

User-Assisted Image Shadow Removal

Han Gong^{a,*}, Darren Cosker^b

^a*School of Computing Sciences, University of East Anglia, Norwich, UK*

^b*Department of Computer Science, University of Bath, Bath, UK*

Abstract

This paper presents a novel user-aided method for texture-preserving shadow removal from single images requiring simple user input. Compared with the state-of-the-art, our algorithm offers the most flexible user interaction to date and produces more accurate and robust shadow removal under thorough quantitative evaluation. Shadow masks are first detected by analysing user specified shadow feature strokes. Sample intensity profiles with variable interval and length around the shadow boundary are detected next, which avoids artefacts raised from uneven boundaries. Texture noise in samples is then removed by applying local group bilateral filtering, and initial sparse shadow scales are estimated by fitting a piecewise curve to intensity samples. The remaining errors in estimated sparse scales are removed by local group smoothing. To relight the image, a dense scale field is produced by in-painting the sparse scales. Finally, a gradual colour correction is applied to remove artefacts due to image post-processing. Using state-of-the-art evaluation data, we quantitatively and qualitatively demonstrate our method to outperform current leading shadow removal methods.

Keywords: image shadow removal, user-assisted computer vision, colour correction, curve fitting, smoothing.

2010 MSC: 68U10, 68U05

1. Introduction

Shadows are ubiquitous in natural scenes, and their removal is an interesting and important area of research. As well as a motivation to solve this problem for artistic image editing, shadows can affect the performance of many computer vision algorithms. For example, unwanted shadow boundaries can cause artefacts in image segmentation and contribute to drift when tracking given moving objects and scenes.

In this paper, a semi-automatic method is proposed for high-quality shadow removal using user-defined flexible single strokes covering the shadow and lit pixels. Our method sacrifices full autonomy for extremely simple user

input, as opposed to existing manual approaches that require fine-scale input, *e.g.* accurate shadow contours. Given detection, our method produces accurate shadow removal optimised for robust penumbra recovery. Using the current state-of-the-art shadow removal ground truth dataset [1], our solution is quantitatively evaluated against other leading methods and demonstrates notably improved performance. Numerous visual comparisons of our method versus existing methods are also presented, demonstrating qualitatively more pleasing results. Our approach represents what we believe to be a state of the art technique for shadow removal with a thorough evaluation against the current leading approaches.

1.1. Related Work

A shadow generally consists of an umbra and penumbra area. The umbra is the darkest part of the shadow

*Corresponding author

Email addresses: h.gong@uea.ac.uk (Han Gong),
dpc@cs.bath.ac.uk (Darren Cosker)

while the penumbra is the wide outer boundary with a gradual intensity change between the umbra and lit area. The penumbra scale is non-uniform and shadowed surface textures generally become weaker within it. A shadow image I_c^+ can be considered to be a Hadamard product of a shadow scale layer \mathcal{S}_c and a shadow-free image I_c^* as shown in Eq. 1.

$$I_c^+ = I_c^* \circ \mathcal{S}_c \quad (1)$$

For a lit pixel, the illumination is constant in both shadow and shadow-free images. For a shadow pixel, its intensity in a shadow image is lower than its intensity in the shadow-free image. Consequently, the scales \mathcal{S}_c of the lit area are 1 and other areas' scales are between 0 and 1.

However, most images appearing on the Internet are not linear images. These images have commonly been post-processed by some non-linear image processing algorithms such as gamma-correction, JPEG compression, and non-linear filters. After a linear shadow recovery process, contrast artefacts can appear in the shadow areas [2].

Approaches to shadow removal can be categorised as either automatic [1, 3–8] or user-aided [2, 9–11]. The problem can be broken down into two stages: shadow detection and shadow removal.

Automatic approaches do not require any user interaction but risk inaccurate shadow detection or require special setups for capture which do not work for general images. Intrinsic image based methods are a popular branch of automatic techniques (*e.g.* [3, 4]). The decomposition of intrinsic images provides shading and reflectance information but can be unreliable leading to over-processed results. The decomposition is generally based on an assumption that the illumination change is smooth or the reflectances of the scene lies on an illumination-invariant direction. Another branch of techniques are shadow feature learning based methods [1, 12–16]. However, detection can be often unreliable due to limited training data and the quality of initial image edge detection and segmentation. Several approaches [12–14, 16] detect shadows

by classifying edges in images using edge features, *e.g.* intensity, texture, chromaticity and intensity ratio. Graphical models [1, 15] can also form the basis of detection. Yao *et al.* [15] detect shadow by using a reliable graph model and colour features to classify pixels. In their approach, each pixel is a node with encodes node reliability based on strength of shadow feature, and node relationships described using similarity between neighbours. Guo *et al.* [1] detect shadows by classifying segments in images that adopt similar shadow features and remove shadows using a variant alpha-matting algorithm. Some methods apply additional active light sources to capture shadowless objects, *e.g.*, by comparing images with an illumination source at different positions [5] and comparing flash and no-flash image pairs [6]. However, active lighting restricts the types of scene that shadow removal can be applied to – as using special lighting setups outdoors is often not practical. Other methods adopt optical filters to acquire multi-spectral information to achieve illumination detection, *e.g.* by comparing NIR and RGB images [7] and by comparing RGB and single-colour-filtered images [8], but these methods are generally limited to special scenario cases, *e.g.* sunlight and non-black surfaces.

User-aided methods generally achieve better shadow detection and removal at the cost of user input. Wu *et al.* [9] require a high degree of user intervention where multiple regions of shadow, lit area, uncertainty and exclusion are identified. They apply a Bayesian optimisation to derive a shadow matte and a shadow-free image. Others [10, 11] require fine input defining the shadow boundary. Liu and Gleicher [10] proposed a curve fitting method and a global alignment of gradients to acquire shadow scales but have issues when relighting the umbra and can introduce artefacts at uneven boundaries. Shor and Lischinski [17] detect shadow using image matting from a grown shadow seed. They only require one shadow pixel as input, but have limitations in cases where the other shadowed surfaces are not surrounded by the initially detected surface

or when the penumbra is too wide. Arbel and Hel-Or [2] apply a thin-plate model to the intensity surface. They require users to specify multiple texture anchor points to detect the shadow mask but the input overhead increases when shadows are distributed in multiple regions. Su and Chen [11] developed a method to estimate shadow scales using dynamic programming. Their gradient alignment for intensity samples allows for less accurate user inputs compared with [9, 10]. They also provide a healing tool for users to manually amend artefacts in highly-curved shadow boundary segments. Gong and Cosker [18] introduced a fast approach which categorises intensity profiles into several sub-groups and derives the shadow scales for each of them. They require two types of scribbles for marking lit and shadow pixels. Similarly, Zhang *et al.* [19] require the same user input of [18]. However, their method requires a shadow matte (guided by the user’s scribbles) to identify shadows, which is sensitive to user-scribbles because their image matting is affected by pixel location.

To date, most shadow removal methods [2, 9–11, 19] have only been evaluated by visual inspection on some selected images – with only a few exceptions performing quantitative evaluation. Guo *et al.* [1] provided the first public ground truth data set for shadow removal and perform quantitative testing. However, their error measurement is variant to the size and darkness of shadows and some of their shadow-free ground truth shows inconsistent illumination compared with the lit area of their corresponding shadow images.

1.2. Contributions

Given our overview of state of the art approaches, 3 main contributions are proposed:

1) Simple user input: Past work, *e.g.* [2, 9–11], requires precise user-input defining the shadow boundary. Our method only requires users to define some single rough strokes covering related shadow and lit pixels – without the need to differentiate between samples in shadow and

lit areas.

2) Intelligent sampling: Adaptive sampling with variable intervals and lengths is proposed to address shadow boundary artefacts in past work [2, 10], which uses fixed intervals and lengths. Unlike past work [2, 10, 11], unqualified samples are intelligently filtered. These can affect the quality of shadow scale estimation, *e.g.* samples with high noise or sampling lines passing through boundaries caused by occlusions or strong background texture.

3) Robust scale estimation Fast local group processing is proposed for selected samples and initially estimated scales to improve smoothness of shadow removal. Post-processing effects cause inconsistency in shadow corrected areas compared with the lit areas both in tone and contrast. Without introducing chromatic artefacts, colour-safe correction is proposed to amend the scales.

To summarise, the paper presents several solutions to improve shadow removal quality, and these have been quantitatively verified using robust error measurement and the standard data set in this area [1].

2. User-Assisted Image Shadow Removal

In this section, our algorithm is first described in brief before being expanded on with technical details for each of its components. Our algorithm consists of 4 steps (see Fig. 1):

1) Pre-processing (§2.1) A shadow mask is detected (Fig. 1 (b)) using a KNN classifier trained from K-Means clustered data from user inputs (*e.g.* Fig. 1 (a)). A *fusion image* is generated, which provides an illumination-insensitive layer, by fusing the channels of YCrCb colour space and de-noising (Fig. 1 (c)).

2) Intensity sampling (§2.2) Intensity profiles are obtained for sampling lines perpendicular to shadow boundaries. *Poor* samples are filtered based on similarity of illumination change (Fig. 1 (d)) and de-noised using directional bilateral filtering (Fig. 1 (e)).

3) Estimation of shadow scale and relighting (§2.3)

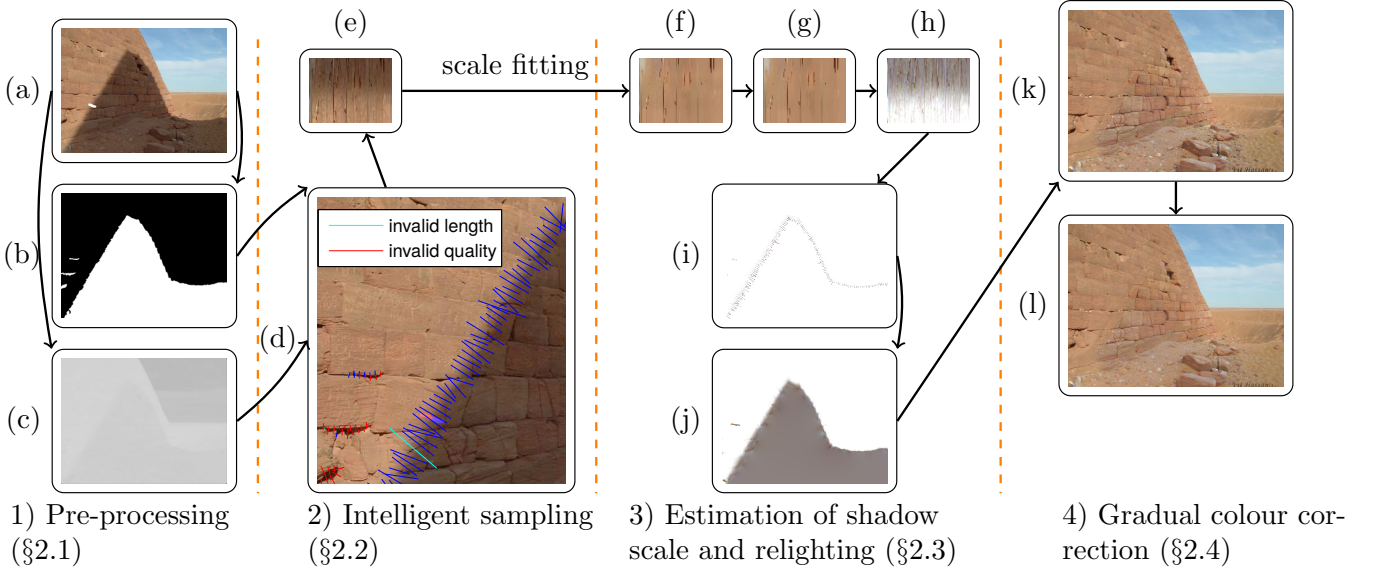


Figure 1: Shadow removal pipeline. (a) input: a shadow image and user strokes in white covering both shadowed and lit pixels; (b) detected shadow mask; (c) fusion image; (d) initial penumbra sampling: the blue lines are valid samples, the other lines are invalid samples. The original single stroke has been divided into lit (red) and shadow (blue) parts; (e) boundary image of samples; (f) re-lit and filtered boundary image; (g) re-lit boundary image after boundary artefact removal; (h) rectified shadow scale of boundary image (each column refers to the scales for a sample); (i) sparse shadow scales; (j) dense shadow scales; (k) initial shadow-free image; (l) colour-corrected shadow-free image.

Given the filtered intensity samples, these are fit through and relit (Fig. 1 (f)) using a piece-wise cubic curve and a boundary image of the samples (Fig. 1 (e)). Any remaining boundary artefacts are removed using directional scale suppression (Fig. 11(g)) over the boundary image. Fitted sparse scales are propagated (Fig. 1 (h-i)) to generate a dense scale field (Fig. 1 (j)). Shadows are then removed (Fig. 1 (k)) by inverse scaling using this dense scale field.

4) Gradual colour correction (§2.4) Any remaining shadow removal artefacts due to image post-processing are finally treated with our colour correction (Fig. 1 (l)). This uses statistics around penumbra boundaries and the shadow scale field.

2.1. Pre-processing

Pre-processing provides a shadow mask and a fusion image to assist intensity sampling (§2.2). Determining the initial shadow mask is the first step of shadow removal and is required in many previous methods including [1, 2, 10, 18, 19]. Although some methods can achieve automatic shadow detection, these results are dependent on,

the quality and variation of training data. In this work, all that is required is the user to supply single strokes covering related shadow and lit pixels (Fig. 1 (a)) – the remaining differentiation and recognition is fully automatic. Under many circumstances, our interaction is easy to perform as it does not require users to explicitly distinguish between shadow and lit pixels. The pixels covered by the single user stroke are first classified as either shadow or lit pixels using K-Means clustering [20]. K-means is applied for this task because it is unsupervised and no training samples for differentiating shadow and lit pixels are provided by a user. The feature used for clustering is the normalised RGB intensity and the normalised pixels coordinates. The cluster with the lowest mean for its RGB intensity is considered as a shadow cluster and vice versa. The classified input pixels' RGB intensities are used as the training features to construct a KNN classifier [21] (number of neighbours: 3). Euclidean distance is used as the distance measure and the majority rule with nearest point tie-break as the classification measure. KNN is applied for this task as it is a supervised algorithm that divides data entries into 2

clusters. The input image can be binarised as a shadow mask, *e.g.* Fig. 1 (b), using the pixel-wise KNN classifier.

We adopt a fusion image [18] for assisting intensity sample collection. The fusion image provides an illumination-insensitive layer, *e.g.* Fig. 1 (c). It can be obtained by linearly fusing the channels of YCrCb colour space. The fused image \mathcal{F}_p is computed as a weighted sum of the 3 normalised channels \mathcal{C}_l as follows:

$$\mathcal{F}_p = \left(\sum_{l=1}^3 \mathcal{C}_l \varphi(\sigma_l) \right) / \left(\sum_{l=1}^3 \varphi(\sigma_l) \right) \quad (2)$$

where l is the channel index, σ_l is the standard derivation of the umbra sample intensities of \mathcal{C}_l . φ is an exponential incentive function for determining the weight for each channel.

$$\varphi(x) = x^{-5} \quad (3)$$

where x is the pixel intensity. Lower variation of intensity is preferred as it means higher intensity uniformity in the umbra segment. To suppress texture noise, a median filter [22] with size h_1 (default: 10) filter window is further applied to \mathcal{F}_p .

2.2. Intensity Sampling

Our intensity sampling rejects inferior intensity samples for robust shadow scale estimation. There are 3 steps: **1) Adaptive raw intensity sampling** RGB intensity profiles are extracted along sampling lines perpendicular to the shadow boundary, *e.g.* Fig. 1 (d), where the boundary is obtained from the shadow mask. To accelerate shadow scale estimation, sampling lines are not measured at each shadow boundary point. Sparse and fixed distance sampling intervals are also avoided, as this may cause artefacts at highly uneven boundary segments [2, 10]. Instead, smaller sampling intervals are adaptively assigned at segments along the shadow boundary based on curvature. Our intention is to sample more intensity samples for curve shadow boundary segments. We compute a curvature array C which stores the curvatures of all shadow boundary points. A sampling mark is set for shadow boundary point

when its normalised absolute curvature is greater than a threshold h_2 (default 0.05) as shown in the first case of Eq. 4.

$$D_m = \begin{cases} 1, & \frac{|C_m|}{\sum_{i=1}^N |C_i|} > h_2 \\ 1, & m = 1 \quad \text{or} \quad m = N \\ 1, & \sum_{i=1}^4 D_{m-i} = 0 \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

where N is the number of boundary points, m specifies the index of boundary points, D is the sampling mark array. The m^{th} shadow boundary point is set for sampling if $D_m = 1$. To provide enough intensity samples for image borders, the first and last boundary points are also set for sampling (second case in Eq. 4). If the boundary is nearly straight, the sampling interval for that segment is fixed to a maxima h_3 (default: 5) as shown in the third case of Eq. 4. As shown in Fig. 2, our variable sampling interval avoids penumbra removal artefacts around sharp boundary segments.

To adapt the variance of penumbra softness, the length of a sampling line is guided by the fusion image. This problem is equivalent to finding the locations of the two ends of a sampling line. A bi-directional search [18] is applied from each boundary point towards the lit area (end point) and the shadow area (start point) as described in Algorithm 1. The start and end points are initially set as the boundary point \mathbf{p}_b . To get the position for a start point, the boundary normal \mathbf{n}_b is iteratively subtracted from the start point (vice versa for the end point) until the average of two ends' projected gradient strength (\mathcal{L}_s and \mathcal{L}_e) is small enough (controlled by an attenuation factor h_4 with a default value 5) or either of the ends is outside the range of image coordinates.

2) Intensity sample selection Outlier intensity samples, *e.g.* Fig. 1 (d), can affect accurate shadow scale estimation and cause unnatural shadow removal results. Two criteria are adopted for outlier detection: a) Length of

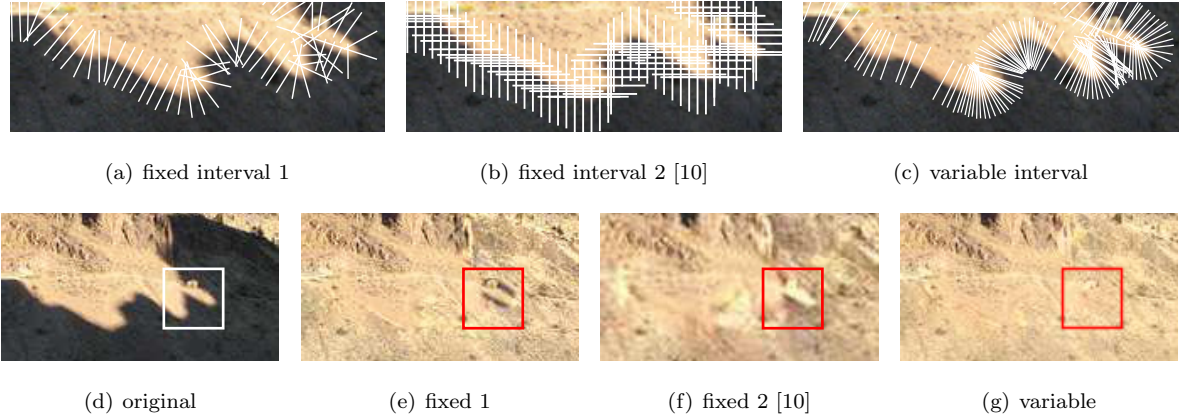


Figure 2: Comparison of sampling scheme. The white lines in (a), (b), and (c) are the sampling lines of the fixed interval using boundary-perpendicular sampling, the fixed interval using horizontal/vertical sampling in [10], and our boundary-perpendicular variable interval sampling respectively. (d) is the original image. (e), (f), and (g) are the corresponding shadow removal results of the three sampling methods respectively.

Algorithm 1: Sample End-Point Selection

input : boundary point \mathbf{p}_b , boundary normal \mathbf{n}_b ,
fusion image \mathcal{F}_p

output: two ends (\mathbf{p}_s , \mathbf{p}_e) of a sampling line

$\tilde{F} \leftarrow \nabla \mathcal{F}_p$; $\mathcal{L} \leftarrow \tilde{F}(\mathbf{p}_b) \cdot \mathbf{n}_b$;

$\mathbf{p}_s \leftarrow (\mathbf{p}_b)$; $\mathbf{p}_e \leftarrow (\mathbf{p}_b)$;

repeat

$\mathbf{v}_s \leftarrow \tilde{F}([\mathbf{p}_s])$; $\mathbf{v}_e \leftarrow \tilde{F}([\mathbf{p}_e])$;

$\mathcal{L}_s \leftarrow \mathbf{v}_s \cdot \mathbf{n}_b$; $\mathcal{L}_e \leftarrow \mathbf{v}_e \cdot \mathbf{n}_b$;

$\mathbf{p}_s \leftarrow \mathbf{p}_s - \mathbf{n}_b$; $\mathbf{p}_e \leftarrow \mathbf{p}_e + \mathbf{n}_b$;

until either \mathbf{p}_s or \mathbf{p}_e is outside \mathcal{F}_p or

$h_4(\mathcal{L}_s + \mathcal{L}_e) < \mathcal{L}$;

sampling line. The minimum length of a sampling line is 3 and the maximum length is $l_\mu + 3l_\sigma$ where l_μ and l_σ are the mean and the standard derivation of sample length respectively. The samples whose lengths are out of this range are removed; b) Similarity of illumination changes. Outliers of intensity samples are often caused by surface texture and such samples can affect illumination change estimation. Fig. 3 shows a synthetic example of various intensity samples which better explains our motivation. A rough RGB intensity profile is obtained by down-sampling each intensity sample to 3 pixels' long using discrete cosine transform (DCT) [23]. This short intensity profile is useful to cancel out texture noise. To

obtain the features of illumination change, this short intensity profile is converted to the Log-domain because the multiplicative shadow scales (Eq. 1) become additive in the Log-domain. The approximate derivatives for each channel of each Log-domain intensity sample are supplied as illumination change features. This results in a 6-D feature vector for each intensity sample (two derivatives for each RGB channel). We assume that the majority of illumination change features are close to each other and can form a major cluster. The objective here is to find out the major cluster and discard the other minor clusters. A density-based DBSCAN clustering method [24] (radius h_5 : 0.2) is used to categorise the samples. The samples belonging to the largest cluster are identified as the samples with valid illumination change and the remaining invalid samples are discarded. DBSCAN is chosen for this task as it does not constrain the number of clusters which is suitable for the fact that many outliers are often located in uncertain number of smaller clusters distant to each other.

3) Sample de-noising Texture noise can still affect the smoothness of shadow scale estimation even when the outliers are removed. Texture noise is removed from the selected intensity samples using a directional (*i.e.* parallel to normals of the shadow boundary) bilateral filter [25]. To achieve this, the raw intensity samples are first re-

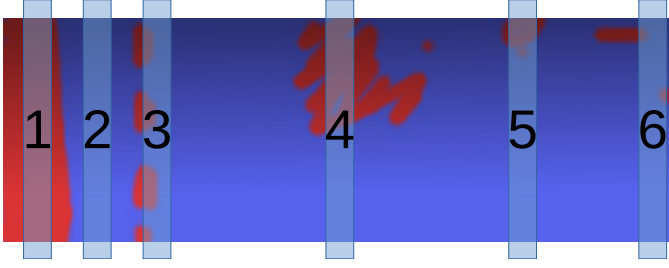


Figure 3: Intensity sample selection example. The 6 marked transparent blue lines are intensity sampling lines. Samples 1, 2, 5, 6 are classified as valid and the rest are invalid. Although the surface materials of sample 1 and 2 are different, their illumination change vectors are exactly the same and they both reflect true illumination change. This is because for each of them, the surface material in the shadow and lit areas are constant. Samples 3 and 4 are classified as outliers because the surface materials in the shadow and lit areas are varying. Sample 5 and 6 contain some varying surface materials, but the variations of the surface material are not significant. Therefore, they are still classified as valid samples.

sized as individual columns and their lengths are set as the maximum length of all raw samples. These columns are concatenated horizontally to form a boundary image, *e.g.* Fig. 1 (e). A bilateral filter [25] is applied to each RGB channel of this image to suppress texture noise. A Bilateral filter is chosen for this task as it removes texture noise and preserves edge and smooth intensity change. 295

2.3. Estimation of Shadow Scale and Relighting

This sub-section explains the procedure for removing shadows based on the processed intensity samples. The following description of the algorithm is applied to the 300 samples of each detected shadow boundary. There are 3 steps:

1) Initial intensity fitting Having obtained filtered and resized intensity samples at different positions along the boundary, our goal is to find illumination scaling values 305 inside the umbra, penumbra and lit area. The shadow scale change function S for each RGB channel of each intensity sample is modelled as follows (see also Fig. 4):

$$S(x) = \begin{cases} K & 0 \leq x \leq x_1 \\ f(x) & x_1 < x \leq x_2 \\ 1 & x_2 < x \leq 1 \end{cases} \quad (5) \quad \text{310}$$

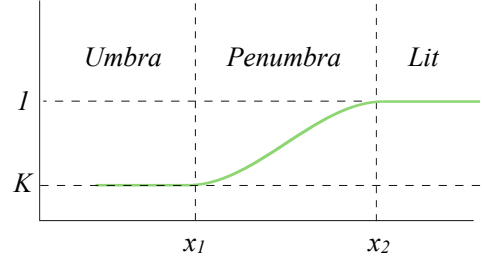


Figure 4: Shadow scale model: x_1 and x_2 define the ends of a penumbra region. K is a scale of umbra intensity scale.

where x is a normalised 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 positive scale constant for sample points within the umbra area ($x < x_1$). The constant 1 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. The function f is parametrised by K , v_1 and v_2 as follows:

$$\begin{cases} f(x) = (1 - K)B(v_1(x - v_2)) + K \\ B(y) = -2y^3 + 1.5y - 0.5 \\ \begin{bmatrix} x_1 & x_2 \end{bmatrix} = v_2 + v_1^{-1} \begin{bmatrix} -0.5 & 0.5 \end{bmatrix} \end{cases} \quad (6)$$

where B is a cubic shape function (a sinusoidal function here also produces adequate results) and y is the input, v_1 and v_2 are two parameters used to define the shape of the illumination change function and are linearly related to x_1 and x_2 . Illumination of each channel usually varies differently (*e.g.* outdoor shadows appear bluish) and therefore 3 independent K (in Eq. 5) are estimated for each channel. B is a smooth curve with two zero 1st order derivatives at both ends, which serves as a rough fit to the intensity change and helps to locate the penumbra location. Penumbra boundaries of the 3 channels are usually the same. Consequently, a common penumbra width and position is assumed for each channel, determined by v_1 and v_2 respectively. This assumption of channel-invariant penumbra location can make the optimisation more robust. This is because the intensity sample of one channel can be very noisy but the others' are not. In summary, for each sampling line, there are 5 parameters to solve in total. The parameters can be solved by least squares fitting with

a sequential quadratic programming algorithm [26]. The fitted parameters of all sampling lines are also smoothed by using a robust smoothing method (maximum iteration: 100, weighting function: bi-square) [27].

2) Boundary artefact removal For some over-processed images, directly relighting the sparse samples using the fitted shadow scales may cause band-like artefacts in the penumbra. Directional scale suppression (Fig. 1 (f-h)) is therefore applied. Fig. 5 shows an example of more obvious boundary artefact removal. The band artefacts appear as a

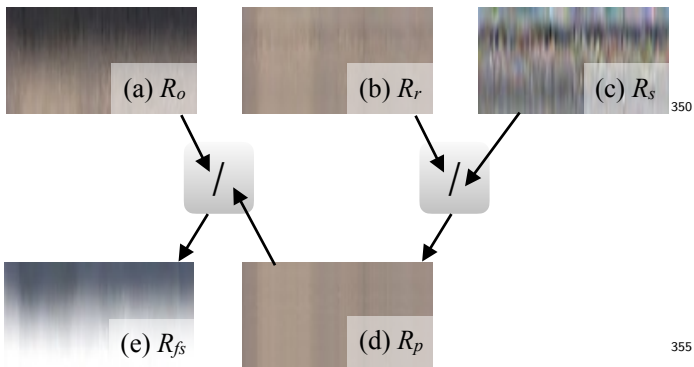


Figure 5: Boundary artefact removal pipeline. (a) The boundary frame image R_o . (b) Initial re-lit image R_r . (c) The band-like artefact image R_s extracted from R_r . R_s is visualised in the log domain. (d) Band-like artefact removed frame image R_p . (e) Amended shadow-scale frame image R_{fs} .

pattern in each re-lit intensity profile (Fig. 1 (b)). To suppress these, the previously re-sized and filtered intensity profile is first aligned according to the estimated penumbra position (x_1 and x_2) so that the illumination change of each resized sample is synchronised. The concatenated boundary image is denoted as R_o . The boundary image R_o is re-lit by inverse scaling according to Eq. 1 and the resulting image is termed R_r . It is assumed that band-like artefacts are locally similar and a local group size $h_6 = 5$ is specified for suppression. To extract the local band pattern, horizontal filtering is applied to R_r using an average kernel (size: 5x1) and the filtered image denoted as R_f . To suppress the band-like artefact, the variance of each column's, *i.e.* each samples, intensities of R_f are minimised. To achieve this, a variance image R_s is computed by di-

viding each column's intensities of R_f by its corresponding average intensity of that column. The artefact-free and re-lit boundary image is computed as $R_p = R_r \oslash R_s$ where \oslash is an operator for element-wise division. Finally, the rectified scale image is computed as $R_{fs} = R_o \oslash R_p$. The sparse scales for each sampling lines corresponds to each column of R_{fs} .

3) Relighting To obtain a dense scale field (e.g. Fig. 1 (j)), the sparse scales in the penumbra region are processed by smoothly interpolating and extrapolating the scales in other regions by using a general image in-painting algorithm [28]. Our adaptive intensity sampling (§2.2) provides enough shadow scale samples, in sub-pixel accuracy, for recovering dense shadow scales in the curve shadow boundaries by using in-painting. The shadow-free image (e.g. Fig. 1 (k)) can be obtained by inverse scaling according to Eq. 1. As the dense shadow field is formed by propagating the shadow scales in penumbra, a wrong shadow scale estimate of a sampling line can produce wrong dense scales for a wider region. Our previous efforts in filtering bad intensity samples are thus crucial for this step.

2.4. Gradual Colour Correction

Some input images may have been significantly post-processed, *e.g.* through JPEG compression or gamma correction. Highly visible artefacts, *e.g.* differences in tone and contrast, may appear in shadow corrected areas as Eq. 1 does not hold in such cases. To address this, a simple gradual colour correction is introduced which is generally compatible for unknown post-processing affects. This step is only necessary for over post-processed images and the difference may otherwise be insignificant for the other images. The shadow removed image is first converted to $L^*a^*b^*$ colour space because $L^*a^*b^*$ colour space is designed for visual perception adjustment [29]. It is assumed that the $L^*a^*b^*$ intensity variation of lower frequency is accurate and the errors appear in the intensity variation of higher frequencies. Fig. 6 shows the intermediate steps

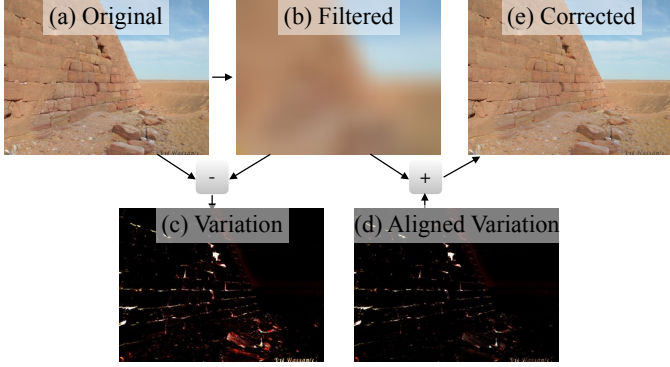


Figure 6: Gradual colour correction pipeline. The initial shadow-removed image (a) exhibits inconsistency between originally-lit and shadow-removed areas. This image is filtered (b) and used to extract an image of higher-frequency variation (c). The variation of shadow and lit areas are aligned. In (d), the original higher variation in the shadow area is made consistent (i.e. suppressed) with the lit areas'. The colour-corrected image is computed by adding (b) and (d). The variation images (c, d) are displayed by applying a gamma 3 to magnify the difference.

of colour correction in corresponding to the result in Fig. 1 (l). Statistics are collected from the lit side pixels \mathbf{P}_l and the umbra side pixels \mathbf{P}_u both near penumbra as the target and source of colour correction respectively. \mathbf{P}_s defines the set of all shadow pixels which include both umbra and penumbra pixels, i.e., $\mathbf{P}_u \subset \mathbf{P}_s$. In $L^*a^*b^*$ colour space, the image of higher frequency intensity variation $I^h = I^r - I^l$ is computed where I^l is the initial shadow removed image I^r filtered by a bilateral filter [25]. The adjustment is completed in $L^*a^*b^*$ colour space as described in Eq. 7.

$$\begin{cases} r_\sigma = \varsigma(I_c^h(\mathbf{P}_l)) / \varsigma(I_c^h(\mathbf{P}_u)) \\ I_c^{ra}(\mathbf{P}_s) = I_c^l(\mathbf{P}_s) + r_\sigma I_c^h(\mathbf{P}_s) \end{cases} \quad (7)$$

where c is the channel index, I^{ra} is the colour corrected image and the intensities of other unmodified pixel of I^{ra} are identical to those of I^r , \mathbf{P}_s is the set of all shadowed pixels, ς is a function computes the median absolute deviation.

Since our colour correction is only applied to the entire shadow segment, some minor intensity discontinuities may appear in the shadow boundary after colour correction. To smooth the colour correction result, an alpha blending is

applied in RGB colour space according to the shadow scale as the follows

$$I_c^f = I_c^r \circ \dot{S}_c + I_c^{ra} \circ (1 - \dot{S}_c) \quad (8)$$

$$\dot{S}_c(x, y) = \max\left(\frac{\mathcal{S}_c(x, y) - S_c^{5\%}}{1 - S_c^{5\%}}, 0\right) \quad (9)$$

where c is the channel index, x and y are the image coordinates, \mathcal{S}^w is the normalised scale field of \mathcal{S} , $S_c^{5\%}$ is the 5% percentile of the values in \dot{S}_c , I_c^f is the final shadow-free image. The 5% percentile value is used for shadow scale normalisation instead of the minimum value because sometimes the minimum value can be an outlier distant to the main cluster of a shadow scale distribution. The maximum operation ensures that the normalised shadow scale values are always non-negative. An example of this alpha-blending effect is shown in Fig. 7.

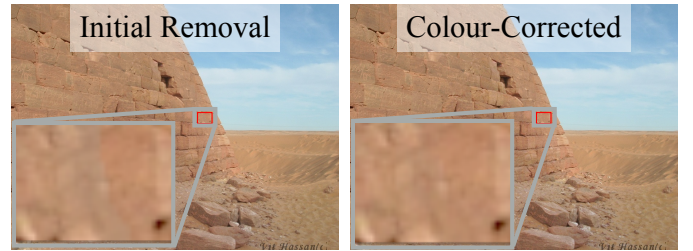


Figure 7: The colour-corrected image is blended with the initial shadow removed image. The resulting blend avoids intensity discontinuity at the shadow boundary. The zoom-in patches are marked by two grey boxes above. A visible crack-like boundary discontinuity is found in the middle of the left patch.

2.5. Customisable Parameters

In Tab. 1, we summarise the important parameters which users can optionally specify. Although fine tuning these parameters (for an individual case) may improve the shadow removal result, in practice, we only adopt a set of default parameters for general shadow removal, which are then used for our following evaluation.

3. Evaluation

In this section, we first show results of tests highlighting algorithm behaviour given variable user inputs. Our

ID	Description	Value
h_1	Filter window size of medium filter to suppress texture noise. (§2.1)	10
h_2	Sampling curvature interval for placing sampling lines (§2.1)	0.05
h_3	Maximum boundary point interval for placing sampling lines (§2.1)	5
h_4	An gradient attenuation factor for locating sampling line ends (§2.1)	5
h_5	DBScan Radius for filtering bad intensity samples (§2.1)	0.2
h_6	Default local group size of sampling lines (§2.3)	5

Table 1: Important parameters and their default values
 algorithm is then evaluated versus other state-of-the-art
 shadow removal methods using both visual comparisons
 and our improved quantitative evaluation measurements.
 This includes an additional algorithm for ground truth
 rectification on the current state-of-the-art open dataset [1].

3.1. Variability under Different User Inputs

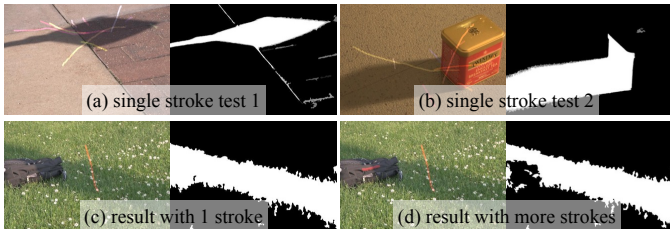


Figure 8: Variable input behaviours: The top row shows two examples using single input strokes. We supply 10 examples of single strokes placed in different locations and used as input (shown in different colours). The 2 grey-level images show the visualised probability of each pixel being marked in these 10 independent tests. Fewer grey pixels indicate higher stability, *i.e.* the image should only show black (0% probability) and white (100% probability) pixels when it is absolute stable. The bottom row shows examples highlighting how additional strokes can improve the detection result (binary mask).

Given user-specified single strokes, our shadow detection generates stable results in different conditions (e.g. Fig. 8 (a) and Fig. 8 (b)). When the shaded surfaces are made of different materials and a single stroke cannot cover all of them, multiple strokes may be needed. Often, more strokes can lead to more robust detection result (e.g. Fig. 8

(c) and Fig. 8 (d)).

3.2. Rectification of Ground Truth

In the dataset of Guo [1], many of the shadow-free ground truth images are collected by entirely blocking the natural light in the scene. This unfortunately causes inconsistency in the brightness between some shadow-test images (e.g. Fig. 9 (a)) and corresponding shadow-free ground truth (e.g. Fig. 9 (b)). This will result in unfaithful quantitative evaluations in some test cases. To compensate for this, ground truth images of this kind can be globally re-lit (e.g. Fig. 9 (c)) before evaluation. The RGB scale vector for global relighting can be estimated from the average RGB intensity of the common lit area. Lit pixels are first detected using a ratio image $I_{gr} = I \oslash I_{gt}$ where I is the original shadow image, I_{gt} is the shadow-free ground truth, \oslash is an operator for element-wise division. K-Means clustering [20] is then used to divide the ratios into two clusters and the cluster with higher average ratios are identified as the lit cluster.



Figure 9: Ground truth adjustment: An example rectified shadow-free ground truth image (c) obtained by correcting (b) shows a higher consistency with the test image (a). Note that in the original ground truth, the corrected image shows dark pixels as opposed to expected light ones (corrected in our rectified example).

3.3. Quantitative Evaluation

In previous work [1], the quality of shadow removal is measured by the per-pixel Root-Mean-Square-Error (RMSE) between the shadow removal result and shadow-free ground truth in RGB colour space. However, the size and darkness of a shadow are often variable and this can result in biased shadow removal quality measurements. For example, an unprocessed image with a small area of shadow can have a smaller RMSE than the error of an image which has a

425 large area of shadow that has only been partially corrected. Therefore, images with larger or darker shadows can affect the overall score. Our error ratio is therefore computed as $\mathbf{E}_r = \mathbf{E}_n/\mathbf{E}_o$ where \mathbf{E}_n is the RMSE between the shadow-465 free ground truth and shadow removal result, and \mathbf{E}_o is the RMSE between the shadow-free ground truth and the original shadow image. All error measurements are assessed in RGB colour space. This normalised measure reflects the degree of shadow removal towards the ground truth independent of original shadow intensity and size. To assess 430 the robustness, the standard derivation is also computed for each measurement. 470

Extending previous work on ground truth based evaluation [1], we categorise shadows into different types. In our test, additional scores for particular categories of shadows 440 with soft penumbra and strong texture background are shown. These special categories are included because they are generally more difficult to process compared with low texture backgrounds with compact shadows. Quantitative results are presented in Tab. 2, where *starred* columns refer 445 to pixels just in the shadow region being considered, and *non-starred* columns refer to the entire image. Our method outperforms the other approaches compared against for most of the scores (especially for soft shadow tests). There are a small number of our scores that are numerically close 450 to the second best ones. However, small numerical differences may indicate visually significant artefacts which are shown in the visual comparison sub-section (§3.4). 485

3.4. Visual Comparisons

Typical examples of our shadow removal algorithm are 455 shown for visual comparison in Fig. 10. Overall, our method produces more qualitatively pleasing removal results against 490 the evaluated methods specially for shadow boundary recovery. However, minor artefacts are sometimes noticeable when the input image has a highly irregular soft penumbra, or the background of the shadow area is highly shadow-495 like. Fig. 11 shows some difficult cases where shadows

are soft and broken or cast on a shadow-like background. These typical examples of limitation cases identified in all tested shadow removal methods represent future research challenges in our field.

4. Conclusion

A user-friendly shadow removal method has been presented that provides several innovations in this area of research. This includes simple user input, intelligent intensity sampling, a local group processing based shadow scale estimation and robust colour correction. The presented algorithm has been quantitatively evaluated using the standard dataset in this area, and demonstrates state-of-the-art performance. Visual comparisons are also presented for a large number of shadow removal cases taken from the evaluation data set. Through our analysis, difficult shadow removal cases such as broken and soft shadows, and shadows on strong texture background have also been identified. These represent exciting research challenges in our area.

Acknowledgement

This work was supported by Engineering and Physical Sciences Research Council (EPSRC) (EP/M023281/1, EP/M001768/1).

References

- [1] R. Guo, Q. Dai, D. Hoiem, Paired regions for shadow detection and removal, *Transaction on Pattern Analysis and Machine Intelligence* PP (99) (2012) 1–1.
- [2] E. Arbel, H. Hel-Or, Shadow removal using intensity surfaces and texture anchor points, *Transaction Pattern Analysis and Machine Intelligence* 33 (6) (2011) 1202–1216.
- [3] G. D. Finlayson, M. S. Drew, C. Lu, Entropy minimization for shadow removal, *International Journal on Computer Vision* 85 (1) (2009) 35–57.
- [4] Q. Yang, K.-H. Tan, N. Ahuja, Shadow removal using bilateral filtering, *Transaction on Image Processing* 21 (10) (2012) 4361–4368.

Guo <i>et al.</i> [1]		Su <i>et al.</i> [11]		Zhang <i>et al.</i> [19]		Gong <i>et al.</i> [18]		Ours	
E	E*	E	E*	E	E*	E	E*	E	E*
Softness									
0.79 (0.41)	0.63 (0.38)	0.63 (0.31)	0.69 (0.67)	0.53 (0.28)	0.33 (0.25)	0.61 (0.27)	0.44 (0.27)	0.49 (0.27)	0.32 (0.27)
Textureness									
0.72 (0.41)	0.46 (0.36)	0.77 (0.35)	0.61 (0.42)	0.65 (0.29)	0.45 (0.59)	0.62 (0.26)	0.42 (0.25)	0.63 (0.28)	0.39 (0.23)
Overall									
0.67 (0.35)	0.48 (0.36)	0.78 (0.43)	0.97 (1.14)	0.67 (0.37)	0.46 (0.51)	0.60 (0.26)	0.42 (0.33)	0.59 (0.28)	0.40 (0.31)

Table 2: Categorised quantitative test results. The non-stared and stared columns indicate the error score where all pixels in the image are used, and just shadow area pixels respectively. The categories of “Soft” and “Texture” refer to shadow images with soft penumbra and strong texture background respectively. “Overall” is the average result for all test cases in the data set. Method [1] is trained using a large shadow detection data set from [12]. The standard derivation for each measurement are in brackets.

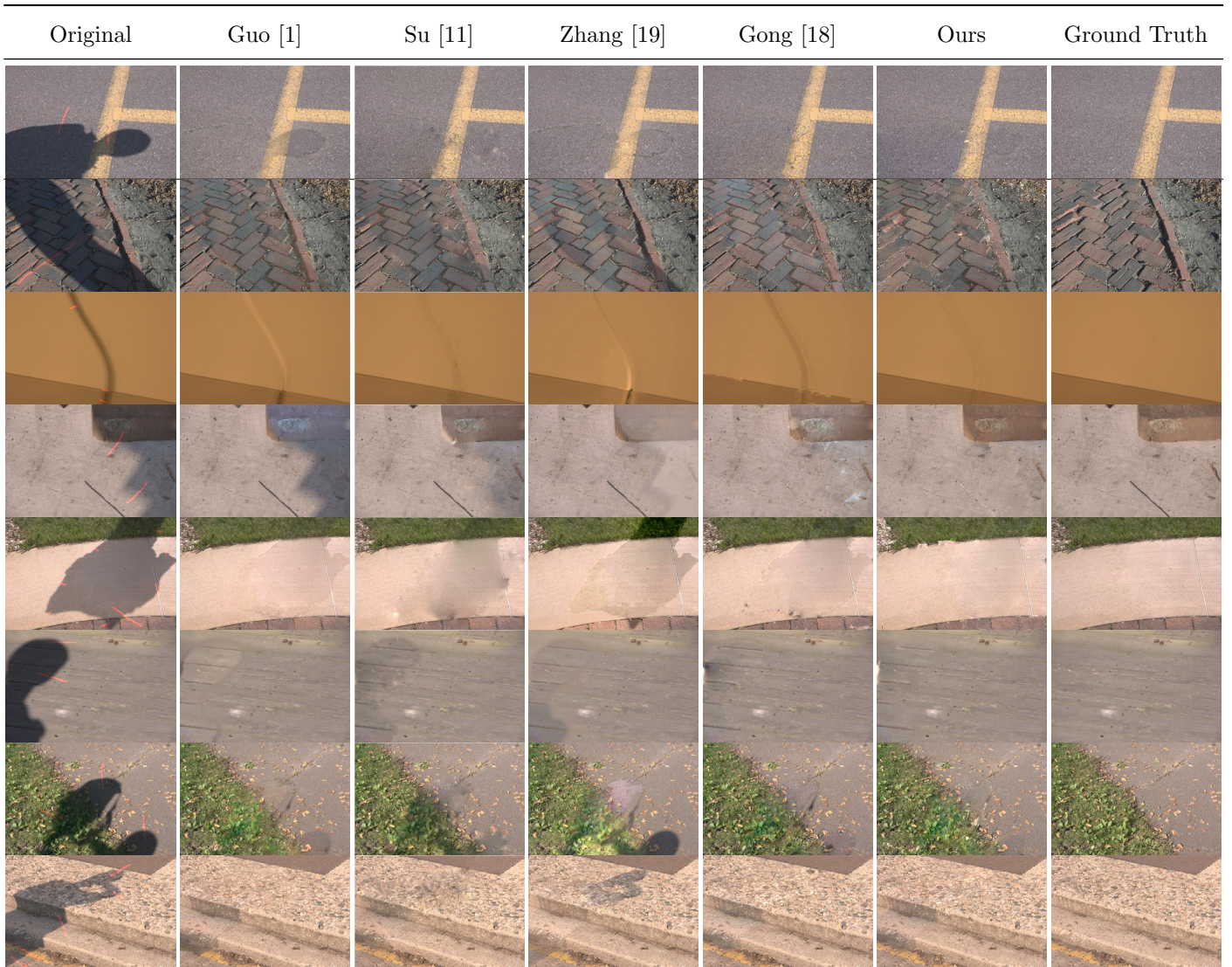


Figure 10: Comparisons of our method against existing state-of-the-art techniques using ground truth images from [1]. Please magnify to examine penumbra recovery in detail. The first column images are the original shadow images with our user input strokes marked in red.

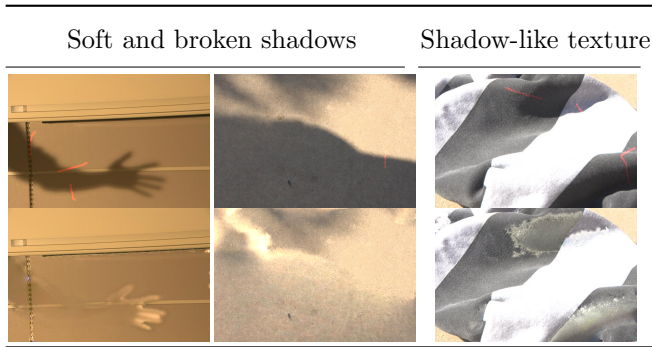


Figure 11: Universal failure cases – where all the leading methods create highly visible image artefacts and fail to remove the shadow.

- [5] C. J. J. Yoon, T.J.Ellis, Shadowflash: an approach for shadow removal in an active illumination environment, in: British Machine Vision Conference, BMVA, 2002.
- [6] M. Drew, C. Lu, G. Finlayson, Removing shadows using flash/noflash image edges, in: International Conference on Multimedia and Expo, IEEE, 2006, pp. 257–260.
- [7] N. Salamaty, A. Germain, S. Susstrunk, Removing shadows from images using color and near-infrared, in: International Conference on Image Processing, IEEE, 2011, pp. 1713 –1716.
- [8] G. Finlayson, C. Fredembach, M. Drew, Detecting illumination in images, in: International Conference on Computer Vision, IEEE, 2007, pp. 1–8.
- [9] T.-P. Wu, C.-K. Tang, M. S. Brown, H.-Y. Shum, Natural shadow matting, *Transactions on Graphics* 26 (2) (2007) 8.
- [10] F. Liu, M. Gleicher, Texture-consistent shadow removal, in: European Conference on Computer Vision, Springer, 2008, pp. 437–450.
- [11] Y.-F. Su, H. H. Chen, A three-stage approach to shadow field estimation from partial boundary information, *Transaction on Image Processing* 19 (10) (2010) 2749–2760.
- [12] J. Zhu, K. G. G. Samuel, S. Masood, M. F. Tappen, Learning to recognize shadows in monochromatic natural images, in: International Conference on Computer Vision and Pattern Analysis, IEEE, 2010, pp. 223–230.
- [13] J.-F. Lalonde, A. A. Efros, S. G. Narasimhan, Detecting ground shadows in outdoor consumer photographs, in: European Conference on Computer Vision, 2010, pp. 322–335.
- [14] J.-B. Huang, C.-S. Chen, Moving cast shadow detection using physics-based features, in: International Conference on Computer Vision and Pattern Analysis, IEEE, 2009, pp. 2310 –2317.
- [15] J. Yao, Z. M. Zhang, Hierarchical shadow detection for color aerial images, *Computer Vision and Image Understanding* 102 (1) (2006) 60–69.
- [16] E. Salvador, A. Cavallaro, T. Ebrahimi, Cast shadow segmentation using invariant color features, *Computer vision and image understanding* 95 (2) (2004) 238–259.
- [17] Y. Shor, D. Lischinski, The shadow meets the mask: Pyramid-based shadow removal, *Computer Graphics Forum* 27 (2) (2008) 577–586.
- [18] H. Gong, D. Cosker, Interactive shadow removal and ground truth for variable scene categories, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2014.
- [19] L. Zhang, Q. Zhang, C. Xiao, Shadow remover: Image shadow removal based on illumination recovering optimization, *Transactions on Image Processing* 24 (11) (2015) 4623–4636.
- [20] G. A. Seber, *Multivariate observations*, Vol. 252, John Wiley & Sons, 2009.
- [21] J. H. Friedman, J. L. Bentley, R. A. Finkel, An algorithm for finding best matches in logarithmic expected time, *ACM Transactions on Mathematical Software* 3 (3) (1977) 209–226.
- [22] E. Cliffs, *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.
- [23] S. Martucci, et al., Image resizing in the discrete cosine transform domain, in: *Image Processing, 1995. Proceedings., International Conference on*, Vol. 2, IEEE, 1995, pp. 244–247.
- [24] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise., in: *SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. 96, ACM, 1996, pp. 226–231.
- [25] S. Paris, F. Durand, A fast approximation of the bilateral filter using a signal processing approach, *International Journal of Computer Vision* 81 (1) (2009) 24–52.
- [26] J. Nocedal, S. Wright, *Numerical optimization*, 2nd Edition, Springer series in operations research, Springer, 2006, Ch. 18.
- [27] D. Garcia, Robust smoothing of gridded data in one and higher dimensions with missing values, *Computational Statistics & Data Analysis* 54 (4) (2010) 1167–1178.
- [28] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: *Transaction on Graphics*, ACM, 2000, pp. 417–424.
- [29] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley, Color transfer between images, *IEEE Computer graphics and applications* 21 (5) (2001) 34–41.