Paired Regions for Shadow Detection and Removal

Ruiqi Guo, *Student Member*, *IEEE*, Qieyun Dai, *Student Member*, *IEEE*, and Derek Hoiem, *Member*, *IEEE*

Abstract—In this paper, we address the problem of shadow detection and removal from single images of natural scenes. Differently from traditional methods that explore pixel or edge information, we employ a region-based approach. In addition to considering individual regions separately, we predict relative illumination conditions between segmented regions from their appearances and perform pairwise classification based on such information. Classification results are used to build a graph of segments, and graph-cut is used to solve the labeling of shadow and nonshadow regions. Detection results are later refined by image matting, and the shadow-free image is recovered by relighting each pixel based on our lighting model. We evaluate our method on the shadow detection dataset in Zhu et al. [1]. In addition, we created a new dataset with shadow-free ground truth images, which provides a quantitative basis for evaluating shadow removal. We study the effectiveness of features for both unary and pairwise classification.

Index Terms—Shadow detection, region classification, shadow removal, enhancement

1 Introduction

Shadows, created wherever an object obscures the light source, are an ever-present aspect of our visual experience. Shadows can either aid or confound scene interpretation, depending on whether we model the shadows or ignore them. If we can detect shadows, we can better localize objects, infer object shape, and determine where objects contact the ground. Detected shadows also provide cues for illumination conditions [3] and scene geometry [4]. But, if we ignore shadows, spurious edges on the boundaries of shadows and confusion between albedo and shading can lead to mistakes in visual processing. For these reasons, shadow detection has long been considered a crucial component of scene interpretation (e.g., [5], [6]). Yet despite its importance and long tradition, shadow detection remains an extremely challenging problem, particularly from a single image.

The main difficulty is due to the complex interactions of geometry, albedo, and illumination. Locally, we cannot tell if a surface is dark due to shading or albedo, as illustrated in Fig. 1. To determine if a region is in shadow, we must compare the region to others that have the same material and orientation. For this reason, most research focuses on modeling the differences in color, intensity, and texture of neighboring pixels or regions.

Many approaches are motivated by physical models of illumination and color [7], [8], [9], [10], [11]. For example, Finlayson et al. [10] compare edges in the original RGB image to edges found in an illuminant-invariant image. This method can work quite well with high-quality images and calibrated sensors, but often performs poorly for typical web-quality consumer photographs [12]. To improve robustness, others have recently taken a more empirical, data-driven approach, learning to detect shadows based on training images. In monochromatic images, Zhu et al. [1] classify regions based on statistics of intensity, gradient, and texture, computed over local neighborhoods, and refine shadow labels using a conditional random field (CRF). Lalonde et al. [12] find shadow boundaries by comparing the color and texture of neighboring regions and employing a CRF to encourage boundary continuity. Panagopoulos et al. [13] jointly infer global illumination and cast shadow when the coarse 3D geometry is known, using a high-order MRF that has nodes for image pixels and one node to represent illumination. Recently, Kwatra et al. [14] proposed an information theoretic-based approach to detect and remove shadows and applied it to aerial images as an enhanced step for image mapping systems such as Google Earth.

Our goal is to detect shadows and remove them from the image. To determine whether a particular region is shadowed, we compare it to other regions in the image that are likely to be of the same material. To start, we find pairs of regions that are likely to correspond to the same material and determine whether they have the same illumination conditions. We incorporate these pairwise relationships, together with region-based appearance features, in a shadow/nonshadow graph. The node potentials in our graph encode region appearance; a sparse set of edge potentials indicate whether two regions from the same surface are likely to be of the same or different illumination. Finally, the regions are jointly classified as shadow/nonshadow using graph-cut inference. Like Zhu et al. [1] and Lalonde et al. [12], we take a data-driven approach, learning our classifiers from training data, which leads to good performance on consumer-quality photographs. Unlike others, we explicitly model the material and illumination relationships of pairs of regions, including nonadjacent pairs.

- *The authors are with the Department of Computer Science, University of Illinois at Urbana Champaign, 201 North Goodwin Avenue, Urbana, IL 61801. E-mail: {guo29, dai9, dhoiem}@illinois.edu.*

Manuscript received 25 Feb. 2012; revised 5 July 2012; accepted 16 Sept. 2012; published online 1 Oct. 2012.
Recommended for acceptance by P. Felzenszwalb, D. Forsyth, P. Fua, and T.E. Boult.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2012-02-0140.
Digital Object Identifier no. 10.1109/TPAMI.2012.214.

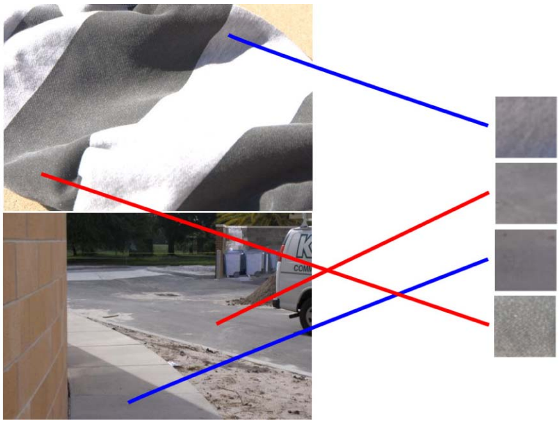


Fig. 1. What is in shadow? Local region appearance can be ambiguous; to find shadows, we must compare surfaces of the same material.

By modeling long-range interactions, we hope to better detect soft shadows, which can be difficult to detect locally. By restricting comparisons to regions with the same material, we aim to improve robustness in complex scenes, where material and shadow boundaries may coincide.

Our shadow detection provides binary pixel labels, but shadows are not truly binary. Illumination often changes gradually across shadow boundaries. We also want to estimate a soft mask of shadow coefficients which indicate the darkness of the shadow, and to recover a shadow-free image that depicts the scene under uniform illumination.

The most popular approach in shadow removal is proposed in a series of papers by Finlayson et al., where they treat shadow removal as an reintegration problem based on detected shadow edges [15], [16], [17]. Arbel and Hel-Or [18], [19] use cubic splines to recover the scalar factor in penumbra regions, and remove nonuniform shadows on curved and textured surfaces. Our region-based shadow detection enables us to pose shadow removal as a matting problem, similarly to Chuang et al. [20] and Wu et al. [21]. However, both methods depend on user input of shadow and nonshadow regions, while we automatically detect and remove shadows in a unified framework (Fig. 2).

Specifically, after detecting shadows, we apply the matting technique of Levin et al. [22], treating shadow pixels as foreground and nonshadow pixels as background. Using the recovered shadow coefficients, we calculate the ratio between direct light and environment light and generate the recovered image by relighting each pixel with both direct light and environment light.

To evaluate our shadow detection and removal, we propose a new dataset with 108 natural scenes, in which ground truth is determined by taking two photographs of a scene after manipulating the shadows (either by blocking the direct light source or by casting a shadow into the image). To the best of our knowledge, our dataset is the first to enable quantitative evaluation of shadow removal on dozens of images. We also evaluate our shadow detection on Zhu et al.'s dataset of manually labeled outdoor scenes, comparing favorably to Zhu et al. [1].

The main contributions of this paper are as follows:

1. A new method for detecting shadows using a relational graph of paired regions.

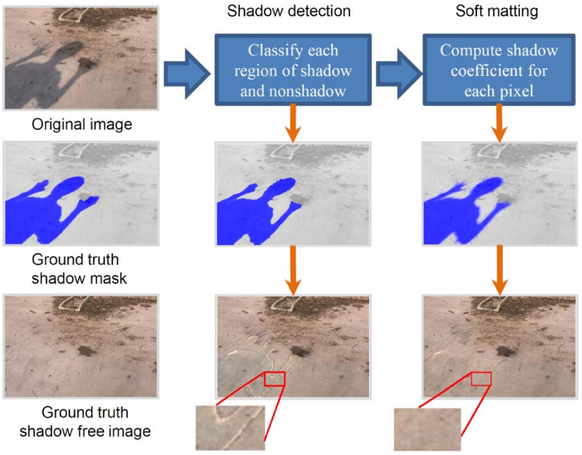

Shadow detection
Soft matting
Classify each region of shadow and nonshadow
Compute shadow coefficient for each pixel
Original image
Ground truth shadow mask
Ground truth shadow free image

Fig. 2. Illustration of our framework. First column: The original image with shadow, ground truth shadow mask, ground truth image. Second column: Hard shadow map generated by our detection method and recovered image using this map alone. Note that there are strong boundary effects in the recovered image. Third column: Soft shadow map computed using soft matting and recovery result using this map.

2. An automatic shadow removal procedure derived from lighting models making use of shadow matting to generate soft boundaries between shadow and nonshadow areas.
3. Quantitative evaluation of shadow detection and removal, with comparison to existing work.
4. A shadow removal dataset with shadow-free ground truth images.

We believe that more robust algorithms for detecting and removing shadows will lead to better recognition and estimates of scene geometry.

A preliminary version of this work appeared in [2]. This paper extends the conference version with evaluation of feature effectiveness and alternative matting strategies, additional experiments on scene-scale images, and additional discussion of applications and limitations.

2 Shadow Detection

To detect shadows, we must consider the appearance of the local and surrounding regions. Shadowed regions tend to be dark, with little texture, but some nonshadowed regions may have similar characteristics. Surrounding regions that correspond to the same material can provide much stronger evidence. For example, suppose region s_i is similar to s_j in texture and chromaticity. If s_i has similar intensity to s_j, then they are probably under the same illumination and should receive the same shadow label (either shadow or nonshadow). However, if s_i is much darker than s_j, then s_i probably is in shadow and s_j probably is not.

We first segment the image using the mean shift algorithm [23]. Then, using a trained classifier, we estimate the confidence that each region is in shadow. We also find *same illumination pairs* and *different illumination pairs* of regions, which are confidently predicted to correspond to the same material and have either similar or different illumination, respectively.

We construct a relational graph using a sparse set of confident illumination pairs. Finally, we solve for the shadow labels $\mathbf{y} = \{-1, 1\}^n$ (1 for shadow) that maximize the following objective:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \sum_{i=1} c_i^{\text{shadow}} y_i + \alpha_1 \sum_{\{i,j\} \in E_{\text{diff}}} c_{ij}^{\text{diff}} (y_i - y_j) - \alpha_2 \sum_{\{i,j\} \in E_{\text{same}}} c_{ij}^{\text{same}} \mathbf{1}(y_i \neq y_j), \tag{1}$$

where c_i^{shadow} is the single-region classifier confidence weighted by region area, $\{i, j\} \in E_{\text{diff}}$ are different illumination pairs, $\{i, j\} \in E_{\text{same}}$ are same illumination pairs, c_{ij}^{same} and c_{ij}^{diff} are the area-weighted confidences of the pairwise classifiers, α_1 and α_2 are parameters, and $\mathbf{1}(.)$ is an indicator function.

In the following sections, we describe the classifiers for single regions (Section 2.1) and pairs of regions (Section 2.2) and how we can reformulate our objective function to solve it efficiently with the graph-cut algorithm (Section 2.3).

2.1 Single-Region Classification

When a region becomes shadowed, it becomes darker and less textured (see [1] for empirical analysis). Thus, the color and texture of a region can help predict whether it is in shadow. We represent color with a histogram in L*a* b space, with 21 bins per channel. We represent texture with the texton histogram with 128 textons, provided by Martin et al. [24]. We train our classifier from manually labeled regions using an SVM with a χ^2 kernel (slack parameter $C = 1$) [25]. We define c_i^{shadow} as the log likelihood output of this classifier times a_i, the pixel area of the ith region.

2.2 Pairwise Region Relationship Classification

We cannot determine whether a region is in shadow by considering only its internal appearance; we must compare the region to others with the same material. In particular, we want to find *same illumination pairs*, regions that are of the same material and illumination, and *different illumination pairs*, regions that are of the same material but different illumination. Differences in illumination can be caused by direct light blocked by other objects, self-shading, or by a difference in surface orientation. Comparison between regions with different materials is uninformative because they have different reflectance.

We detect shadows using a relational graph, with an edge connecting each illumination pair. To better handle occlusion and to link similarly lit regions that are divided by shadows, we enable edges between regions that are not adjacent in the image. Because most pairs of regions are not of the same material, our graph is still very sparse. Examples of such relational graphs are shown in Fig. 3. When regions are classified as having different illuminations, the shadowed region is specified.

We train classifiers (SVM with RBF kernel; $C = 1$ and $\sigma = 0.5$) to detect illumination pairs based on comparisons of their color and texture histograms, the ratio of their intensities, their chromatic alignment, and their distance in the image. These features encode the intuition that regions of the same reflectance share similar texture and color distribution when viewed under the same illumination; when viewed under different illuminations, they tend to have similar texture but differ in color and intensity. We also take into account the distance between two regions, which greatly reduces false comparisons while enabling more flexibility than considering only adjacent pairs.

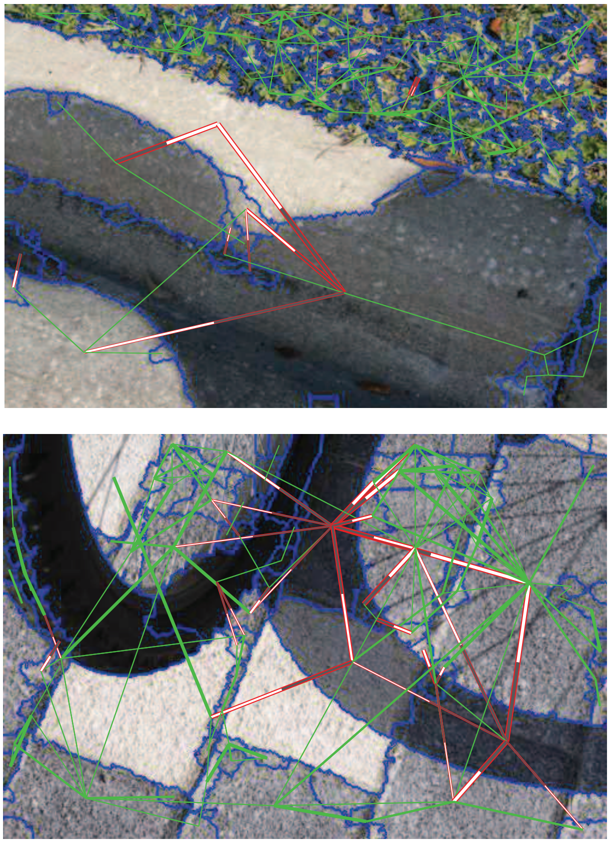


Fig. 3. Illumination relation graph of two example images. Green lines indicate *same illumination pairs*, and red/white lines mean *different illumination pairs*, where white ends are the nonshadow regions and dark ends are shadows. The width shows the confidence of the pair.

χ^2 distances between color and texture histograms are computed as in Section 2.1. We also compute normalized texton histogram, where we normalize the sum of filter responses at each pixel to 1. Regions of the same material will often have similar texture histograms, regardless of differences in shading. When regions have both similar color and texture, they are likely to be same illumination pairs.

Ratios of RGB average intensity are calculated as $(\rho_R = \frac{R_{avg1}}{R_{avg2}}, \rho_G = \frac{G_{avg1}}{G_{avg2}}, \rho_B = \frac{B_{avg1}}{B_{avg2}})$, where R_{avg1}, for example, is the average value of the red channel for the first region. For a shadow/nonshadow pair of the same material, the nonshadow region has a higher value in all three channels.

Chromatic alignment. Studies have shown that color of shadow/nonshadow pairs tend to align in RGB color space [26]. Simply put, the shadow region should not look more red or yellow than the nonshadow region since direct light usually has a higher color temperature than the ambient light (e.g., the sun is yellow and the sky is blue). This ratio is computed as ρ_R/ρ_B and ρ_G/ρ_B.

Normalized distance in position. Because distant image regions are less likely to correspond to the same material, we also add the normalized distance as a feature, computing it as the euclidean distance of the region centers divided by the square root of the geometric mean of the region areas: $\tilde{D}(R_i, R_j) = \frac{D(R_i, R_j)}{\sqrt{a_i^{1/2} a_j^{1/2}}}$.

We define c_{ij}^{same} as the log likelihood output of the classifier for same-illumination pairs times $\sqrt{a_i a_j}$, the geometric mean of the region areas. Similarly, c_{ij}^{diff} is the log likelihood output of the classifier for different-illumination pairs times $\sqrt{a_i a_j}$. Edges are weighted by region area and classifier score so that larger regions and those with more confidently predicted relations have more weight. Note that the edges in E_{diff} are directional: They encourage y_i to be shadow and y_j to be nonshadow. In the pairwise classification, each pair of regions is labeled as either different illumination, same illumination, or different material, whichever is most confident. Regions of different material pairs are not directly connected in the graph. If there are many edges, to make the inference procedure faster we include the top 100 most confident edges, which empirically yield very similar results.

2.3 Graph-Cut Inference

We can apply efficient and optimal graph-cut inference by reformulating our objective function (1) as the following energy minimization:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \sum_k cost_k^{\text{unary}}(y_k) + \alpha_2 \sum_{\{i,j\}\in E_{\text{same}}} c_{ij}^{same}\mathbf{1}(y_i \neq y_j), \tag{2}$$

with

$$\begin{aligned} cost_k^{\text{unary}}(y_k) = -c_k^{\text{shadow}} y_k - \alpha_1 \sum_{\{i=k,j\}\in E_{\text{diff}}} c_{ij}^{\text{diff}} y_k \\ + \alpha_1 \sum_{\{i,j=k\}\in E_{\text{diff}}} c_{ij}^{\text{diff}} y_k. \end{aligned} \tag{3}$$

Because this is regular (binary, with pairwise term encouraging affinity), we can solve for $\hat{\mathbf{y}}$ using graph cuts [27]. In our experiments, α_1 and α_2 are determined by cross-validation on the training set. We set $\alpha_1 = 1$ and $\alpha_2 = 2$.

3 Shadow Removal

Our shadow removal approach is based on a simple shadow model where lighting consists of single-source direct light and environment light. We try to identify how much direct light is occluded for each pixel in the image and relight the whole image using that information. First, we use a matting technique to estimate a fractional shadow coefficient value. Then, we estimate the ratio of direct to environmental light in each color channel, which, together with the shadow coefficient, enables a shadow-free image to be recovered.

3.1 Shadow Model

In our illumination model, there are two types of light sources: direct light and environment light. Direct light comes directly from the source (e.g., the sun), while environment light is from reflections of surrounding surfaces. Nonshadow areas are lit by both direct light and environment light, while for shadow areas part or all of the direct light is occluded. The shadow model can be represented by the formula below:

$$\mathbf{I}_i = (k_i \ \cos\theta_i \ \mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i, \tag{4}$$

where $\mathbf{I}_i$ is a vector representing the value for the ith pixel in RGB space. Similarly, both $\mathbf{L_d}$ and $\mathbf{L_e}$ are vectors of size 3, each representing the intensity of the direct light and environment light, also measured in RGB space. $\mathbf{R}_i$ is the surface reflectance of that pixel, also a vector of three dimensions, each corresponding to one channel. θ_i is the angle between the direct lighting direction and the surface norm, and k_i is a value between $[0, 1]$ indicating how much direct light gets to the surface. Equations (4), (5), (6), (7), (12), (14), and (15) for matrix computation refer to a pointwise computation, while (8) and (9) refer to standard matrix computation. When $k_i = 1$, the pixel is in a nonshadow area, and when $k_i = 0$, the pixel is in an umbra; otherwise, the area is in a penumbra. For an shadow-free image, every pixel is lit by both direct light and environment light and can be expressed as

$$\mathbf{I}_i^{shadow\_free} = (\mathbf{L_d}\cos\theta_i + \mathbf{L_e})\mathbf{R}_i. \tag{5}$$

To simplify the model, we assume θ is consistent across shadow/nonshadow pairs, and use $\mathbf{L_d}$ to represent $\mathbf{L_d}cos\theta_i$. Though this assumption is not always true, especially for complex scenes, experiment results show that satisfactory results can be achieved under such assumptions.

3.2 Shadow Matting

The shadow detection procedure provides us with a binary shadow mask where each pixel i is assigned a $\hat{k}_i$ value of either 1 or 0. However, in natural scenes, instead of having sharp edges, illumination often changes gradually along shadow boundaries. Also, automatic segmentation may result in inaccurate boundaries. Using detection results as shadow coefficient values in recovery can result in strong boundary effects. To get more accurate k_i values and get smooth changes between nonshadow regions and recovered shadow regions, we apply a soft matting technique.

Given an image $\mathcal{I}$, matting tries to separate the foreground image $\mathcal{F}$ and background image $\mathcal{B}$ based on the following formulation:

$$\mathbf{I}_i = \gamma_i \mathbf{F}_i + (1-\gamma_i)\mathbf{B}_i, \tag{6}$$

where $\mathbf{I}_i$ is the RGB value of the ith pixel of the original image $\mathcal{I}$, and $\mathbf{F}_i$ and $\mathbf{B}_i$ are the RGB value of the ith pixel of the foreground $\mathcal{F}$ and background image $\mathcal{B}$. By rewriting the shadow formulation given in (4) as

$$\mathbf{I_i} = k_i(\mathbf{L_d}\mathbf{R}_i + \mathbf{L_e}\mathbf{R}_i) + (1-k_i)\mathbf{L_e}\mathbf{R}_i, \tag{7}$$

an image with shadow can be seen as the linear combination of a shadow-free image $\mathbf{L_d}\mathcal{R} + \mathbf{L_e}\mathcal{R}$ and a shadow image $\mathbf{L_e}\mathcal{R}$ ($\mathcal{R}$ is a 3D matrix whose ith entry equals to $\mathbf{R}_i$), a formulation identical to that of image matting.

We employ the matting algorithm from [22], minimizing the following energy function:

$$E(\mathbf{k}) = \mathbf{k}^T\mathcal{L}\mathbf{k} + \lambda(\mathbf{k}-\hat{\mathbf{k}})^T D(\mathbf{k}-\hat{\mathbf{k}}), \tag{8}$$

where $\hat{\mathbf{k}}$ indicates the estimated shadow label (Section 2), with $\hat{k}_i = 0$ being shadow areas and $\hat{k}_i = 1$ being nonshadow. D is a diagonal matrix where $D(i,i) = 1$ when the k_i for the ith pixel needs to agree with $\hat{k}_i$ and 0 when the k_i value is to be predicted by the matting algorithm. $\mathcal{L}$ is the matting Laplacian matrix proposed in [22], aiming to enforce smoothness over local patches (in our experiments, a patch size of 3×3 is used).

To account for inaccuracies in shadow detection and to allow smooth transitions across shadow boundaries, we allow the matting algorithm to calculate the coefficient values for the majority of the pixels while requiring consistency for the rest of the pixels (we will refer to these pixels as the constraints). To generate good-quality soft shadow masks, we would like the constraints to capture the appearance variance of both the shadow and nonshadow regions, while allowing the matting algorithm to generate finer details and provide gradual change from shadow to nonshadow. To achieve such a goal, we draw inspirations from the user scribbles provided in [22] and use morphology thinning (we use Matlab's `bwmorph` function call with the parameters "thin" set to 50) to generate the skeletons for each shadow/nonshadow regions. Since pixels on boundaries between shadow and nonshadow areas are often soft shadow regions with a k_i value between 0 and 1, we first apply the erosion operation (with a 9-by-9 matrix) to the hard mask so the boundaries pixels will not become part of the constraints. An example of generated pixel constraints is shown in Fig. 4. We also experimented with other constraint selection methods; see Section 5.1 for a detailed discussion.

The optimal $\mathbf{k}$ value is the solution to the following sparse linear system:

$$(\mathcal{L} + \lambda D)\mathbf{k} = \lambda \mathbf{d}\hat{\mathbf{k}}, \tag{9}$$

where $\mathbf{d}$ is the vector comprised of elements on the diagonal of the matrix D. In our experiments, we empirically set λ to 0.01.

3.3 Ratio Calculation and Pixel Relighting

Based on our shadow model, we can relight each pixel using the calculated ratio and $\mathbf{k}$ value. The new pixel value is given by

$$\mathbf{I}_i^{shadow\_free} = (\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i \tag{10}$$

$$= (k_i\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i \frac{\mathbf{L_d} + \mathbf{L_e}}{k_i\mathbf{L_d} + \mathbf{L_e}} \tag{11}$$

$$= \frac{\mathbf{r} + 1}{k_i\mathbf{r} + 1}\mathbf{I}_i, \tag{12}$$

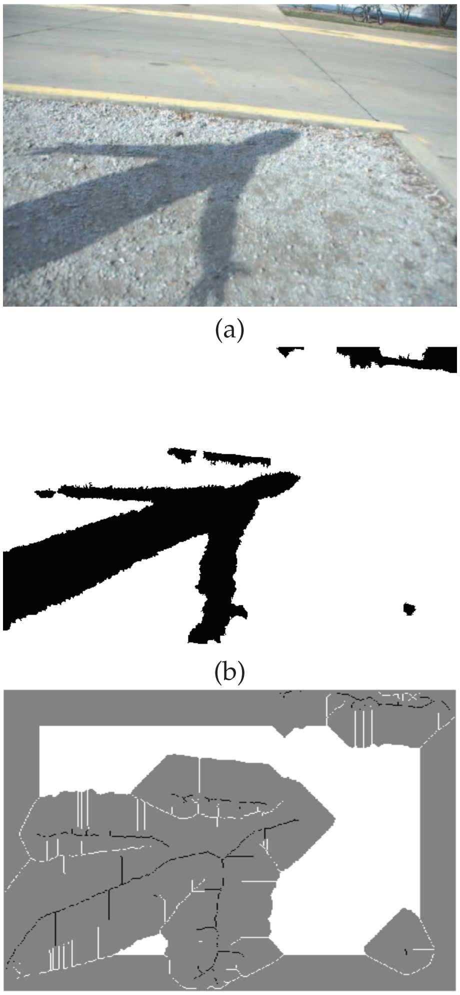
where $\mathbf{r} = \frac{\mathbf{L_d}}{\mathbf{L_e}}$ is the ratio between direct light and environment light and $\mathbf{I}_i$ is the ith pixel in the original image. For each channel, we recover the pixel value separately. We now show how to recover $\mathbf{r}$ from detected shadows and matting results.

To calculate the ratio between direct light and environment light, our model checks for pairs or regions along the shadow boundary. We believe these patches are of the same material and reflectance. We also assume direct light and environment light is consistent throughout the image. Based on the lighting model, for two pixels with the same reflectance we have

$$\mathbf{I}_i = (k_i\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i, \tag{13}$$

$$\mathbf{I}_j = (k_j\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_j, \tag{14}$$

with $\mathbf{R}_i = \mathbf{R}_j$.


(a)

(b)

(c)

Fig. 4. Generating constraints for shadow matting. (a) Original input image. (b) Detected hard shadow mask. (c) Generated constraints for matting. The gray pixels are unconstrained, the white pixels are constrained nonshadow pixels, and the black pixels are constrained shadow pixels.

From the above equations, we can arrive at

$$\mathbf{r} = \frac{\mathbf{L_d}}{\mathbf{L_e}} = \frac{\mathbf{I}_j - \mathbf{I}_i}{\mathbf{I}_i k_j - \mathbf{I}_j k_i}. \tag{15}$$

In our implementation, we uniformly sample patches from both sides of an edge between shadow/nonshadow pair, and use the average pixel values and k value in each patch as $\mathbf{I}_i$, $\mathbf{I}_j$, k_i, and k_j. To find the best ratio value and account for misdetections, we perform voting in the joint RGB ratio space with a bin size of 0.1, and the patch size is set to 12×12.

4 Experiments and Results

In our experiments, we evaluate both shadow detection and shadow removal results. For shadow detection, we evaluate how explicitly modeling the pairwise region relationship affects detection results and how well our detector can generalize cross datasets. For shadow removal, we evaluate the results quantitatively on our dataset by comparing the recovered image with the shadow-free ground truth and

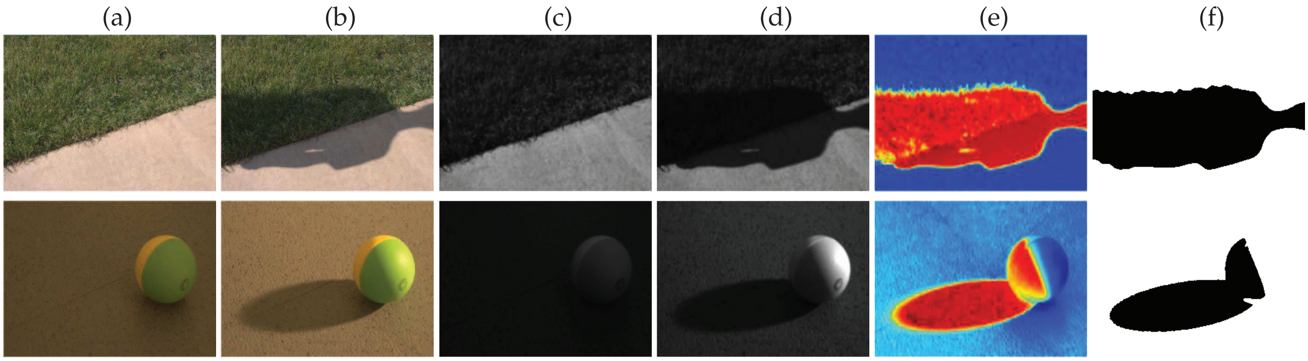


Fig. 5. Generating ground truth shadow masks. First row: Using “ground truth” image with shadow source removed. Second row: Using “ground truth” image with light source blocked. Columns: (a) shadow-free image, (b) shadow image, (c) gray-value shadow-free image, (d) gray-value shadow image, (e) heat map of difference between gray-value images, and (f) shadow mask after thresholding.

show the qualitative results on both our dataset and the UCF shadow dataset [1].

4.1 Data Set

Our shadow detection and removal methods are evaluated on the UCF shadow dataset [1] and our proposed new dataset. Zhu et al. [1] made available a set of 245 images they collected themselves and from the Internet, with manually labeled ground truth shadow masks.

Our dataset contains 108 images, each of which consists of a shadow image (input to the detection and removal algorithm) and a “ground truth” shadow-free image used to generate ground truth shadow masks. Thirty-two images are chosen as training data, and the other 76 as testing data. To account for different types of shadows, such as soft shadows and self-shadows, out of the 76 image pairs, 46 image pairs are created by removing the source of the shadow while the light source remains the same. In the remaining 30 images, the shadows are caused by objects in the scene, and the image pair is created by blocking the light source and leaving the whole scene in shadow.

We automatically generate the ground truth shadow mask by thresholding the ratio between the two images in a pair (Fig. 5). This approach is more accurate and robust than manually annotating shadow regions. To generate training data for the unary and pairwise SVM, we manually annotated shadow/nonshadow regions as well as different illumination pairs, same illumination pairs, and different material pairs. The number of annotated pairs in each image varies from 10 to 20 pairs, depending on the complexity of the scene.

4.2 Shadow Detection Evaluation

Two sets of experiments are carried out for shadow detection. First, we try to compare the performance when using only the unary classifier, only the pairwise classifier, and both combined. Second, we conduct cross dataset evaluation, training on one dataset and testing on the other. The per pixel accuracy on the testing set is reported in Table 1 and the qualitative results are shown in Fig. 6. Our quantitative results differ slightly from the conference version [2] due to the inclusion of additional features, selection of parameters via cross-validation, and the replacement of SVM classifier scores with log probability caliberated scores.

4.2.1 Comparison between Unary and Pairwise Information

Using only unary information, our performance on the UCF dataset is 87.1 percent, versus 83.4 percent achieved by classifying everything to nonshadow and 88.7 percent reported in [1]. Differently from our approach, which makes use of color information, Zhu et al. [1] conduct shadow detection on gray-scale images. By combining unary information with pairwise information, we achieve an accuracy of 90.2 percent. Note that we are using a simpler set of features and simpler learning method than [1]. The pairwise illumination relations are shown to be important, eliminating 24 percent of the pixel labeling errors on the UCF dataset and 40 percent of the errors on our own dataset. As shown by the confusion matrices, this reflects a large increase in the labeling accuracy of shadowed regions.

The pairwise illumination relations are especially important on our dataset. Using them on our dataset, the overall accuracy increases by more than 7 percent and 30 percent more shadow areas than with the single-region classifier.

4.2.2 Feature Evaluation

We examine the features used by our classifiers by looking at their influence on unary and pairwise classification (Table 2). We report equal error rate (EER), the rate at which the number of false positives equals the number of false negatives, on these tasks as a summary of performance. Table 2a shows EER with different unary features. Both color and texture cues are helpful and the classifier works better when combined.

Classification using pairwise features is summarized in Table 2b. Texture and distances are more useful on material classification, but less informative of the illumination. Color distances, ratio of RGB average, and color alignment perform strongly on illumination classification task. The confusion matrix of pairwise classification is shown on Table 2c. The most confusion comes from different illumination pairs and different material pairs since textures can look slightly different when viewed in shadow, especially the texture due to 3D geometry, e.g., little bumps on a rough surface.

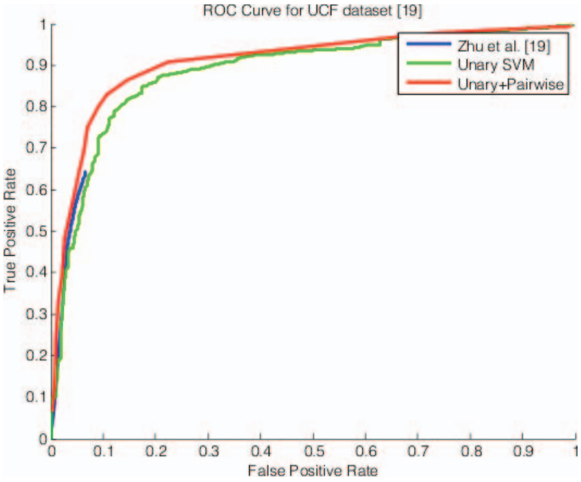
4.2.3 Cross Dataset Evaluation

The results in Table 3 indicate that our proposed detector can generalize across datasets. This is especially notable since the two datasets are very different in nature, with [1]

TABLE 1
(a) Confusion Matrices for Shadow Detection, (b) ROC Curve on the UCF Dataset, and (c) the Average per Pixel Accuracy on both Datasets

Our dataset (unary)	Shadow	Non-shadow
Shadow(GT)	0.543	0.457
Non-shadow(GT)	0.089	0.911
Our dataset (unary+pairwise)	Shadow	Non-shadow
Shadow(GT)	0.716	0.284
Non-shadow(GT)	0.048	0.952
UCF (unary)	Shadow	Non-shadow
Shadow(GT)	0.366	0.634
Non-shadow(GT)	0.027	0.973
UCF (unary+pairwise)	Shadow	Non-shadow
Shadow(GT)	0.733	0.267
Non-shadow(GT)	0.063	0.937
UCF (Zhu et al. [1])	Shadow	Non-shadow
Shadow(GT)	0.639	0.361
Non-shadow(GT)	0.067	0.934

(a) Detection confusion matrices



(b) ROC Curve on UCF dataset

	UCF shadow dataset	Our dataset
BDT+BCRF [1]	0.887	-
Our method		
Unary SVM	0.871	0.817
Pairwise SVM	0.716	0.789
Unary SVM + adjacent Pairwise	0.898	0.881
Unary SVM + Pairwise	**0.902**	**0.891**

(c) Shadow detection evaluation (per pixel accuracy)

containing more large scale scenes and hard shadows. As shown in Table 3, the unary and pairwise classifiers trained on [1] perform well on both datasets. This is understandable since their dataset is more diverse and contains more training images.

4.3 Shadow Removal Evaluation

To evaluate shadow-free image recovery, we used as measurement the root mean square error (RMSE) in L *a* b color space between the ground truth shadow free image and the recovered image, which is designed to be locally perceptually uniform. We evaluate our results on the whole image as well as shadow and nonshadow regions separately.

The quantitative evaluation is performed on the subset of images with ground truth shadow-free image (a total of 46 images). Shadow/nonshadow regions are given by the ground truth shadow mask introduced in the previous section. As shown in Table 4, our shadow removal procedure based on image matting yields results that are quantitatively close to ground truth.

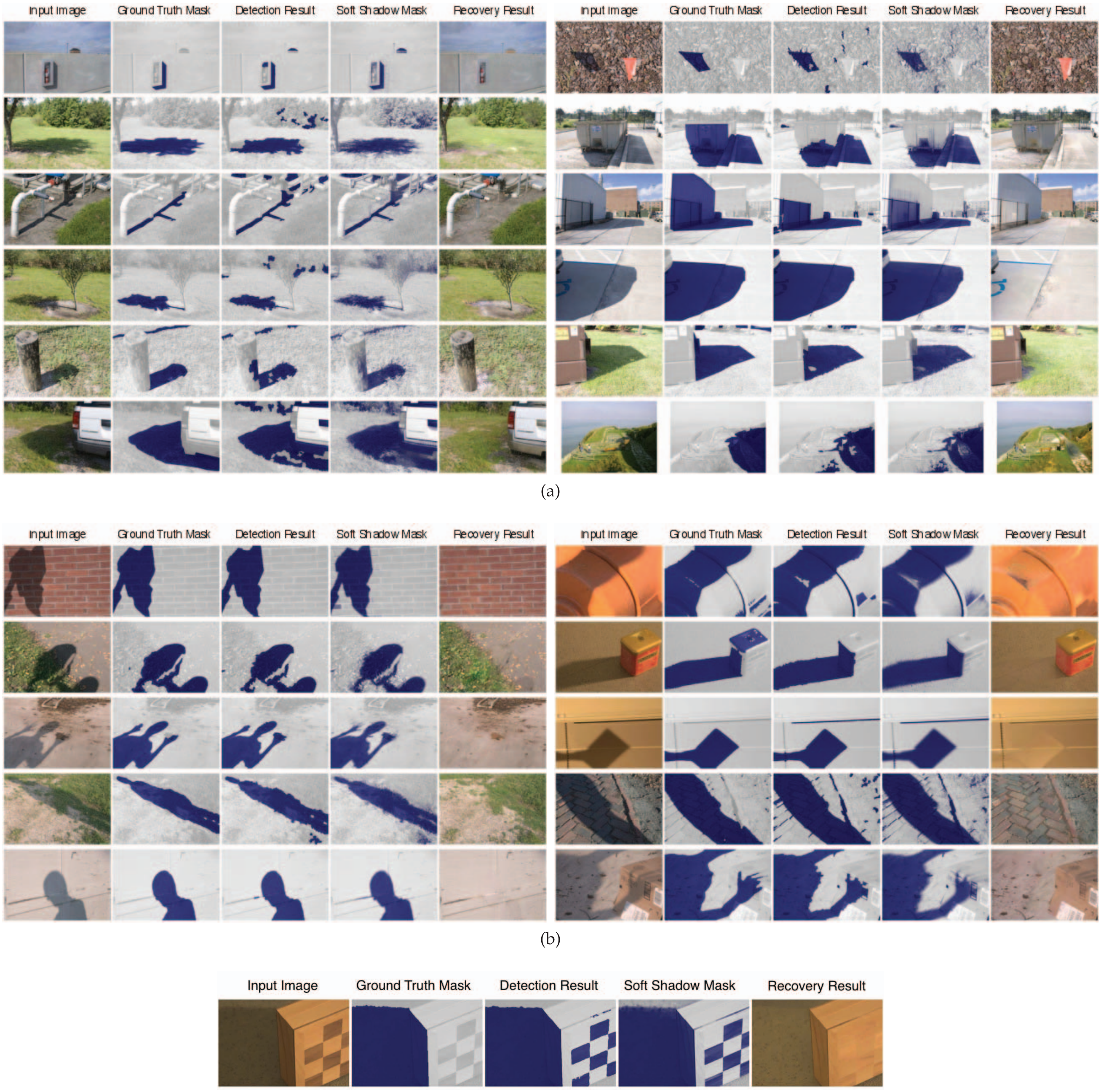
We show results overall and individually for shadow and nonshadow regions (according to the binary ground truth labels). The "nonshadow" regions may contain light shadows so that error between original and ground truth shadow-free images is not exactly zero for these regions. To show that matting helps achieve smooth boundaries, we also compare the recovery results using only the detected hard mask. We also show results using soft matte generated from the ground truth hard mask, which provides a more accurate evaluation of the recovery algorithm.

The qualitative results for shadow removal are shown in Fig. 6: Fig. 6a shows the detection and removal results on the UCF dataset [1]; Fig. 6b demonstrates results on our dataset. An interesting failure example is shown in Fig. 6c where the darker parts of the checkerboard are paired with the lighter parts by the pairwise detector and thus removed in the recovery stage.

5 Discussions

5.1 Generating Good Soft Masks

As mentioned in Section 3.2, the matting algorithm takes detected hard shadow masks as input and aims at generating soft shadows (i.e., penumbra) and accounting for minor errors in the detection step. Whether a good soft mask can be generated largely depends on the constraints (the set of supplied shadow/nonshadow pixels). Ideally, the set of constraint pixels should be dense enough to contain pixels of different colors (i.e., the green grassland as well as the blue sky) and sparse enough to allow smooth transition between shadow and nonshadow regions. We experimented with three different strategies: 1) remove a thin band around shadow boundaries (edges between shadow/nonshadow regions) and use the rest of the pixels as constraints; 2) randomly sample a subset of the pixels from each connected shadow/nonshadow region and use them as constraints (we tried sampling from two distributions, the uniform distribution and a distribution where the probability is proportional to the distance to the shadow boundary); 3) choose pixels from the skeleton of each image region as constraint. The results for two sample images are shown in Fig. 7. Experiment results show that the generated soft masks tend to resemble the hard map when more pixel labels are given as constraints, and will better reflect the gradient/illumination of the original image (i.e., image with shadow) when fewer constraints are imposed. As a result,



(c)

Fig. 6. (a) Detection and recovery results on the UCF dataset [1]. These results show that our detection and recovery framework also works well in complicated scenes. Notice in (a) that the cast shadows of the trees are correctly recovered, even though they are complicated and soft. (b) Detection and recovery results on our dataset. (c) Failed example. The darker parts of the chessboard are mistakenly detected as shadow and, as a result, removed in the recovery process.

method (a) works best when the detection results are accurate and when the input image contains complicated scenes. However, the generated soft masks often fail to capture finer grained shadow, such as the ones cast by trees. Although methods (b) and (c) often reflect the details in the original image, they sometimes mistake shading and surface texture near shadow boundaries for shadows. All our experiment results are generated using method (c).

5.2 Shadow Detection in Scene-Scale Images

One interesting question is how well our shadow detection algorithm works for general scenes which are often more complicated and sometimes cluttered with different objects. To answer that question, we applied our shadow detector a randomly sampled subset of the Pascal VOC 2010 [28] trainval set of 100 images. We used the model trained on the UCF dataset [1] because their dataset contains more scene scale images and thus more closely resembles the Pascal dataset. Selected results are shown in Fig. 8 containing both success and failure cases.

Based on the generated results, we can see that our shadow detector is generally good at detecting cast shadows, regardless of the material of the surface. Though we do not have examples of animals or people in our training set, the shadow detector succeeds in spotting the

TABLE 2
(a) Equal Error Rate (EER) of Unary Features of Shadow/Nonshadow Classification Task;
(b) EER of Pairwise Features on Annotated Pairs on the UCF Dataset [1]; the Results Are Reported Using Fivefold Cross-Validation on the Training Set (c) Confusion Matrix of Pairwise Classification, Also from Cross-Validation

Feature	EER
Texton Histogram	0.807
LAB Color Histogram	0.697
Both	0.828

(a) Unary feature evaluation on UCF dataset

Task	Same/Different Material		Same/Different Illumination	
Feature name	With only	Everything except	With only	Everything except
χ^2 Texture distance	0.696	0.832	0.735	0.976
χ^2 Color distance	0.679	0.832	0.963	0.964
RGB average	0.683	0.799	0.969	0.962
Normalized distance	0.706	0.809	0.490	0.976
Color alignment	0.671	0.827	0.831	0.966
All	0.836	-	0.976	-

(b) Pairwise Feature Evaluation on UCF Dataset

Pairwise Relation	Diff. Illu	Diff. Illu. (Rev)	Same Illu.	Diff. Material
Diff. Illu	0.643	0.014	0.003	0.040
Diff. Illu. (Rev)	0.018	0.631	0.004	0.043
Same Illu.	0.024	0.022	0.893	0.051
Diff. Material	0.315	0.333	0.101	0.866

(c) Confusion matrix of pariwise classification on UCF dataset

"Different illumination (Rev)" indicates the different illumination pair where the shadow/nonshadow relation is switched.

TABLE 3
Cross Dataset Tasks, Training the Detector on One Dataset and Testing It on the Other One

Training source	pixel accuracy on UCF dataset	pixel accuracy on our dataset
Unary UCF	0.871	0.755
Unary UCF,Pairwise UCF	**0.902**	0.815
Unary UCF,Pairwise Ours	0.890	0.863
Unary Ours	0.689	0.817
Unary Ours,Pairwise UCF	0.792	0.870
Unary Ours,Pairwise Ours	0.791	**0.898**

TABLE 4
The Per Pixel RMSE for Shadow Removal Task

Region Type	Original	No matting	Automatic matting	Matting with Ground Truth Hard Mask
Overall	13.7	8.2	7.4	6.4
Shadow regions	42.0	16.7	13.9	11.8
Non-shadow regions	4.6	5.4	5.4	4.7

The first column shows the error when no recovery is performed; the second column is when detected shadow masks are directly used for recovery and no matting is applied; the third column is the result of using soft shadow masks generated by matting; the last column shows the result of using soft shadow masks generated from ground truth mask.

shadows cast on animal skin (first row, last pair) and human clothes (second row, second pair). However, as suggested by the failure cases, the detector often mistakes darker image regions for shadows. This is probably due to the bias toward dark shadows in the Zhu dataset and could be solved by introducing diversity to the training data. Also, the shadow detector often incorrectly finds shadows in overcast or diffusely lit scenes.

5.3 Applications

Our algorithm can be used to recover scene illumination for applications of image manipulation. Karsch et al. [4] used our shadow detection algorithm [2] to find confident highlight regions, or light shafts. They first use the algorithm to determine a soft shadow mask of regions that are not illuminated by the light shafts and then take the inverse. They then use the estimated scene geometry to recover the direction of the shafts. This automatically estimated illumination information provides an alternative to user annotation and is further used to realistically render synthetic objects in existing images, as shown in Fig. 9c. The soft shadow matte can be used for compositing shadows into other images [20]. In addition, with the shadow matte, we can remove direct light and render the whole scene in shadow, as in Fig. 9f.

5.4 Limitations and Future Work

Our algorithm on shadow detection and removal has a number of restrictions. We cannot differentiate between shading differences due to surface orientation changes and due to cast shadows. Also, in our shadow removal procedure, we are implicitly making the assumption that all surfaces that contain shadows should be roughly planar and parallel to each other. However, our detection method

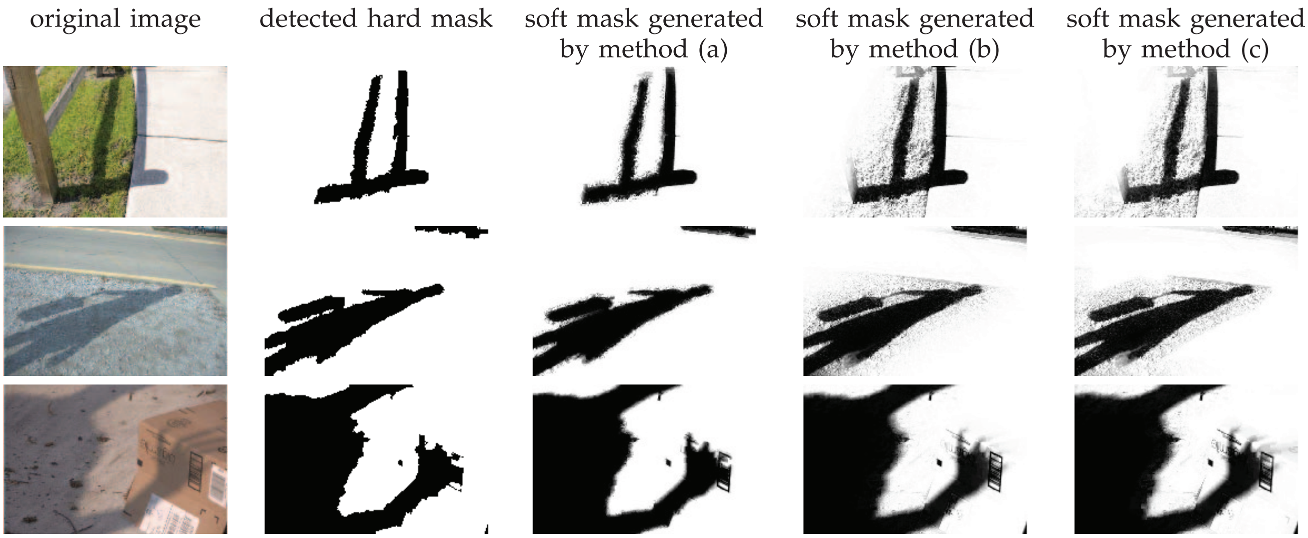

original image
detected hard mask
soft mask generated by method (a)
soft mask generated by method (b)
soft mask generated by method (c)

Fig. 7. Soft shadow masks generated using different constraints. Left to right: Original image with shadow, hard shadow mask, soft shadow mask generated using methods (a), (b), and (c). For method (b), we sample 10 percent of the pixels using the weighted distribution. First row: The case where method (a) works the best while both methods (b) and (c) include the texture of the grass as shadows. Second row: The case where methods (b) and (c) correctly capture the detection error (the thin line-shaped shadow) while mistaking part of the ground texture as shadow. In both the first two rows, method (b) tends to include more texture as shadow. Last row: Methods (b) and (c) correct the detection error by capturing part of the shadows cast by the fingers, with method (c) capturing more of it.

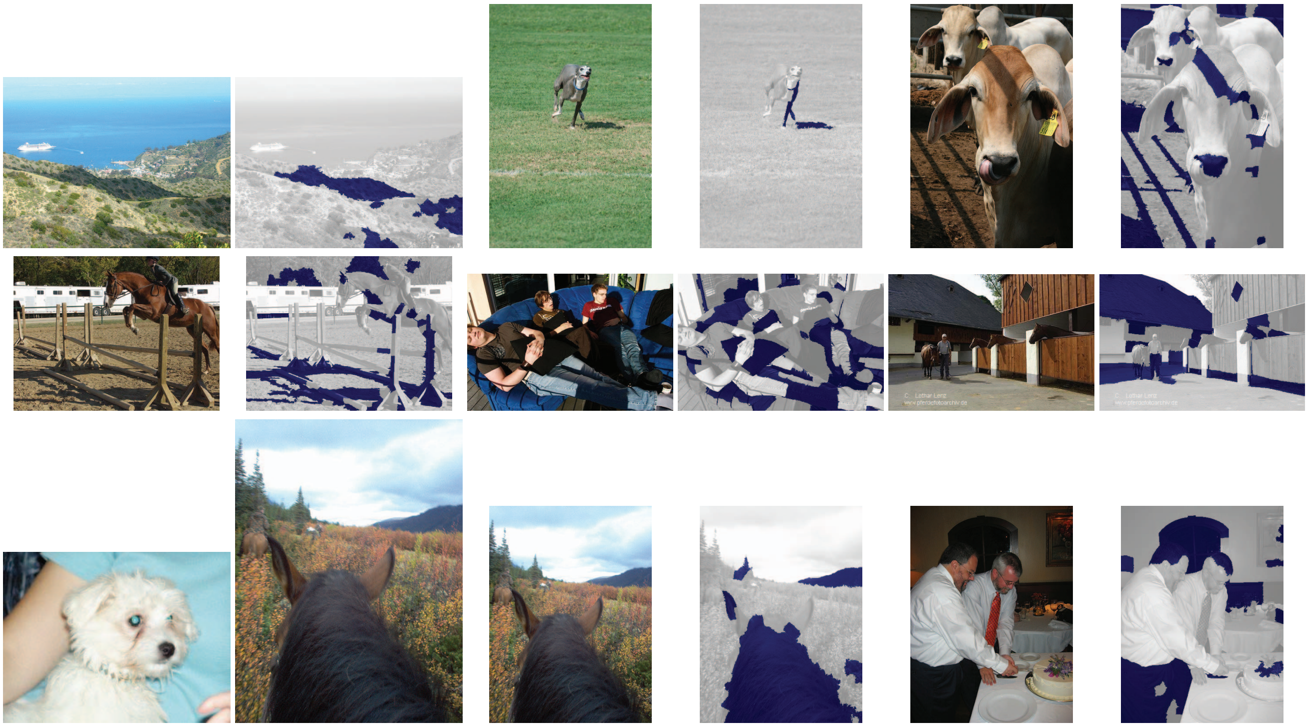


Fig. 8. Shadow detection results on the Pascal VOC 2010 trainval set. From top to bottom: Generally successful/mixed/failed detection examples. For each image pair, the left one is the input image and the right one is the detected shadow mask.

does not need this constraint. Currently, our detection method relies on the initial segmentation, which may group soft shadows with nonshadow regions. Also, the detection algorithm may fail in the case of multiple light sources.

We could further improve detection by incorporating more sophisticated features, such as the set of unary region features introduced in [1]. We can also incorporate geometry and shadow boundary estimates into our detection framework, as in [12]. This could possibly remove false pairings between regions and provide valuable hints regarding the source of the shadow. We have made our dataset and code available,[1] which we hope will facilitate the use of shadow detection and removal in scene understanding.

5.5 Conclusion

In conclusion, we proposed a novel approach to detect and remove shadows from a single still image. For shadow detection, we have shown that pairwise relationships

[1] http://www.cs.illinois.edu/homes/guo29/projects/shadow.html.

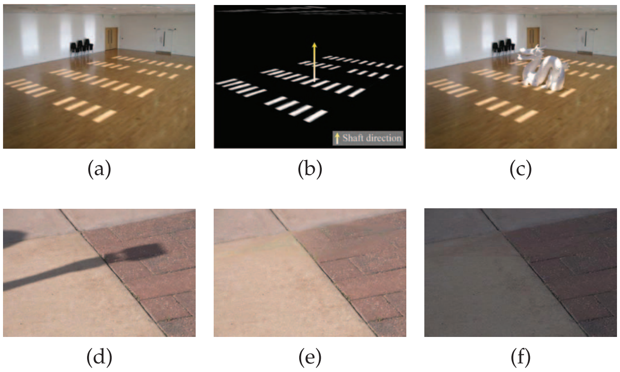


Fig. 9. Application of illumination manipulations. First row: Inserting synthetic objects into legacy photos by Karsch et al. [4]. (a) The original image. (b) Light shaft detection and shaft direction estimation. (c) Insertion of objects with realistic illumination with consistent illumination. Second row: Rendering the whole scene in shadow. (d) The original image. (e) Remove the shadow; all pixels relit with both direct and ambient light. (f) Remove the direct light component of the scene by relighting all pixels with only ambient light.

between regions provides valuable additional information about the illumination condition of regions, compared with simple appearance-based models. We also show that by applying soft matting to the detection results, the lighting conditions for each pixel in the image are better reflected, especially for those pixels on the boundary of shadow areas. Our conclusions are supported by quantitative experiments on shadow detection and removal in Tables 1 and 4.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) under IIS-0904209 and a Google Research Award. The authors would like to thank David Forsyth for useful comments and Kevin Karsch for the images of Fig. 9c.

References

[1] J. Zhu, K.G.G. Samuel, S. Masood, and M.F. Tappen, "Learning to Recognize Shadows in Monochromatic Natural Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[2] R. Guo, Q. Dai, and D. Hoiem, "Single-Image Shadow Detection and Removal Using Paired Regions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2011.

[3] J.-F. Lalonde, A.A. Efros, and S.G. Narasimhan, "Estimating Natural Illumination from a Single Outdoor Image," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[4] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem, "Rendering Synthetic Objects Into Legacy Photographs," *Proc. ACM Siggraph,* 2011.

[5] D.L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," technical report, 1972.

[6] H. Barrow and J. Tenenbaum, "Recovering Intrinsic Scene Characteristics from Images," *Computer Vision Systems,* pp. 3-26, 1978.

[7] E.H. Land and J.J. McCann, "Lightness and Retinex Theory," *J. Optical Soc. of Am.,* vol. 61, pp. 1-11, 1971.

[8] B.A. Maxwell, R.M. Friedhoff, and C.A. Smith, "A Bi-Illuminant Dichromatic Reflection Model for Understanding Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[9] S.G. Narasimhan, V. Ramesh, and S.K. Nayar, "A Class of Photometric Invariants: Separating Material from Shape and Illumination," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* 2003.

[10] G.D. Finlayson, S.D. Hordley, C. Lu, and M.S. Drew, "On the Removal of Shadows from Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 59-68, Jan. 2006.

[11] G.D. Finlayson, M.S. Drew, and C. Lu, "Entropy Minimization for Shadow Removal," *Int'l J. Computer Vision,* vol. 85, no. 1, pp. 35-57, 2009.

[12] J.-F. Lalonde, A.A. Efros, and S.G. Narasimhan, "Detecting Ground Shadows in Outdoor Consumer Photographs," *Proc. 11th European Conf. Computer Vision,* 2010.

[13] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios, "Illumination Estimation and Cast Shadow Detection through a Higher-Order Graphical Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2011.

[14] M.H.V. Kwatra and S. Dai, "Shadow Removal for Aerial Imagery by Information Theoretic Intrinsic Image Analysis," *Proc. IEEE Int'l Conf. Computational Photography,* 2012.

[15] G.D. Finlayson, S.D. Hordley, and M.S. Drew, "Removing Shadows from Images Using Retinex," *Proc. Color Imaging Conf.,* 2002.

[16] C. Fredembach and G.D. Finlayson, "Fast Re-Integration of Shadow Free Images," *Proc. Color Imaging Conf.,* 2004.

[17] C. Fredembach and G. Finlayson, *Proc. British Machine Vision Conf.,* 2005.

[18] E. Arbel and H. Hel-Or, "Shadow Removal Using Intensity Surfaces and Texture Anchor Points," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 6, pp. 1202-1216, June 2011.

[19] E. Arbel and H. Hel-Or, "Texture-Preserving Shadow Removal in Color Images Containing Curved Surfaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[20] Y.-Y. Chuang, D.B. Goldman, B. Curless, D. Salesin, and R. Szeliski, "Shadow Matting and Compositing," *ACM Trans. Graphics,* vol. 22, no. 3, pp. 494-500, 2003.

[21] T.-P. Wu, C.-K. Tang, M.S. Brown, and H.-Y. Shum, "Natural Shadow Matting," *ACM Trans. Graphics,* vol. 26, no. 2, 2007.

[22] A. Levin, D. Lischinski, and Y. Weiss, "A Closed-Form Solution to Natural Image Matting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, pp. 228-242, Feb. 2008.

[23] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[24] D.R. Martin, C. Fowlkes, and J. Malik, "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 530-549, May 2004.

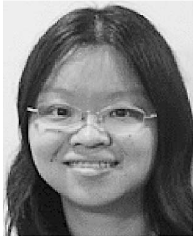
[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intelligent Systems and Technology,* vol. 2, pp. 27:1-27:27, 2011.

[26] M. Baba and N. Asada, "Shadow Removal from a Real Picture," *Proc. ACM Siggraph,* 2003.

[27] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 65-81, Feb. 2004.

[28] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2012.

Ruiqi Guo received the BE degree in computer science from Zhejiang University, China, in 2009. He is currently working toward the PhD degree in the Computer Science Department at the University of Illinois at Urbana-Champaign, advised by Derek Hoiem. Before that, he was an exchange student at the School of Engineering, Tohoku University, Japan, from 2007 to 2008. His research interests include computer vision and machine learning, with a focus on visual scene understanding. He is a student member of the IEEE.



Qieyun Dai received the BEng degree from Nanjing University, China. She is currently working toward the PhD degree in the Department of Computer Science at the University of Illinois at Urbana-Champaign (UIUC), advised by Derek Hoiem. Her research interests include computer vision and machine learning. She is a student member of the IEEE.

Derek Hoiem received the PhD degree in robotics from Carnegie Mellon University in 2007, advised by Alexei A. Efros and Martial Hebert. He joined the University of Illinois at Urbana-Champaign (UIUC) faculty in 2009. He was a postdoctoral fellow at the Beckman Institute from 2007 to 2008. He is an assistant professor in computer science at UIUC. His research on scene understanding and object recognition has been recognized with a 2006 CVPR Best Paper award, a 2008 ACM Doctoral Dissertation Award honorable mention, and a 2011 US National Science Foundation (NSF) CAREER award. He is a member of the IEEE.