

The Shadow Meets the Mask: Pyramid-Based Shadow Removal

Yael Shor Dani Lischinski

School of Computer Science and Engineering
The Hebrew University of Jerusalem

Abstract

In this paper we propose a novel method for detecting and removing shadows from a single image thereby obtaining a high-quality shadow-free image. With minimal user assistance, we first identify shadowed and lit areas on the same surface in the scene using an illumination-invariant distance measure. These areas are used to estimate the parameters of an affine shadow formation model. A novel pyramid-based restoration process is then applied to produce a shadow-free image, while avoiding loss of texture contrast and introduction of noise. Unlike previous approaches, we account for varying shadow intensity inside the shadowed region by processing it from the interior towards the boundaries. Finally, to ensure a seamless transition between the original and the recovered regions we apply image inpainting along a thin border. We demonstrate that our approach produces results that are in most cases superior in quality to those of previous shadow removal methods. We also show that it is possible to easily composite the extracted shadow onto a new background or modify its size and direction in the original image.

1. Introduction

The removal of shadows from a single image is an interesting and important research problem. In computational photography one may wish to remove shadows due to aesthetic reasons — for example, a shadow cast by the photographer onto the scene, or shadows added due to the use of a flash. Shadows also often interfere with common computer vision tasks, such as segmentation, tracking, and object recognition [UII96]. Thus, removing shadows as a preprocessor to these tasks could result in improved performance. In addition, special effects often require removing objects, in which case their shadows must be removed as well. Finally, shadows provide important visual cues to our perception of shape, occlusion, contact, etc. Thus, being able to remove, add, and modify shadows in an image is an important image manipulation tool.

The shadow removal process consists of two challenging subtasks: detecting the shadowed region and restoring the illumination in that region. The detection task involves some degree of image understanding in order to determine whether a pixel is dark due to a shadow or the reflectance at the corresponding scene point. Accomplishing this task requires

making some assumptions about the shadowed surfaces in the scene and/or asking the user for some hints. The restoration task is also challenging, as it attempts to eliminate any perceivable differences between the originally lit and the restored parts of the image. In particular, it is difficult to avoid differences in local contrast and in the amount of noise between the two regions. Another difficulty is presented by shadowed regions where the intensity of the shadow is non-uniform.

In this paper we propose new techniques addressing both of the subtasks mentioned above. Our main technical contributions are in the illumination restoration stage, consisting of a novel pyramid-based restoration process that avoids loss of texture contrast in the recovered regions by applying an affine shadow recovery model at multiple scales. A more detailed overview of our approach and contributions follows the survey of related work below.

1.1. Related Work

Shadow detection and removal are closely related to the recovery of intrinsic images by separating an image to its reflectance and illumination components [BT78]. It is possible to derive the reflectance component from a sequence of im-

ages with varying illumination [MNIS04, Wei01]. However, in this work we are interested in the case of a single image.

A method applicable to single images is described by Finlayson *et al.* [FDL04, FHD02]. In this approach, an *illuminant invariant* image is obtained by projecting the colors in an image onto a direction orthogonal to that of the intensity and color change. The edges that are present in the original image but absent in the invariant image are assumed to lie on the boundaries of the shadows. Zeroing the corresponding gradients and re-integrating yields a shadow-free image. Re-integration may be done by solving the 2D Poisson equation [FDL04, FHD02], or by performing 1D integration along a Hamiltonian path that enters and exits each shadow region exactly once [FF05]. A further simplification, which replaces the integration by scaling the shadow with a single constant factor (per color channel) has recently been proposed [FF06].

These conceptually elegant methods have produced some impressive results and represent the state-of-the-art in automatic shadow removal from a single image. However, they assume that within the shadow region the illumination is constant and the gradients are entirely due to changes in reflectance. Such an assumption does not account for the self-shadowing caused by the small scale geometry on the surface and for the changes in the amount of ambient illumination that might occur across the shadowed region. Furthermore, precise detection of shadow edges is a delicate process, which requires careful parameter tuning. Small errors in this process introduce visible errors in the images reconstructed after these edges are removed. The methods of Fredembach and Finlayson [FF05, FF06] further assume that the shadow boundaries are rather sharp and that an entry point may be identified where the pixels on both sides of the shadow boundary have the same reflectance (and thus the gradient across the boundary is only due to the change in illumination).

Mohan *et al.* [MTC07] describe a shadow editing tool that does not assume sharp shadow edges. By fitting a gradient-domain shadow edge model to the shadows they are able not only to remove shadows, but also to perform a variety of other shadow manipulations. However, in their system, the user is asked to mark rough estimates of shadow edges and to provide an initial estimate of the shadow sharpness, while we are interested in a nearly-automatic approach.

Oh *et al.* [OCDD01] describe a “texture-illuminance decoupling filter”, which uses the bilateral filter to decouple large and small-scale features thereby discounting the effect of illumination on uniformly textured areas. This method assumes that all large scale variations on such surfaces come from changes in illumination and does not handle detailed shadows of small objects correctly. Furthermore, the user is required to specify the texture feature size.

Also related are methods for *shadow matting*, whose main goal is to extract a matte for compositing a shadow onto

a new background. Chuang *et al.* [CGC^{*}03] introduce a shadow compositing equation

$$I = \beta L + (1 - \beta)S, \quad (1)$$

which expresses the observed image I as a linear combination of a shadow-free image L and a (fully) shadowed image S . They use a video sequence of the shadow-casting object moving in front of a static background to extract the images L and S , and then obtain the shadow matte β from eq. (1). In contrast, we recover L from a single image, and use a different (and more general) affine relationship between the shadowed and the lit regions.

Wu *et al.* [WTBS07] extract both a shadow-free image and a shadow matte from a single image. However, their approach requires considerable user input: a *quadmap* indicating shadowed and lit regions with similar textures, regions of uncertainty, and regions to be excluded. As we shall demonstrate, the approach presented in this paper is more automated, while producing results of comparable or better quality.

1.2. Overview

In this paper we propose a new method for shadow removal from a single image. Our goal is a nearly-automatic tool capable of producing high-quality shadow-free images. Our method is applicable to cast shadows resulting from the occlusion of a single primary light source (*e.g.*, outdoor scenes). We further assume that each shadow to be removed is cast onto a scene surface (a region with coherent color and texture) that has both shadowed and unshadowed (lit) parts. The shadow may, however, cover additional adjacent regions. It should be noted that, in practice, we have been able to handle all of the test cases used in previous methods.

Our shadow removal process is depicted in Figure 1. It begins with the user indicating the shadow to be removed by a single mouse click in the interior of the shadowed region (Figure 1a). From this minimal user input we automatically compute a mask that indicates the lit part of the surface (Figure 1c), another mask that indicates the shadowed part (Figure 1d), and a mask for the entire shadowed area (Figure 1e). This process is described in Section 2.

In Section 3, which contains the main technical contributions of our work, we present a novel pyramid-based restoration process that is applied to produce the shadow-free image without loss of texture contrast or introduction of noise. We begin by deriving an affine relationship between the colors of shadowed pixels in an image and their unshadowed colors. The parameters of this affine model may be estimated for different areas inside the shadowed region using the masks obtained in the previous stage. Doing this at multiple scales makes it possible to accurately reproduce the texture contrast in the recovered areas. Furthermore, in contrast to previous approaches, we account for varying shadow intensity inside

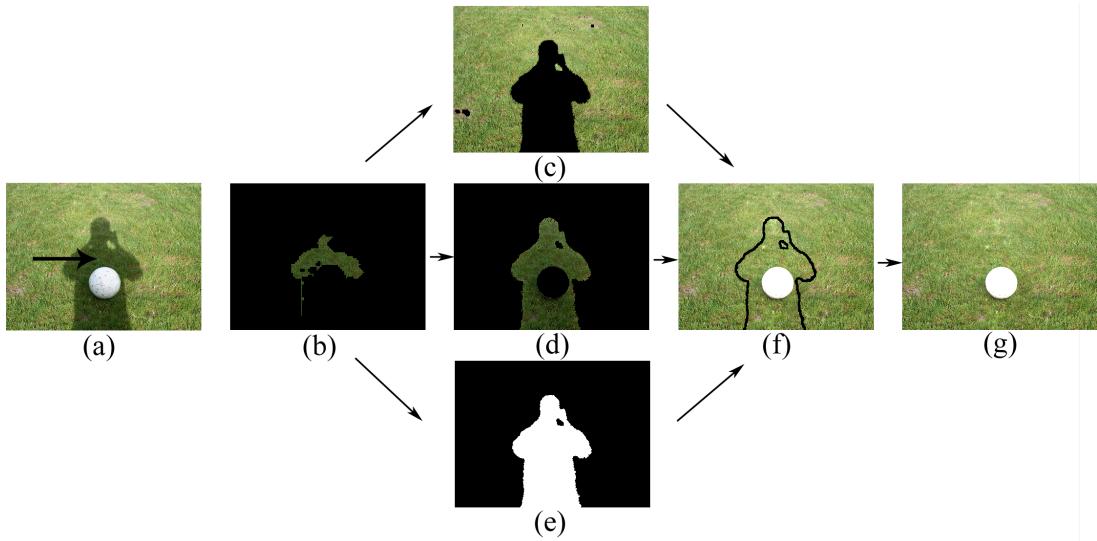


Figure 1: Our shadow detection and removal process: (a) input image (the arrow points to the user-indicated shadowed pixel). (b) shadow seed. (c) and (d) lit and shadowed areas on the same surface are detected. (e) a complete shadow mask is computed. (f) the interior of the shadowed region is restored. (g) the removal process is completed by inpainting along the borders.

the shadowed region by processing it from the interior towards the boundaries.

Finally, to ensure a seamless transition between the original and the recovered regions we apply image inpainting along a thin border (Section 4). In Section 5 we demonstrate that our approach produces results that are generally superior in quality to those of previous automatic shadow removal methods. Furthermore, our results also compare favorably with those of previous interactive methods, although we require only minimal user intervention. We also demonstrate that it is possible to easily composite the shadows extracted by our method onto a new background or modify their size and direction in the original image.

2. Detection

The goal of the detection phase is to identify the shadowed pixels to be recovered, as well as identify additional regions that will be of use later in the recovery phase. More specifically, the outcome of this phase is a collection of three masks:

1. A mask M_s , which specifies an area that is entirely inside the shadow, with all of its pixels belonging to a single surface (a region with roughly uniform color and texture).
2. A mask M_l , which specifies an area entirely outside the shadow, with all of its pixels belonging to the same surface as the previous mask.
3. A mask M_{shadow} , which specifies all of the pixels where shadow removal is required.

The first two masks are used in the recovery phase to derive the parameters of the shadow formation model. An example of these three masks is shown in Figure 1(c–e). Note that M_s is a proper subset of M_{shadow} , which may consist of shadowed regions on several different surfaces (e.g., the soccer ball in Fig. 1).

2.1. The shadow seed

To initiate the shadow removal process, the user indicates the shadow to be removed. This may be done by a single mouse click in the interior of the shadow, provided that the indicated shadow location is on a surface that has both shadowed and lit parts. For example, in Figure 1 the user should click on the shadowed grass, rather than on the soccer ball. Next, an iterative region growing process is applied that extends the user-indicated point into a small patch inside the shadowed portion of the surface. We refer to the resulting patch as the *shadow seed*. An example is shown in Figure 1b.

More specifically, we perform a few (3–4) region growing iterations, where in each iteration pixels that are adjacent to the region are added to it, provided their colors lie within a small tolerance from pixels already in the region, while taking care not to cross strong edges in the image. This is done on a filtered and downsampled version of the image in order to reduce sensitivity to small variations in color across a surface, as well as to speed up the process.

2.2. Computing M_s and M_l

We next extend the shadow seed to a complete mask M_s for the shadowed portion of the surface and compute a mask

M_l for the illuminated portion of the same surface. Our approach is to first identify all of the pixels in the image that belong to the same surface as the shadow seed, either illuminated or shadowed, and then grow the shadow seed on that surface until it contains the entire shadowed region. The complementary part of the surface will be our illuminated region. The image is filtered using a bilateral filter prior to this stage, for the same reasons mentioned above.

To find other pixels (either shadowed or illuminated) on the same surface as the shadow seed, we select pixels whose distance from the color of the shadow seed is small under an illumination invariant color distance measure. This is similar in spirit to Finlayson's idea of computing an invariant image [FHD02]. Our invariant distance between two RGB colors is defined as $1 - |\cos \theta|$, where θ is the angle between their corresponding 3-vectors. Our intuition here is that the RGB colors of points with similar reflectances in the scene correspond to nearly collinear vectors, both when shadowed and when illuminated. A related observation was also exploited by Omer and Werman [OW04]. This is a reasonable approximation if the illuminant is roughly white (has a roughly uniform spectrum). This assumption is only used in the detection phase and will be relaxed in Section 3.

Using the above distance measure, we compute the distance between each pixel in the image and the median color of the shadow seed. When examining the histogram of these distances, one typically observes a large peak of pixels whose distance from the seed is near zero. We therefore set a threshold in the first valley in the histogram following this initial peak. All of the pixels whose distance is below the threshold are then classified as belonging to the same surface as the seed.

Next, in order to identify the shadowed part of the surface we look for a connected region with a similar distribution of values to that of the pixels inside the shadow seed. This is done by another iterative region growing process starting from the shadow seed and using only those pixels that we classified as belonging to the surface containing the seed. Let A denote this set of pixels on the surface. We initialize M_s to contain the pixels of the shadow seed, and initialize M_l to $A - M_s$. In each iteration some pixels in M_l that are spatially adjacent and similar in color to pixels in M_s are moved from M_l to M_s . In the course of this process the standard deviation of the pixels in M_s increases, while that of M_l decreases. We stop the process one iteration *before* the standard deviation of the pixels in M_s exceeds that of M_l . The intuition behind this heuristic is that the standard deviation among the shadowed pixels should not exceed that of the lit pixels on the same surface, and we found it to work well in practice.

In the above process we use two different metrics to measure the proximity between colors. The first metric simply looks at the Y (luma) channel, since the shadowed pixels should have similar brightness to that of the shadow seed, while the lit pixels should be brighter. However, in practice



Figure 2: The results of the shadow detection phase. Each row shows from left to right: input image, the unshadowed pixels in M_l , the shadowed pixels in M_s , the complete shadow mask M_{shadow} .

there are also cases (*e.g.*, due to self-shadowing) where some pixels in the unshadowed region are also quite dark. In such cases we found it more effective to measure proximity between the pixels using their chroma (Cb,Cr) channels. We therefore perform the region growing process twice, using the two different metrics. Eventually, the process that induces the smallest number of connected components in M_s and M_l is used.

2.3. Computing M_{shadow}

To obtain a complete binary mask of the shadowed region we use the alpha matting approach described by Levin *et al.* [LLW06]. As input to the alpha matting algorithm we construct a *trimap*, which indicates for each pixel whether it is definitely inside the shadow, definitely outside the shadow, or unknown. The matting algorithm produces a complete alpha channel for the image. This matte is then thresholded (at $\alpha = 0.5$) to produce the binary shadow mask M_{shadow} .

The trimap is constructed as follows: we start with the M_s region obtained earlier in the process and augment this region with adjacent regions that are also shadowed, but belong to a different surface in the scene (*i.e.*, pixels which are surrounded by M_s , but do not belong to M_l).

All of the pixels in this augmented region are considered to be definitely in shadow. We then dilate this region and complement it to obtain the pixels considered to be definitely outside. The remaining pixels are considered unknown.

Figure 2 shows the three masks computed using the techniques described in this section on two examples.

3. Illumination Recovery

We begin this section with the derivation of the affine shadow formation model, which we use (after estimating its parameters) in order to recover the illumination inside the shadowed areas. The parameter estimation is described in section 3.2. Section 3.3 includes the description of our multi-resolution recovery procedure in which we match statistics across multiple scales and section 3.4 describes how we handle non-uniform shadows.

3.1. Shadow formation model

We start with the familiar image formation equation [BT78],

$$I(x, \lambda) = L(x, \lambda) R(x, \lambda), \quad (2)$$

where $I(x, \lambda)$ is the intensity reflected from point x in the scene at wavelength λ , while L and R are the illumination and the reflectance at the same point and wavelength. Assuming a scenario where the cast shadows are due to a single primary source of illumination (such as the Sun in outdoor scenes). If a point x in the scene is unshadowed (lit), we may express the illumination there as a sum of two terms,

$$L(x, \lambda) = L^d(x, \lambda) + L^a(x, \lambda), \quad (3)$$

where L^d is the direct illumination and L^a is the remaining indirect (ambient) illumination. Thus,

$$I^{\text{lit}}(x, \lambda) = L^d(x, \lambda) R(x, \lambda) + L^a(x, \lambda) R(x, \lambda) \quad (4)$$

Next, suppose that some object in the scene occludes the primary light source, casting a shadow on point x . Note that the same occluder would typically also block some of the ambient illumination that would have otherwise arrived at x . In this case, the reflected intensity is

$$I^{\text{shadow}}(x, \lambda) = a(x) L^a(x, \lambda) R(x, \lambda), \quad (5)$$

where $a(x)$ is a spatially variant factor that accounts for the attenuation of the ambient illumination by the occluder inside the shadowed area. Here we assume that the ambient illumination has roughly the same spectral distribution from all incident directions, otherwise the attenuation factor a should also depend on the wavelength λ .

Combining the last two equations we can express the lit intensity at x as an affine function of the shadowed intensity there:

$$I^{\text{lit}}(x, \lambda) = L^d(x, \lambda) R(x, \lambda) + \frac{1}{a(x)} I^{\text{shadow}}(x, \lambda). \quad (6)$$

When a scene is photographed, the actual color at a pixel p corresponding to point x in the scene is obtained by integrating $I(x, \lambda)$ with the camera's sensor spectral response functions. This linear operation does not change the affine nature of the relationship between the shadowed and illuminated intensities, and we obtain

$$I_k^{\text{lit}}(p) = \alpha_k(p) + \gamma(p) I_k^{\text{shadow}}(p), \quad (7)$$

where $\alpha_k, k \in \{R, G, B\}$, is the response of the camera to the reflected direct illumination in the three *RGB* color channels, and $\gamma(p) = 1/a(x)$ is the inverse of the ambient attenuation factor, which, under our assumptions, does not depend on the wavelength. Thus, the illuminated pixel color may be recovered from its shadowed color by estimating those four affine parameters.

It should be noted that while Chuang's [CGC*03] shadow matting equation (1) also implies an affine relationship between the lit and the shadowed intensity at a pixel, their

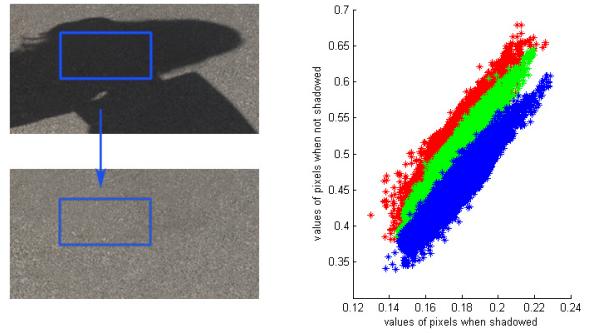


Figure 3: The relation between the intensities of pixels in a shadowed scene (top left) and their intensities in the same scene without a shadow (bottom left), for each of the three color channels. The horizontal axis corresponds to the shadowed intensity, and the vertical axis to the illuminated intensity. The plotted pixels come from the window indicated in blue.

derivation does not account for the variation in the attenuation of ambient light inside the shadowed regions, and they have not used it to recover the lit intensities from the shadowed ones.

A qualitative empirical demonstration of our affine model is shown in Figure 3. Two photographs of the same outdoor scene were taken in rapid succession with and without the presence of an object casting a shadow on the floor. For each pixel in the shadowed region we plot its shadowed intensity (horizontal axis) versus its illuminated intensity (vertical axis). Each of the three RGB color channels is plotted with the corresponding color. Examining this plot one can indeed observe a roughly affine relationship similar to that of equation (7): The relationships for the three color channels look like straight lines with the same slope, but shifted by different amounts from the origin. The fact the the lines are not perfect may be attributed to the presence of noise, particularly in the darker shadowed pixels, and to the variations in the reflectance of the asphalt surface onto which the shadow is cast.

3.2. Parameter estimation

As shown above, in order to recover the illuminated intensity at a shadowed pixel we need to estimate the four parameters of the affine model (7). In order to estimate these parameters we use two strips of pixels: one inside the shadowed region, and the other outside the region. The pixels contained in these two strips may come from different surfaces in the scene. In order to make the parameter estimation reliable we should use only pixels that come from the same surface. Therefore, among all the pixels contained in each strip we only use those with colors that appear with sufficient frequency in the corresponding mask (M_s for the shad-

owed strip and M_l for the strip outside the shadow). Let \mathcal{S} and \mathcal{L} denote the two resulting sets of pixels (shadowed and illuminated, respectively). We then estimate the four parameters $\alpha_R, \alpha_G, \alpha_B$ and γ of eq. (7) based on the mean colors of \mathcal{S} and \mathcal{L} and the standard deviations of their luminances.

More formally, let $\mu(\mathcal{S})$ and $\mu(\mathcal{L})$ denote the mean colors of the pixels in \mathcal{S} and \mathcal{L} , and let $\sigma(\mathcal{S})$ and $\sigma(\mathcal{L})$ denote the standard deviation of their luminances. Then we set

$$\gamma = \frac{\sigma(\mathcal{L})}{\sigma(\mathcal{S})} \quad (8)$$

and

$$\alpha_k = \mu_k(\mathcal{L}) - \gamma \mu_k(\mathcal{S}), \quad k \in \{R, G, B\}. \quad (9)$$

Obviously, after applying these parameters to the pixels in \mathcal{S} , the mean and the standard deviation of the resulting set would match those of \mathcal{L} .

It should be noted that the above approach resembles the color transfer method of Reinhard *et al.* [RAGS01], which was also used by Wu *et al.* [WTBS07] in order to compute a rough approximation to the shadow-free image.

3.3. Pyramid-based restoration

Simply applying the estimated affine parameters to each pixel inside the shadowed region yields results that sometimes exhibit two artifacts, which may also be observed in the results of previous methods.

The first artifact is that the texture in the recovered region has less contrast than in the originally lit areas. One reason for this is that the directional direct illumination in unshadowed regions yields higher contrast than the hemispherical indirect illumination in shadowed regions. Moreover, neither our affine model, nor any of the previous shadow formation models, account for the fact that textured regions often exhibit some fine scale self-shadowing even in the lit regions. The second artifact is that the recovered region might exhibit more noise than the surrounding lit areas. This is particularly noticeable when the removed shadow was cast on a smooth or a dark surface in the scene.

In order to overcome these problems and match the texture appearance of the restored shadow region to the originally illuminated region, we perform the parameter estimation and the shadow removal using a Laplacian pyramid representation of the image. The intuition behind this scheme is that the texture in the recovered region will match better that of the illuminated region, if we match the color statistics of the pixels in \mathcal{S} and \mathcal{L} at a variety of spatial scales. A similar idea was previously used in Heeger and Bergen's seminal texture synthesis work [HB95], and recently, in a different context, by [ND06] for reconstruction of BTFs from sparse measurements.

More specifically, we compute the Laplacian pyramid of the image and also generate downsampled versions of the

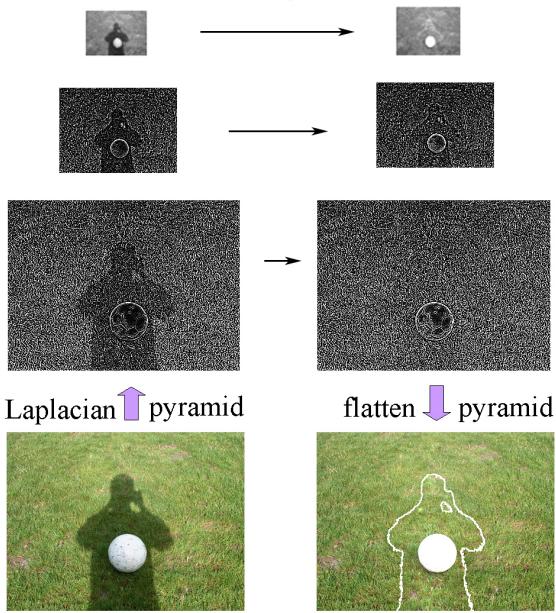


Figure 4: Shadow restoration using the Laplacian pyramid. Left column: the Laplacian pyramid of the input image. In each level the affine model is applied in the areas under M_{shadow} . The resulting modified pyramid (right column) is flattened to obtain a restored image.

简单修复会出现的问题，1.修复区域的纹理的对比度比光照区域低，2.修复区域会出现噪
声。

corresponding to each level estimate the parameters of in the previous section, and a restored level. Finally, the obtain the restored image. Figure 4. Figure 5 compares the results obtained with pyramid-based restoration to those obtained using a single (original) resolution. It may be seen that texture is preserved better and the amount of introduced noise is smaller.

To conclude, by matching the texture appearance across multiple scales both artifacts mentioned above are reduced. On the one hand, in textured shadow regions, the contrast of the recovered texture is increased by matching its statistics to those of the illuminated texture. On the other hand, in textureless areas, matching the statistics with those in the smooth illuminated region, tends to reduce the amplitude of noise.

3.4. Non-uniform shadows

Another problem we would like to deal with is the fact that in some scenes, the shadow intensity is not uniform across the shadowed region. Typically, cast shadows become darker closer to the occluding object (see Figure 8, rows 5 and 10, column a). One explanation for this is that more of the in-

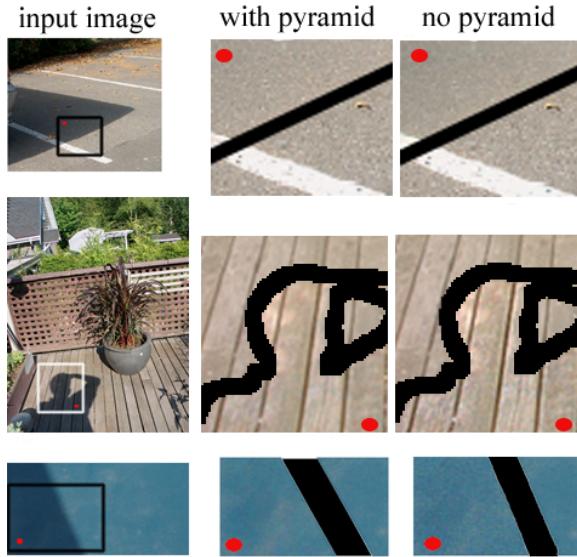


Figure 5: Multi-resolution vs. single resolution shadow restoration. Top row: the texture in the restored area (above the black border) exhibits more contrast and looks more similar to the original texture (under the black border) when restored using the pyramid-based method. Middle and bottom rows: the texture inside the restored area is smoother and looks more similar to the original when restored using the pyramid-based method. In contrast, the results in the right column exhibit some noise amplified by the restoration.

direct illumination (*e.g.*, sky light) is being blocked. Obviously, in this case the shadow cannot be completely removed by applying a single set of parameters to the entire shadowed region.

We address this problem by estimating a different set of affine parameters for different areas inside the shadowed regions. We generate a sequence of strips inside the shadowed region, expanding from the interior of the region towards its boundary (see Figure 6), and compute a different set of affine parameters for each strip. Note, that as mentioned previously, the parameters are computed only using the subset of pixels on the strip whose colors appear (with a sufficient frequency) in the corresponding mask (M_s or M_l). These parameters are then only used to recover the intensities of the pixels in the same strip. This process is only applied at the coarsest level of the pyramid (the DC component of the image), since we did not find this to be necessary at the other (band-pass) levels. The effectiveness of this approach may be seen in Figure 8 (rows 5 and 10, column b).

4. Border Handling

The borders of the shadow require special treatment since: (i) the shadow mask M_{shadow} typically has some inaccura-



Figure 6: Using different strips when dealing with a non-uniform shadow. Each strip of shadowed pixels is restored using its own set of affine parameters estimated with respect to the same strip of illuminated pixels just outside the shadowed region.



Figure 7: Left: An input image with a shadow. Middle: The same scene without the shadow (ground truth). Right: A shadow-free image produced with our method.

cies along its boundaries, and (ii) the shadow borders typically correspond to a penumbral region where the shadow intensity is changing more rapidly than in the interior of the shadow.

By processing the shadowed areas using thin strips, and fixing each strip using the affine model as described in Section 3 the problem area is typically reduced to a very thin strip of pixels along the boundary of the shadow mask which we then inpaint, as described below. Actually, we only inpaint those parts of the border strip that separate the shadowed area from the illuminated area on the same surface (as opposed to parts that separate the shadowed area from another object). These parts are easily identified by checking that pixels on one side of the border belong to M_s , while those on the other side belong to M_l .

For the inpainting task we have experimented with a number of existing approaches (*e.g.*, [CPT04]), but eventually obtained the best results with a greedy approach that uses graph-cut texture synthesis [KSE*03].

The inpainting process is quite straightforward since the region to be inpainted forms a very narrow curve. The idea is to pick uniformly spaced points along the curve, consider a window around each point and look for the closest matching window in the close vicinity of the point. To measure distances between windows we use SSD, but exclude the pixels to be inpainted. Once such a window is found it replaces the window around the point, using min-cuts to find the optimal seams, as described by Kwatra *et al.* [KSE*03].

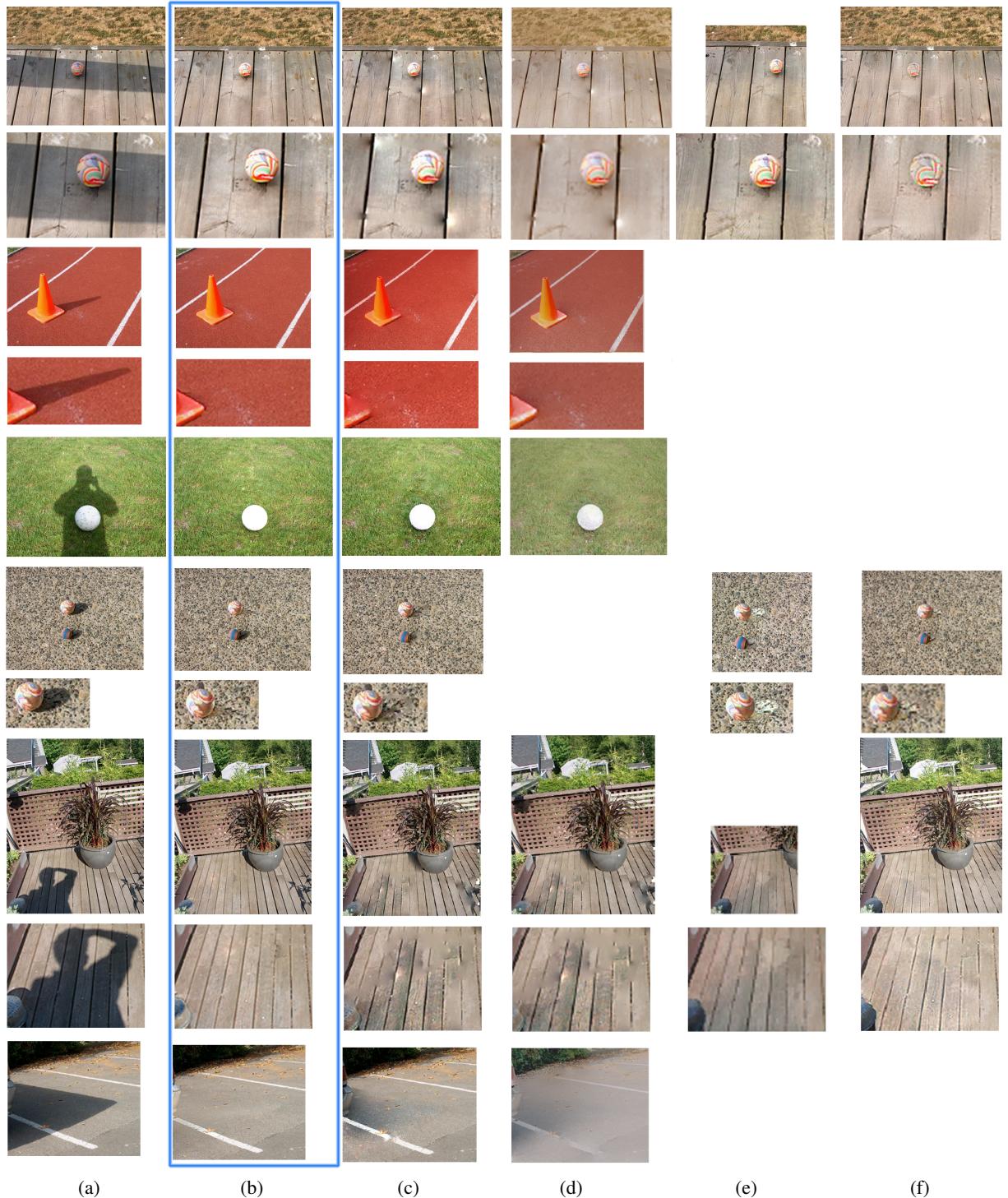


Figure 8: Comparison of our results with state-of-the-art shadow removal methods. (a) Input images. (b) Shadow removed with our method. (c) Results from Finlayson et al. [FDL04]. (d) Results from Finlayson et al. [FHLD06]. (e) Results from Hamiltonian path based shadow removal [FF05]. (f) Results from Wu et al. [WTBS07]. Please see the text for a detailed discussion of the differences.

5. Results

Our shadow removal technique was implemented in Matlab. On a 300x400 image the shadow detection stage takes 6 to 40 seconds, the recovery stage 9 to 18 seconds, and the border inpainting takes between 12 and 80 seconds. These timings were measured on a 3.0GHz CPU with 2GB RAM. The restoration process is demonstrated in the accompanying video.

In order to validate our method, we photographed the same scene, once with a shadow cast on it and once without, to serve as the ground truth shadow-free image. The two images are shown in Figure 7 (left and middle, respectively). The result of removing the shadow with our method is also shown (on the right). Visually, our result is practically indistinguishable from the ground truth image. In order to quantitatively assess our method's accuracy we also computed the relative L_1 error between our result and the ground truth image. The error, computed only for those pixels whose values were adjusted by our algorithm, was under 4 percent, on average. For comparison, the magnitude of noise in the ground truth image is between 0.5 and 1 percent (the noise magnitude was estimated from four images without the shadow, taken in rapid succession). Thus, the inaccuracies introduced by our method are larger than noise by only a small factor.

Figure 8 shows some of the shadow removal results obtained with our technique (in each image we dealt with one component of the shadow) and compares them to the results reported in previous works [FDL04, FHLD06, FF05, WTBS07], which represent the state-of-the-art in shadow removal from a single image. While all of these methods are able to effectively remove shadows, a close examination of the results reveals certain artifacts present in previous methods, but not in our results. In the results of Finlayson *et al.* [FDL04, FHLD06] (columns c and d) exhibit some smoothing artifacts along the removed shadow boundaries. The reason is that along shadow boundaries the gradients are typically only in part due to the shadow boundary, and in part due to changes in reflectance (texture). Thus, completely suppressing these gradients leads to smoothing errors in the re-integrated image. For example, in the top row, artifacts may be seen where the shadow boundary crosses the vertical dark edges between the planks of wood (easier to see in the second row, which shows a magnified portion of the images).

The Hamiltonian path based technique [FF05] (column e) does not suffer from such artifacts on the border, but one can still observe that the areas from which the shadow was removed are somewhat darker than the surrounding original illuminated areas (may be seen in rows 1–2 and 8–9). The same artifact is also present in the results of Finlayson *et al.* [FDL04, FHLD06] (particularly noticeable in rows 5 and 10). Our results do not exhibit such darkening because of our explicit handling of shadow non-uniformity.

The results reported by Wu *et al.* [WTBS07] (column f)



Figure 9: Top: extracted shadows may be composited onto a new background. Bottom: Warping the shadow mask and compositing may be used to fake changes in the direction of illumination.

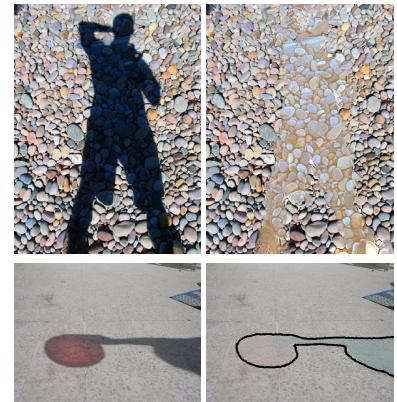


Figure 10: Some failure examples. The input is on the left and the output is on the right (without border inpainting in the bottom example).

do not exhibit the same artifacts, but under closer examination it may be seen that there is noise and some loss of color in the reconstructed areas. For example, in the result in the top row, the colors on the ball are faded and resemble those of the surrounding wood, while the result in the rows 8–9 exhibits some noise (may be seen better when the image is magnified). Furthermore, each of these results were obtained with user interaction, where the user was required to indicate the shadowed region, the corresponding illuminated region, regions to be excluded, and the region of uncertainty (along the shadow border). In contrast, the results produced with our method were obtained from minimal user input, and exhibit none of the aforementioned artifacts.

Shadow compositing: Once the shadow mask has been obtained and the affine shadow formation parameters have been estimated for the pixels inside the shadow mask, the shadow may be easily composited onto a novel background image. Several such composites are shown in Figure 9. By warping the shadow mask and compositing the warped shadow back onto the shadow-free image (or onto a new image) it is also possible to plausibly fake changes in the

direction of the illumination. This is demonstrated by the sequence of images in the bottom row of Figure 9.

Failure cases: When the assumptions of our method are strongly violated, the method may fail to produce a high quality shadow-free image. Two failure examples are shown in Figure 10. In the top example, the shadow is cast onto a surface consisting of large pebbles with very strong self-shadowing effects, and strong variations in color. Although the shadow is detected accurately, no single set of affine parameters (for the entire shadowed region, or for entire strips inside the region) is able to remove the shadow in a satisfactory manner.

The bottom example in Figure 10 features a shadow cast, in part, by a colorful semi-transparent beach ball, and in part by an opaque object. Thus, the shadow consists of parts formed by two different processes, and when it is removed by our method, some residue of its color remains in the shadow-free image.

6. Conclusions

We have described a new nearly-automatic method for producing a high-quality shadow-free image from a single input image. The main contributions of our work is a new multi-resolution restoration routine, and the ability to handle shadows of non-uniform intensity, by recovering the parameters of an affine shadow model at multiple scales and locations inside the shadowed regions. We have demonstrated that we are able to faithfully reconstruct the color and texture inside the shadowed regions without the artifacts that may be observed in the results of previous techniques and without significant user interaction.

In the future we plan to work on lifting, as much as possible, the current limitations of our method. For example, we believe our method could be extended to work on a multi-colored shadowed surface by employing a more sophisticated local parameter estimation strategy. The handling of non-uniform shadows can also be improved by analyzing the actual low-frequency changes in shadow intensity, rather than simply assuming that the shadow becomes darker away from the boundaries.

Acknowledgements: The authors would like to thank Alex Rav-Acha for his help in the beginning of this project and Anat Levin for making her matting code available. This research was supported in part by grants from the GIF foundation and the Israel Science Foundation.

References

- [BT78] BARROW H. G., TENENBAUM J. M.: Recovering intrinsic scene characteristics from images. In *Computer Vision Systems* (1978), pp. 3–26.
- [CGC*03] CHUANG Y.-Y., GOLDMAN D. B., CURLESS B., SALESIN D. H., SZELISKI R.: Shadow matting and compositing. *ACM Trans. Graph.* 22, 3 (July 2003), 494–500.
- [CPT04] CRIMINISI A., PEREZ P., TOYAMA K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing* 13, 9 (Sept. 2004), 1200–1212.
- [FDL04] FINLAYSON G. D., DREW M. S., LU C.: Intrinsic images by entropy minimization. In *Proc. ECCV* (2004), vol. III, pp. 582–595.
- [FF05] FREDEMBACH C., FINLAYSON G. D.: Hamiltonian path based shadow removal. In *Proc. BMVC05* (2005).
- [FF06] FREDEMBACH C., FINLAYSON G. D.: Simple shadow removal. In *Proc. ICPR* (2006), vol. I, pp. 832–835.
- [FHD02] FINLAYSON G. D., HORDLEY S. D., DREW M. S.: Removing shadows from images. In *Proc. ECCV* (2002), vol. IV, Springer-Verlag, pp. 823–836.
- [FHLD06] FINLAYSON G. D., HORDLEY S. D., LU C., DREW M. S.: On the removal of shadows from images. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 1 (Jan. 2006), 59–68.
- [HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In *Proc. SIGGRAPH '95* (1995), pp. 229–238.
- [KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.* 22, 3 (July 2003), 277–286.
- [LLW06] LEVIN A., LISCHINSKI D., WEISS Y.: A closed form solution to natural image matting. In *Proc. CVPR* (June 2006), IEEE Computer Society.
- [MNIS04] MATSUSHITA Y., NISHINO K., IKEUCHI K., SAKAUCHI M.: Illumination normalization with time-dependent intrinsic images for video surveillance. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 10 (2004), 1336–1347.
- [MTC07] MOHAN A., TUMBLIN J., CHOUDHURY P.: Editing soft shadows in a digital photograph. *IEEE Comput. Graph. Appl.* 27, 2 (2007), 23–31.
- [ND06] NGAN A., DURAND F.: Statistical acquisition of texture appearance. In *Proc. EGSR* (2006).
- [OCDD01] OH B. M., CHEN M., DORSEY J., DURAND F.: Image-based modeling and photo editing. In *Proc. SIGGRAPH 2001* (2001), ACM Press, pp. 433–442.
- [OW04] OMER I., WERMAN M.: Color lines: Image specific color representation. In *Proc. CVPR* (June 2004), vol. II, pp. 946–953.
- [RAGS01] REINHARD E., ASHIKHMAR M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Comput. Graph. Appl.* 21, 5 (Sept./Oct. 2001), 34–41.
- [Ull96] ULLMAN S.: *High-Level Vision: Object Recognition and Visual Cognition*. MIT Press, 1996.
- [Wei01] WEISS Y.: Deriving intrinsic images from image sequences. In *Proc. ICCV* (2001), pp. 68–75.
- [WTBS07] WU T.-P., TANG C.-K., BROWN M. S., SHUM H.-Y.: Natural shadow matting. *ACM Trans. Graph.* 26, 2 (2007), Article No. 8.