

# Texture-Preserving Shadow Removal in Color Images Containing Curved Surfaces

Eli Arbel

Department of Computer Science  
University of Haifa  
Israel

Hagit Hel-Or

Department of Computer Science  
University of Haifa  
Israel

## Abstract

Several approaches to shadow removal in color images have been introduced in recent years. Yet these methods fail in removing shadows that are cast on curved surfaces, as well as retaining the original texture of the image in shadow boundaries, known as penumbra regions. In this paper, we propose a novel approach which effectively removes shadows from curved surfaces while retaining the textural information in the penumbra, yielding high quality shadow-free images. Our approach aims at finding scale factors to cancel the effect of shadows, including penumbra regions where illumination changes gradually. Due to the fact that surface geometry is also taken into account when computing the scale factors, our method can handle a wider range of shadow images than current state-of-the-art methods, as demonstrated by several examples.

## 1. Introduction

Shadows are an integral part of many natural images. While shadows, and in particular cast shadows, can provide valuable information on an acquired scene, e.g. cues for spatial layout and surface geometry [12], they can also pose difficult problems and limitations for various computer vision algorithms. For example, darker regions in the image caused by cast shadows can introduce spurious segments in segmentation algorithms. In digital photography, shadows may be considered as artifacts in the image that need to be removed, whether for enhancing certain parts of the image or simply for esthetic reasons.

Following common notation, an image  $\mathcal{I}(x, y)$  is considered to be related to the reflectance field  $\mathcal{R}(x, y)$  and the illumination field  $\mathcal{L}(x, y)$  [1] as follows:

$$\mathcal{I}_k(x, y) = \mathcal{R}_k(x, y) \cdot \mathcal{L}_k(x, y) \quad (1)$$

where  $k \in \{R, G, B\}$  and  $\cdot$  denotes pixel-wise multiplication.

Shadow regions are assumed to be formed by reduction in the illumination field, resulting in changes of image intensities by multiplicative scalars  $\mathcal{C}_k(x, y)$ :

$$\mathcal{I}_k(x, y) = \mathcal{R}_k(x, y) \cdot \mathcal{L}_k(x, y) \cdot \mathcal{C}_k(x, y) \quad (2)$$

Taking the logarithm of both sides of equation (2) we obtain:

$$I_k(x, y) = R_k(x, y) + L_k(x, y) + C_k(x, y) \quad (3)$$

such that  $I$ ,  $R$ ,  $L$  and  $C$  are the logarithms of  $\mathcal{I}$ ,  $\mathcal{R}$ ,  $\mathcal{L}$  and  $\mathcal{C}$ , respectively. Thus in the log domain, a shadow implies an *additive* change in intensities.

Shadow regions are assumed to be uniform, in the sense that a single scaling factor is sufficient to capture the intensity changes in a shadow region. In practice, this is not the case, since shadow boundaries are usually diffused regions that are partially lit and partially shadowed, hence implying different scaling constants across the boundary.

A classical approach to shadow removal in a single image is to identify the shadow edges, zero the derivatives of those pixels and then integrate to obtain a shadow-free image. Alternatively, shadow regions can be removed by adding a constant factor in the log domain to the intensities enclosed within the shadow edge. These approaches produce good results when the shadow edges are sharp and the shadow occurs on a flat non-textured surface. However, poor results are obtained when shadows are on curved and textured surfaces (see Figure 1c). This is due to the fact that both textural information and surface gradient information existing at the shadow boundary are removed.

To intuitively illustrate these problems with the classic approach as mentioned above, consider the one-dimensional signals depicted in Figure 1b. The test signal (dark line) is a cross section of the textured and curved surface shown in Figure 1a. Removing the shadow in the test signal using the classic approach of nullifying shadow edges, a shadow-free signal is obtained as depicted in Figure 1d (dark line). It can be seen that the shadowed region

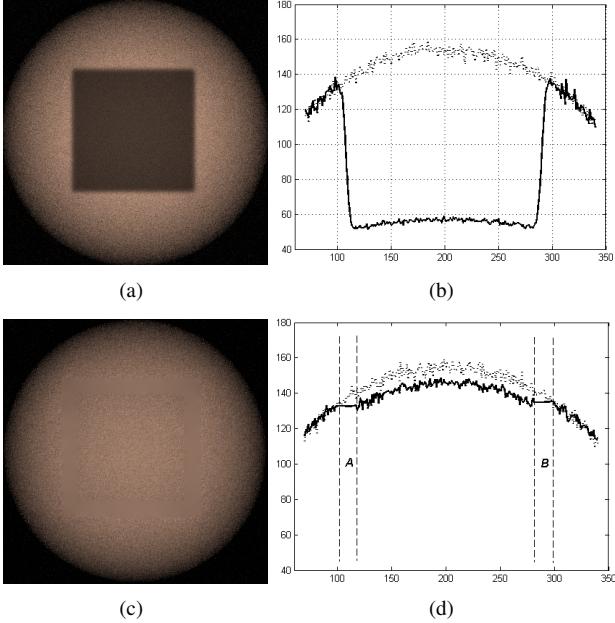


Figure 1. Shadow removal using classical methods. (a) An image of a curved surface with artificial shadow. (b) A horizontal cross section of the shadow image (dark line), compared with a cross section of the original shadow-free image (gray line). (c) Shadow-free image as obtained using classical methods. (d) Cross section of the image in c (dark line), compared with the desired result (gray line).

of the test signal is incorrectly reconstructed: pixel intensities are lower than they should be compared to the original non-shadowed signal (gray line). This is a direct result of the assumption that the surfaces in the image are flat; by setting the shadow edge derivatives to zero, the derivatives of intensity that are due to the curved surface are also nullified. Additionally, it can be seen that due to zeroing of derivatives, the shadow boundary regions (intervals *A* and *B* in Figure 1d) appear almost flat in the reconstructed signal. This indicates the loss of textural information in these regions. In [6], In-Painting is used to reconstruct these pixels, however, this does not solve for the incorrectly reconstructed intensity profile due to the curved surface, and can not deal with local intensity variations that are not repetitive.

In this paper we focus on solving the shadow removal problem in images containing shadows with wide boundaries, known as *penumbrae*, which are cast on non-flat surfaces. Our aim is to produce a shadow-free image while maintaining original local and textural information within shadow regions and penumbrae.

We propose a novel approach for both estimating shadow scale factor for geometrically curved surfaces, and for removing shadows from penumbra regions while retaining the original information in the image, even for wide-penumbra

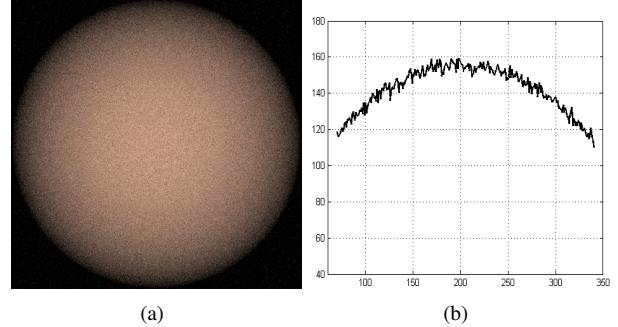


Figure 2. (a) Shadow-free sphere image produced by our method. (b) A horizontal cross section of the shadow-free sphere image.

shadows. Our approach finds scale factors which are used to cancel the effect of shadows, including penumbra regions where illumination changes gradually. Using this scale factor estimation approach, rather than 2D integration, we also ensure that non shadow regions of the image remain untouched (a discussion on the effects of 2D integration can be found in [6]). Figure 2 shows the shadow-free version of the sphere image of Figure 1a, as produced by our method.

## 2. Previous work

Various approaches have been proposed to address the problem of shadow removal in real images. In [18] a sequence of images is used, in which the illumination changes over time but the reflectance remains constant. Derivation filters are applied on each of the images followed by the application of a median filter across the derivated images. Finally, 2D integration of the resulting median image is performed producing the intrinsic reflectance image in which all illumination effects are removed, including shadows. While this method provides good results, it requires a fully stationary camera for capturing the image sequence. Additionally, in order to correctly recover the reflectance image, it is imperative that the images in the sequence exhibit large variations in the illumination conditions. Otherwise, shadows with soft edges may not be completely removed by the median filters, and static shadow regions in the sequence may not be removed at all.

In [5] Finlayson *et al.* proposed a method that, given a single RGB image, removes cast shadows based on the derivation of an illumination invariant image. Edges in this image reflect changes in material reflectance. Thus, it can be used to detect edges in the image that are due to shadows. For each channel's gradient image, shadow edges are thickened by morphological operations and then set to zero. An integration step as in [18] of each channel's gradient image recovers three shadow-free images given up to multiplicative constants which are then estimated in order to obtain the final shadow-free color image.

To solve the problem of flat shadow boundaries in

shadow-free images, caused due to the fact that shadow-boundary gradients are nullified, pixel and edge In-Painting techniques are used in [4, 6] to fill in the missing information in these regions. While this approach can yield a more visibly-pleasant images, the original texture and curvature information in these regions is still lost (see Figure 3 for illustration and comparison to our method<sup>1</sup>). Additionally, instead of performing 2D integration as proposed in [18], a 1D integration process using Hamiltonian paths is used in [6]. Although this integration technique can reduce errors produced by 2D integration, as shown by the authors of [6], the problem of calculating the correct scale factor of the shadow region still remains. The work in [7] suggests



Figure 3. Comparison of various approaches for handling penumbra regions. Top row: shadow images, Middle row: shadow-free images - Nullifying shadow-boundary gradients (left); Shadow edge In-Painting (right), Bottom row: results of our method.

a method for finding the scale factor of a shadow region by sampling pixels along the shadow edge, and estimating the value that minimizes the difference between pixels outside the shadow region and those inside it. This approach is much faster than traditional integration since it only requires the application of a scale factor (or additive constant in the log domain) on the shadow region pixels. However, it still does not account for scenes containing curved surfaces with shadows, since the minimization of differences between the pixels on either side of the shadow boundary assumes that the shadowed surface is geometrically flat in the penumbra region. As in [6], the authors used In-Painting

<sup>1</sup>Images in Figure 3 are from [2](left column) and [4] (right column)

to complete missing information at shadow boundaries.

We argue that in most high resolution images the shadow boundary, *i.e.* the penumbra, is noticeably wide thus may significantly affect the estimation of the scale factors as well as producing visible artifacts and incorrectly reproduced texture at the shadow boundaries.

### 3. Our method for shadow removal

Shadow removal is performed in two stages: 1) the detection stage in which shadow regions are detected, typically by determining the shadow boundaries and 2) the reconstruction stage in which the shadow is actually removed and a shadow-free image is produced.

Various techniques for detecting shadow regions in images have been suggested. They can be classified into two groups: 1) algorithms that are based on invariant color models [5, 14] and 2) algorithms that assume some prior knowledge of the scene, such as geometry, color ratios [9] or object properties within the scene.

This paper suggests a novel reconstruction stage for shadow removal, *i.e.* removing the shadows in an image once they have been detected. Any shadow detection algorithm can be used, but since our method is not confined to images with certain illumination conditions such as outdoor scene images, one could choose to use a specific shadow detection algorithm that best suits the illumination conditions in the given image.

In accordance to Equation (3), it is assumed throughout this section, that shadow removal is performed on each channel of the image represented in the log domain, which is then exponentiated back once the shadows are removed.

#### 3.1. Calculating shadow region scale factor

The essence of the proposed algorithm is to evaluate the correct scale factor,  $\bar{C}(x, y)$  (Equation (2)) with which to reconstruct the shadowed region. It is assumed that  $C(x, y)$  is constant within the shadow region. However, this is not the case across the penumbra region at the shadow boundary. Previous studies attempt to overcome this problem by zeroing gradients within a thickened region of the detected shadow boundaries, thus ignoring the penumbra region altogether. However, as shown in the shadow-free signal in figure 1d, the assumption of a zero average difference between the pixels on either side of the penumbra does not hold in the case of shadows on curved surfaces. On unshadowed curved surfaces, pixel intensities change with surface normal producing what we call an *intensity surface*. Thus, zeroing gradients of the shadow penumbra on a curved surface nullifies the shadow edge but also incorrectly nullifies the gradients of the intensity surface that are due to change in surface normal of the object within the scene. The reconstructed image is then incorrect as can be seen in Figure 1c.

In the general case, when the geometry of the objects in the scene, and accordingly the intensity surface, is not constrained, the problem of finding the correct scale factor along with the intensity surface in the penumbra region, is massively ill-posed. To this end, we introduce an a priori assumption that *the intensity surface in the penumbra region should locally act as a thin plate* in the shadow-free image. In practice, we require that the first and second order directional derivatives of penumbra pixels in the shadow-free surface are continuous. Although this seems a strong requirement, we found it suitable for virtually all natural images we examined.

Thin plate models are widely studied in various fields and applications, which include reconstruction of 3D objects [17], deformable surfaces in computer graphics [16] and image morphing [8].

To solve for the scale factor under the continuity constraint of first and second order derivatives, we formulate the shadow removal problem as a surface reconstruction problem, where the missing data is the intensity-surface values of the penumbra pixels. We found the thin plate model a convenient tool for this purpose, since it allows us to control the *stiffness* of the reconstructed surface in the penumbra region, which compromises between fitting the data and smoothness.

Mathematically, we consider the image as a surface  $g$  defined in 3D Euclidean space over the rectangular domain  $\Omega = [0, 1] \times [0, 1]$ , that is  $g : \Omega \mapsto \mathcal{R}$ .

The scale factor is a value  $c$  that minimizes the following energy functional of the reconstructed thin plate  $F$ :

$$E(F) = E_f(F) + E_s(F) \quad (4)$$

such that  $E_f(F)$  and  $E_s(F)$  are the data fitting error and the smoothness measure, respectively, given below:

$$E_f(F) = \iint_{\Omega} \omega(x, y) |F(x, y) - [g(x, y) + C(x, y)]|^2 dx dy \quad (5)$$

$$E_s(F) = \iint_{\Omega} [(\frac{\partial^2 F}{\partial x^2})^2 + 2(\frac{\partial^2 F}{\partial x \partial y})^2 + (\frac{\partial^2 F}{\partial y^2})^2] dx dy \quad (6)$$

and:

$$\omega(x, y) = \begin{cases} 0 & (x, y) \text{ is a penumbra point} \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

and since  $C(x, y)$  is assumed constant in the shadow region,  $C(x, y) = c$  iff  $(x, y)$  is a point inside the shadow region but not the penumbra and equals 0 elsewhere.

Formulating the problem in the discrete domain is straightforward. The surface  $g$  is defined as a lattice consisting of pixel values in the input channel and Equation (6) is calculated using finite differences. Note that the surface  $g$  represents the intensity surface and not the actual geometry of the objects in the scene, although the two are correlated.

Even when the value  $c$  is given, finding such  $F$  in the discrete domain that minimizes  $E(F)$  is a computationally intensive task even for moderate-sized images, as it requires the solution of a linear system consisting of a linear equation for each image pixel. There are several studies that address the computational problem of surface reconstruction, e.g. by using Finite Element methods [3] or multi resolution approaches [11]. Nevertheless, we developed a light-weight approximation to the thin plate model using cubic smoothing splines, which are known in their robustness in fitting noisy data while maintaining the continuity constraint of first and second order derivatives.

The splines are constructed based on the shadow edge and penumbra region as illustrated in Figure 4. Given an image with shadow as in Figure 4a, we create a spline for each pixel of the shadow edge. Each spline is extended bidirectionally from its associated pixel in directions perpendicular to the shadow edge as seen in Figure 4a. The extent of the spline was found to be dependent on the thickness of the penumbra region. In our experiments we set the extent to be three times the width of the penumbra. Finally, all data sites within the penumbra region are ignored as they represent missing data that is to be calculated by the thin plate reconstruction. To achieve this we use the penumbra mask image (see Section 4) as in Figure 4b. The final result is shown in Figure 4c.

Each cubic smoothing spline  $s$  is the minimizing curve of the energy functional (Equations (4)-(6)) calculated in 1D, with respect to the 1D sampled image data  $g(t)$ :

$$E(s) = \int \omega(t) |s(t) - [g(t) + C(t)]|^2 dt + \int (\frac{\partial^2 s}{\partial t^2})^2 dt \quad (8)$$

where:

$$\omega(t) = \begin{cases} 0 & s(t) \text{ is in the penumbra region} \\ 1 & \text{otherwise} \end{cases} \quad (9)$$

and  $C(t) = c$  iff  $s(t)$  is a point inside the shadow region but not the penumbra and equals 0 elsewhere. The term  $c$  is the scale factor within the shadow region and is constant for all splines.

The total energy for the set  $S$  of splines is denoted as  $\hat{E}$  and calculated as:

$$\hat{E} = \sum_{s \in S} E(s) \quad (10)$$

Evaluating the unique scale factor  $c$  for the shadow region is performed by minimizing the energy in Equation (10) using a gradient descent algorithm over  $c$ . Note that  $\hat{E}$  as a function of scale factors is always convex, thus the gradient descent algorithm produces a global minimum.

In this manner we obtain the correct scale factor  $c = C(x, y)$  for the shadow region, taking into account the

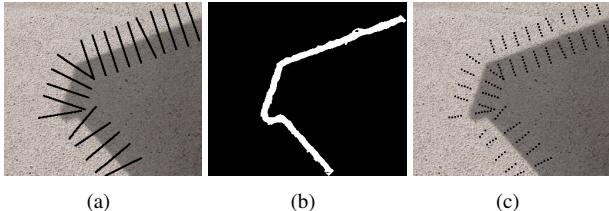


Figure 4. Finding shadow scale factor using splines. (a) Spline layout along the shadow edge when projected on the image plane. (b) Penumbra mask in which penumbra pixels are labeled in white. (c) Final sampling sites of the splines. Penumbra regions are ignored.

curved intensity-surface which may cause non-zero differences between pixels on either side of the shadow boundary. Given the scale factor, the shadow-free image can be reconstructed inside the shadow region (excluding the penumbra) using Equation (3). Note that the thin plate in itself can not be used as the intensity-surface since it is smooth (due to the smoothness constraint) and does not capture the textural information of the surface which appears in the form of local variations in intensity. Reconstructing the internal shadow region using the evaluated scale factor and Equation(3) preserves the textural information. An example is given in Figure 5b.

### 3.2. Calculating penumbra scale factors

Having determined the scale factor within the shadow region, we now reconstruct the scale factors across the penumbra. While the scale factors of each pixel inside a shadow region are uniform, this is not the case in penumbra regions. Across the penumbra region shadow scale factors vary gradually from zero to the constant scale factor value inside the shadow region. Width and rate of transition from light to shadow are penumbra properties which are determined by many factors such as the shape and distance of the light source, shape and distance of the occluding object and diffraction of light rays [13]. Since these parameters can not be easily extracted from a single image, it is impossible to calculate scale factors of penumbra pixels based solely on an analytical model of the penumbra.

Estimating the penumbra scale factors by interpolating values (linear or higher order) between the internal shadow scale factor and zero produces incorrect results and artifacts as observed also by [7].

Although scale factors across the penumbra do not change linearly (nor second order) the shadow-free intensity surface is assumed to behave as a thin plate, including the penumbra region. Thus we again use the thin plate  $F$  estimated above to reconstruct the penumbra region. We estimate the scale factors  $C(x, y)$  in the penumbra region by evaluating the difference between the smooth thin plate

and the penumbra pixels in the original image:

$$\hat{C}(x, y) = F(x, y) - I(x, y) \quad (11)$$

If the intensity surface was perfectly smooth, the estimated scale factors could be used to correctly reconstruct the penumbra regions, again, using Equation(3). However for textured surfaces such a correction would eliminate the texture. In these cases, the estimated scale factors  $\hat{C}(x, y)$  are not smooth and vary according to the surface texture. Hence a smoothing process is applied to the estimated scale factors  $\hat{C}(x, y)$  to produce a smoothly varying profile of scale factors across the penumbra. These smoothed scale factors can then be applied to image pixels in the penumbra region using Equation(3). In practice, however, the smoothing process produces artifacts in the reconstructed penumbra region in the form of Mach Bands as can be seen in Figure 5c. This is due to the sigmoidal-like profile of the shadow edge. To overcome this problem we perform directional smoothing rather than homogeneous smoothing. In our implementation we used shadow edge information to compute the exact direction of the smoothing mask for each penumbra pixel. The final shadow-free image of the image in figure 5a derived by applying directional smoothing is depicted in figure 5d.

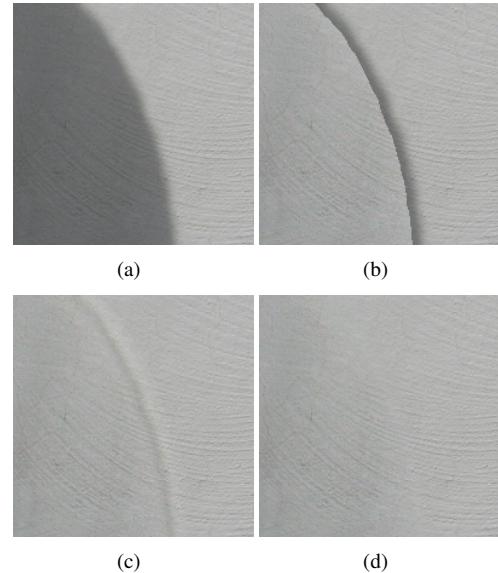


Figure 5. Calculating penumbra scale factors. (a) A shadow image. (b) Applying shadow scale factor to the shadow region. Penumbra pixels are left untouched. (c) Mach Bands effect due to homogeneous smoothing of the scale factors. (d) Result obtained when performing directional smoothing of the scale factors.

### 4. Labeling penumbra pixels

In order to obtain an accurate estimation of the shadow scale factor using the method suggested in this paper, it is

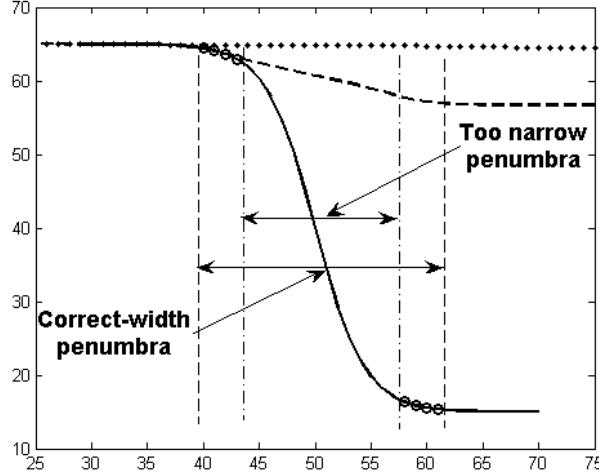


Figure 6. The effect of incorrect labeling of penumbra pixels. Solid line: the original shadowed signal. Dashed line: reconstructed spline using incorrectly labeled penumbra pixels (marked by small circles), resulting in a smaller scale factor than expected. Dotted line: correct spline reconstruction when penumbra width is correctly determined.

imperative to correctly determine the penumbra regions. Incorrect labeling of penumbra pixels as non-shadow pixels or as internal shadow pixels produces an incorrect scale factor that is smaller than the true value. This is shown in Figure 6; When computing the thin plate splines, the fitting term (Equation (5)) of the energy functional forces the spline (dashed line in Figure 6) to fit the incorrectly labeled penumbra pixels (depicted as circles on the solid line in Figure 6), thus resulting in an attenuated scale factor.

In an attempt to correctly label penumbra pixels, we consider image gradients. It has been shown [15, 18], that natural images are characterized by the sparsity of their gradient magnitudes field. This implies that for any image pixel, the probability of it being an edge pixel, and in particular a shadow edge pixel, is relatively low. Based on this insight, we introduce a method to accurately label penumbra pixels.

Consider an image  $I$  of size  $N \times M$  and its gradient magnitude field  $\|\nabla I\|$ . The gradient magnitude distribution image  $P^I$  is calculated as follows:

$$P_{xy}^I = \Pr(\|\nabla I(x, y)\|) \quad (12)$$

where  $\Pr(\|\nabla I(x, y)\|)$  is the probability of the gradient magnitude at pixel  $(x, y)$  in the image.

At this stage, simply labeling penumbra pixels by direct thresholding of  $P^I$  produces many false alarms and misses. Instead our suggested approach exploits pixels with strong evidence of being edge pixels, and propagates this evidence to their neighboring pixels. Thus low-evidence pixels are supported by neighboring edge pixels. We implement this scheme using a Markov Random Field (MRF) [10] over the image  $P^I$  such that a unique random variable is associated

with each pixel in  $P^I$ . The random variables are defined over the domain  $\{1, 0\}$ , denoting whether the underlying pixel should be labeled as an edge pixel or not, respectively. Thus, the probability of a pixel to be labeled as an edge pixel depends not only on its probability in  $P^I$  but on the probabilities of its neighboring pixels as well.

Let  $f_{xy}$  be the MRF random variable at location  $(x, y)$ . We define the *posterior energy* [10] of the field  $f$  as follows:

$$\sum_{x,y} (1 - f_{xy})[(1 - P_{xy}^I) + \sum_{f_{x'y'} \in \mathcal{N}_{xy}} \psi(f_{x'y'}, f_{xy})] + \alpha f_{xy} \quad (13)$$

where  $\mathcal{N}_{xy}$  represents the 4-neighborhood of pixel  $(x, y)$ . The term  $(1 - P_{xy}^I)$  is the *prior energy* related to the probability of a pixel being an edge pixel.  $\psi(f_{x'y'}, f_{xy})$  is the *likelihood energy* of a pixel which depends on its neighboring pixels:

$$\psi(f_{x'y'}, f_{xy}) = \begin{cases} 1 & f_{x'y'} \neq f_{xy} \wedge |P_{x'y'}^I - P_{xy}^I| \leq t_1 \\ & f_{x'y'} = f_{xy} \wedge |P_{x'y'}^I - P_{xy}^I| \geq t_2 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The defined MRF is parameterized by  $\alpha$ ,  $t_1$  and  $t_2$ . Parameter  $\alpha$  bounds the local energy of a pixel when labeled as an edge pixel. Parameters  $t_1$  and  $t_2$  are thresholds,  $t_1 < t_2$ . The likelihood energy penalizes for neighboring pixels differing in label but with similar edge probabilities, and neighboring pixels of similar label but differing in edge probabilities.

Although there exist numerous methods for automatically estimating optimal parameter values [10], we set  $\alpha$ ,  $t_1$  and  $t_2$  manually:  $t_1 = 0.05$ , and  $t_2 = (\max P^I - \min P^I) - t_1$ . Setting  $\alpha = 1.45$  produced good results on all our test images.

Minimization of the posterior energy in Equation (13) over the field  $f$  is implemented using a gradient descent algorithm. Given the minimizing  $f$ , we extract the penumbra pixels by exploiting shadow edge information, namely by finding binary connected components on  $f$  originating from pixels that appear both in the shadow edge image and in  $f$ .

## 5. Experimental results

Several real shadow images and their resulting shadow-free images produced by our algorithm are shown in Figure 7. Figure 7a contains an in-door image of a curved sheet of text covered paper. The shadow-free image (center) was obtained using the method of [5]. Since the penumbra falls on a line of text, and since shadow gradients are set to zero in [5], the line of text is completely obliterated. The shadow-free image (bottom) is the result of our algorithm. The textural information is preserved. Additionally, the shadow region in the center image appears darker than in the bottom image due to the inability of the method in [5] to derive the correct scale factor on curved surfaces.



(a)

(b)

(c)



(d)

(e)

(f)

Figure 7. Experimental results of our proposed method. (a-c) Shadow images of curved surfaces. (d-f) Shadow images of highly textured surfaces.

The results in Figures 7 show additional examples demonstrating the ability of our algorithm to remove shadows while preserving textural information in penumbra regions. This can be further appreciated by examining the shadow-free text image in Figure 7d, where the text remains visually intact despite the significant width of the penumbra.

## 6. Conclusions

In this paper we presented a shadow removal method that can handle images containing shadows cast on curved surfaces, as well as retaining the original textural information in the shadow-free image. In the proposed method, shadow is removed by determining a shadow scale factor for each shadow pixel, and then adding these scale factors in the log domain. We demonstrated that simply calculating the scale factor of shadow regions from the pixels adjacent to shadow edges can result in inaccurate scale factor estimation on curved surfaces. We proposed to use cubic smoothing splines, as an approximation of a thin plate surface in penumbra regions, to calculate the scale factor. Furthermore, we also showed how to exploit the thin plate approximation in order to obtain a high-quality shadow-free image, in which the textural information of the original image is preserved. The approach performs well even for images containing shadows with wide penumbras, as shown by several examples.

## References

- [1] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*. Academic Press, 1978.
- [2] O. Bryt and A. Adler. Shadow detection and removal research. Technical report, Technion, 2005.
- [3] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11):1131–1147, 1993.
- [4] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28:59–68, 2006.
- [5] G. D. Finlayson, S. D. Hordley, and M. S. Drew. Removing shadows from images. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 823–836, London, UK, 2002. Springer-Verlag.
- [6] C. Fredembach and G. Finlayson. Hamiltonian path based shadow removal. In *Proceedings of 16th British Machine Vision Conference (BMVC)*, pages 970–980, 2005.
- [7] C. Fredembach and G. Finlayson. Simple shadow removal. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pages 832–835, Washington, DC, USA, 2006. IEEE Computer Society.
- [8] S. Lee, K. Chwa, J. Hahn, and S. Shin. Image morphing using deformable surfaces. In *Image Morphing Using Deformable Surfaces*, 1994.
- [9] M. D. Levine and J. Bhattacharyya. Removing shadows. *Pattern Recogn. Lett.*, 26(3):251–265, 2005.
- [10] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [11] C. Lürig, L. Kobbelt, and T. Ertl. Hierarchical solutions for the deformable surface problem in visualization. *Graphical Models*, 62(1):2–18, 2000.
- [12] P. Mamassian, D. C. Knill, and D. Kersten. The perception of cast shadows. *Trends in Cognitive Sciences*, 2(8):288–295, Aug. 1998.
- [13] H. Primack, H. Schanz, U. Smilansky, and I. Usiskin. Penumbra diffraction in the quantization of dispersing billiards. *Physical Review Letters*, 76:1615–1618, Mar. 1996.
- [14] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Comput. Vis. Image Underst.*, 95(2):238–259, 2004.
- [15] E. P. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conf on Signals, Systems and Computers*, pages 673–678, Pacific Grove, CA, 1997. IEEE Computer Society.
- [16] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214, New York, NY, USA, 1987. ACM Press.
- [17] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3D object recognition. *IJCV*, 1(3):211–221, Oct. 1987.
- [18] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, pages 68–75, 2001.