the lit area piece ($x > x_2$) as this falls inside a lit area of the image and does not require re-scaling. In order to solve x_1 , x_2 , K , and f , we solve a piece-wise function G_i parameterised by v_1 , v_2 , v_3 , and v_4

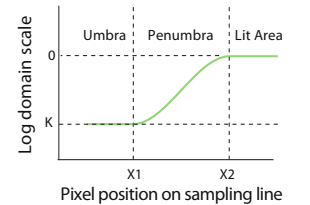


Fig. 5. Scale model

(recall that our intensity samples are in the log domain):

$$\begin{cases} G_i(x) = S_i(x) + v_4 \\ f(x) = v_1 B(v_2(x - v_3)) \\ B(y) = -0.25y^3 + 0.75y - 0.5 \\ \begin{bmatrix} x_1 & x_2 \end{bmatrix} = v_3 + v_2^{-1} \begin{bmatrix} -1 & 1 \end{bmatrix} \\ K = -v_1 \end{cases} \quad (6)$$

where B is a cubic shape function (a sinusoidal function here also produces adequate results) and y is the input. Our solution is thus reduced to solving v_1, v_2, v_3 , and v_4 for each RGB channel (twelve in total). Illumination of each channel may vary differently while penumbra boundaries of three channels are usually the same. We thus assume a common penumbra width and position for each channel, determined by v_2 and v_3 respectively. We formulate solving the eight remaining parameters as an optimisation problem which balances curve fitness and local group fitting similarity. We minimise the energy function E_i for the i^{th} sampling line as follows:

$$E_i = \alpha_1 E_{fit}(V_i, Z_i) + \alpha_2 E_{gs}(V_i, V_{i-1}, \dots, V_{i-r-1}) \quad (7)$$

where r is the number of members in a local sampling line group (default value 5), α_1 and α_2 are two balancing weights (default values 1 and 0.2 respectively), E_{fit} measures the sum-of-square fitting error between the three piece-wise functions G_i (defined by parameter vector V_i) and the original three-channel intensity sample matrix Z_i , E_{gs} measures the parameter similarity between the neighbouring members of a local group. In practice, penumbra width affects the removal quality most significantly. We thus only compare the similarity of the shared v_2 in Eq. 6. E_{gs} is defined as follows:

$$\begin{cases} V_i = [v_{R1}, v_{G1}, v_{B1}, v_{RGB2}, v_{RGB3}, v_{R4}, v_{G4}, v_{B4}] \\ E_{gs}(V_i, V_{i-1}, \dots, V_{i-r-1}) = \left(\frac{1}{b_i} - \frac{\sum_{k=1}^r \varphi(\dot{e}_{i-k})}{\sum_{j=1}^r \varphi(\dot{e}_{i-j}) b_{i-j}} \right)^2 \end{cases} \quad (8)$$

where v_{R1}, v_{G1} and v_{B1} are the v_1 for each channel, v_{R4}, v_{G4} and v_{B4} are the v_4 for each channel, v_{RGB2} and v_{RGB3} are the shared v_2 and v_3 for all channels, b_i is the v_{RGB2} of the parameter vector V_i , \dot{e}_u indicates the fitting error of the previous u^{th} fittings, φ is the function defined in Eq. 2. We solve this using a sequential quadratic programming algorithm [13].

However, interval-variable sampling can not always guarantee good sample quality. Strong surface textures introduce more significant intensity change than illumination change. Unlike past work [5, 6, 8, 7], we ignore sampling lines with higher fitting errors and pick their most suitable neighbours. Based on our empirical tests on various images, we model the initial fitting error distribution as a log-normal probability distribution. As the initial fitting error is distributed log-normally, we can convert it to its corresponding normal distribution by taking the exponential of it. According to the empirical 3-sigma rule of normal distribution, we ignore sampling

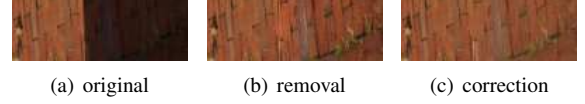


Fig. 6. (b) is the initial shadow removal of (a) and is corrected by our gradual colour transfer as shown in (c).

lines with fitting errors higher than a threshold $\mu + \sigma$ which accounts for 15.8% of all samples where μ and σ are the mean and standard deviation of errors. For each sampling line with bad samples, we only attempt to pick its nearest sampling lines within a short distance, i.e. no further than its neighbouring chosen sampling lines. To evaluate the replacements' quality and compute their sparse scales, we apply the same optimisation method described previously.

After optimisation, we obtain sparsely distributed scales, defined in Eq. 1, of all sampled pixels inside and around the penumbra area. We interpolate the scales within the penumbra area using linear interpolation and extrapolate the other scales in the lit and umbra area using in-painting [14].

2.3. Gradual Colour Transfer

In practice, the theoretical shadow affect formulation does not often hold. Image acquisition devices usually apply postprocessing, e.g. Gamma correction. Lossy compressions, e.g. JPEG, are also common in images such that compression artefacts (e.g. affecting contrast) in the shadow area become noticeable when removal is applied. To address this, we extend the colour transfer in [15] with scale field S_m . We compute the normalised scale increase h_i of the i^{th} sampling line according to Eq. 5 as follows:

$$h_i(x) = (\exp(f_i(x)) - \exp(K_i)) / (1 - \exp(K_i)) \quad (9)$$

where x is the location of pixel of a sampling line, K_i and f_i are respectively the lit constant K and the cubic function piece of the i^{th} sampling line. We apply the same method described in § 2.2.2 to interpolate and extrapolate sparse scale increase values computed by Eq. 9 to form a dense scale increase field S_m . We convert the initial shadow removal image from RGB space to LAB space. For each LAB channel, we compute the mean μ_u and deviation σ_u of the umbra side pixel intensities near penumbra as the source and we compute the mean μ_t and deviation σ_t of the lit side pixels near penumbra as the target. We adjust the initial removal image channel L to the final image channel \tilde{L} as follows:

$$\begin{cases} \mu_s(x, y) = \mu_u + S_m(x, y)(\mu_t - \mu_u) \\ \sigma_s(x, y) = \sigma_u + S_m(x, y)(\sigma_t - \sigma_u) \\ \tilde{L}(x, y) = \mu_t + (L(x, y) - \mu_s(x, y))\sigma_t/\sigma_s(x, y) \end{cases} \quad (10)$$

where x and y are the coordinates of pixels, μ_s and σ_s are the fields of gradual source mean and deviation. We show an example of colour transfer in Fig. 6.

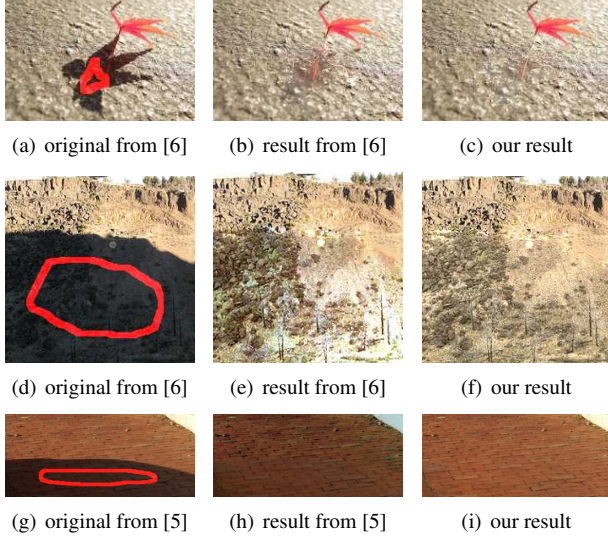


Fig. 7. From (a), the red light component that passes through semi-transparent object is still in (b) and is reduced by our method in (c). Ours also removes shadow residuals near the highly uneven boundary in (b). The shadow removal result of (d) shown in (e) results in over-saturation of the umbra. Also, the texture and colour variations across the shadow boundary are not smooth and consistent. We overcome this issue as shown in (f). Our removal result of (g), as shown in (i), is improved over [5] as shown in (h).

3. RESULTS

Due to the lack of definitive ground truth from past work, we visually compare our shadow removal results with state-of-art methods using the same input images and some other representative images. Our method has been tested on various photos from existing work and additional representative photos. Our method is highly user-friendly as shown in Fig. 1 and produces state-of-art quality shadow removal as shown in Fig. 1, 3, 4, 7, 6, 8, and 9. For each shadow area, we require one stroke, shown as red curves in figures, marking an characteristic umbra segment. The stroke can be very rough and does not necessarily need to follow the shadow boundary. We also overcome some cases that past work fail to handle.

In our comparisons, we focus primarily on more recently studied texture-preserving methods as opposed to older texture-lossy ones such as in-painting [2] and zero-gradient [16]. Our previous figure (Fig. 4) has already highlighted the issues caused by uneven boundary processing. This compares our variable sampling interval – which is boundary-perpendicular – against our own test of fixed interval boundary-perpendicular sampling and the fixed interval vertical and horizontal sampling method in [6].

Compared with [6], our method reduces the red light component that passes through the semi-transparent leaf as illustrated in Fig. 7. The redness of different parts of the leaf is

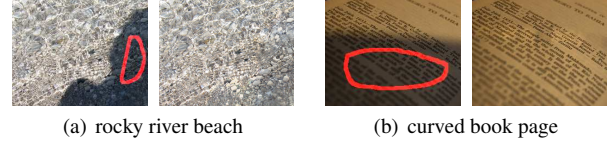


Fig. 9. Failures. Left: original images; Right: our results.

different and thus the amount of passed red light is not uniform. To handle this, we assume non-uniform scales in the umbra, i.e. different umbra scale constants as mentioned in § 2.2.2, and process the RGB channels separately based on a shared penumbra area. Our interval-variable sampling (see Fig. 7(c)) removes some minor residual shadow fragments near the shadow boundary (noticeable in Fig. 7(b)). In Fig. 7(d), the shadowed sandy surface is consistently recovered. As the surface in the lit area is not saturated, the same surface beneath the shadow should also not be saturated in Fig. 7(e). Fig. 7(f) shows our method avoids over-saturation artefacts and achieves more consistent texture and smoother colour variance across the shadow boundary. In Fig. 7(h), the result from [5] appears darker in both the shadow and lit areas. Our result shows a consistently coloured texture between the lit area and shadowed area. Figs. 8 and 9 demonstrate our results on images with various textures, reflectances, and shadow boundaries. Fig. 8(a) shows the removal of soft shadow cast on the curved surface and the texture consistency. Fig. 8(b) reveals our smooth and texture-consistent shadow removal applied to the earlier example illustrated in Fig. 4. The colour of the tree bottom of the hill is consistent with the trees at the hill top. The smoothness of the hill-side’s colour and texture are also recovered. In Fig. 8(d), Fig. 8(e), Fig. 8(c), and Fig. 8(f), the texture and self-shadows are kept after removal.

However, our method still shows minor limitations in highly complex cases (still unsolved in state of the art work). In Fig. 9(a), the river bed is recovered but the ripple highlights in the shadow area are missing. These highlights are mainly produced by light reflection on the wavy water surface and the light refraction from the river bed. As the directed light is blocked in the shadow area, these complex affects cannot be recovered by simple relighting. In another challenging case in Fig. 9(b) with non-white light, the book texture in the wide penumbra area is retained with minor over-saturation artefacts. This is due to the strong broken-shadow-like text texture which affects the spline fitting.

4. CONCLUSION

We present a user-friendly texture-preserving shadow removal method that overcomes some common limitations from the past work. Specifically, our method retains shadowed texture and preforms well on highly-uneven shadow boundaries, non-uniform umbra illumination, and non-white lighting. Our main technical contributions are (1) highly user-friendly in-

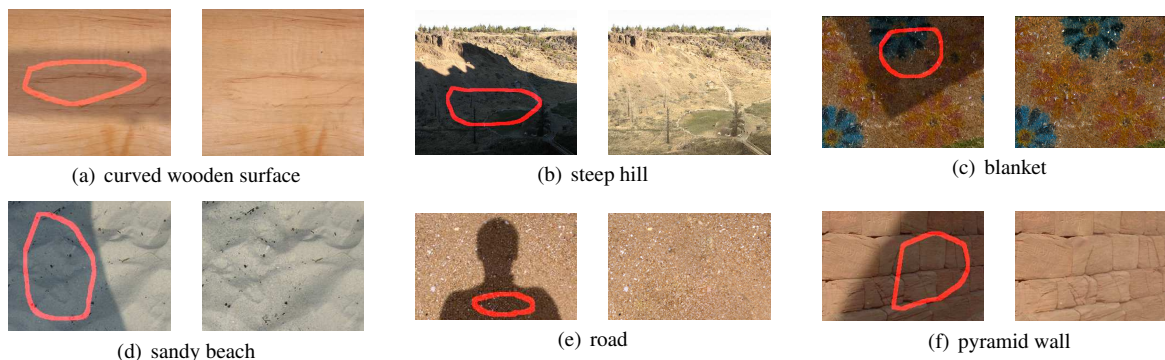


Fig. 8. Demonstrations. Left: original images; Right: our results. The original images in (a) and (c) are from [8] and the original image in (b) and (f) are from [6] and [5] respectively.

put design; (2) interval-variable sampling; (3) local group optimisation; (4) gradual colour transfer. In future work, we would like to focus on more complex cases, such as highly broken shadows, shadowed surfaces with very strong shadow-like textures, and complex reflections in transparent scenes.

5. REFERENCES

- [1] Y. Weiss, “Deriving intrinsic images from image sequences,” in *Proc. Eighth IEEE Int. Conf. Computer Vision 2001*, 2001, vol. 2, pp. 68–75.
- [2] G. D. Finlayson, S. D. Hordley, Cheng Lu, and M. S. Drew, “On the removal of shadows from images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.
- [3] Ruiqi Guo, Qieyun Dai, and D. Hoiem, “Single-image shadow detection and removal using paired regions,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 2033–2040.
- [4] Tai-Pang Wu and Chi-Keung Tang, “A bayesian approach for shadow extraction from a single image,” in *Proc. IEEE Int. Conf. Computer Vision*, 2005, vol. 1, pp. 480–487.
- [5] Ankit Mohan, Jack Tumblin, and Prasun Choudhury, “Editing soft shadows in a digital photograph,” *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 23–31, 2007.
- [6] Feng Liu and Michael Gleicher, “Texture-consistent shadow removal,” in *Proc. of European Conf. on Computer Vision (ECCV)*, 2008, pp. 437–450.
- [7] E. Arbel and H. Hel-Or, “Shadow removal using intensity surfaces and texture anchor points,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1202–1216, 2011.
- [8] E. Arbel and H. Hel-Or, “Texture-preserving shadow removal in color images containing curved surfaces,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [9] Yael Shor and Dani Lischinski, “The shadow meets the mask: Pyramid-based shadow removal,” *Comput. Graph. Forum*, vol. 27, no. 2, pp. 577–586, 2008.
- [10] I. Katramados, S. Crumpler, and T.P. Breckon, “Real-time traversable surface detection by colour space fusion and temporal analysis,” in *Proc. Int. Conf. on Computer Vision Systems*, 2009, vol. 5815, pp. 265–274.
- [11] Sylvain Paris and Frédo Durand, “A fast approximation of the bilateral filter using a signal processing approach,” *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.
- [12] Ross T. Whitaker, “A level-set approach to 3d reconstruction from range data,” *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.
- [13] J. Nocedal and S.J. Wright, “Numerical optimization,” Springer series in operations research, chapter 18. Springer, second edition, 2006.
- [14] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, “Image inpainting,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, SIGGRAPH ’00, pp. 417–424.
- [15] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [16] Graham D. Finlayson, Steven D. Hordley, and Mark S. Drew, “Removing shadows from images using retinex,” in *Color Imaging Conference*, 2002, pp. 73–79.