Detecting ground shadows in outdoor consumer photographs

Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan

School of Computer Science, Carnegie Mellon University
Project webpage: http://graphics.cs.cmu.edu/projects/shadows

Abstract. Detecting shadows from images can significantly improve the performance of several vision tasks such as object detection and tracking. Recent approaches have mainly used illumination invariants which can fail severely when the qualities of the images are not very good, as is the case for most consumer-grade photographs, like those on Google or Flickr. We present a practical algorithm to automatically detect shadows cast by objects onto the ground, from a single consumer photograph. Our key hypothesis is that the types of materials constituting the ground in outdoor scenes is relatively limited, most commonly including asphalt, brick, stone, mud, grass, concrete, etc. As a result, the appearances of shadows on the ground are not as widely varying as general shadows and thus, can be learned from a labelled set of images. Our detector consists of a three-tier process including (a) training a decision tree classifier on a set of shadow sensitive features computed around each image edge, (b) a CRF-based optimization to group detected shadow edges to generate coherent shadow contours, and (c) incorporating any existing classifier that is specifically trained to detect grounds in images. Our results demonstrate good detection accuracy (85%) on several challenging images. Since most objects of interest to vision applications (like pedestrians, vehicles, signs) are attached to the ground, we believe that our detector can find wide applicability.

1 Introduction

Shadows are everywhere! Yet, the human visual system is so adept at filtering them out, that we never give shadows a second thought; that is until we need to deal with them in our algorithms. Since the very beginning of computer vision, the presence of shadows has been responsible for wreaking havoc on a wide variety of applications, including segmentation, object detection, scene analysis, stereo, tracking, etc. On the other hand, shadows play a crucial role in determining the type of illumination in the scene [1, 2] and the shapes of objects that cast them [3]. But while standard approaches, software, and evaluation datasets exist for a wide range of important vision tasks, from edge detection to face recognition, there has been comparatively little work on shadows in the last 40 years. Approaches that use multiple images [4], time-lapse image sequences [5, 6] or user inputs [7–9] have demonstrated impressive results, but detecting shadows reliably and automatically from a single image remains an open problem. This is because the appearances and shapes of shadows outdoors depend on several hidden factors such as the color, direction and size of the illuminants (sun, sky, clouds), the geometry of the objects that are casting the shadows and the shape and material properties of objects onto which the shadows are cast.

Most works for detecting shadows from a single image are based on computing illumination invariants that are physically-based and are functions of individual pixel values [10–14] or the values in a local image neighborhood [15]. Unfortunately, reliable computations of these invariants require high quality images with wide dynamic range, high intensity resolution and where the camera radiometry and color transformations are accurately measured and compensated for. Even slight perturbations (imperfections) in such images can cause the invariants to fail severely (see Fig. 4). Thus, they are ill-suited for the regular consumer-grade photographs such as those from Flickr and Google, that are noisy and often contain compression, resizing and aliasing artifacts, and effects due to automatic gain control and color balancing. Since much of current computer vision research is done on consumer photographs (and even worse-quality photos from the mobile phones), there is an acute need for a shadow detector that could work on such images.

Our goal is to build a reliable shadow detector for consumer photographs of outdoor scenes. While detecting all shadows is expected to remain hard, we explicitly focus on the shadows cast by objects onto the ground plane. Fortunately, the types of materials constituting the ground in typical outdoor scenes are (relatively) limited, most commonly including concrete, asphalt, grass, mud, stone, brick, etc. Given this observation, our key hypothesis is that the appearances of shadows on the ground are not as widely varying as the shadows everywhere in the scene and can be learned from a set of labelled images of real world scenes. This restriction by no means makes the problem trivial: the ground shadow detector still needs to contend with myriad other non-shadow visual manifestations such as markings and potholes on the roads, pavement/road boundaries, grass patterns on lawns, etc. Further, since many objects (pedestrians, vehicles, traffic signs, etc) of interest to vision applications, are attached to the ground and cast shadows onto the ground, we believe such a ground shadow detector will find wide applicability.

1.1 Overview

Our approach consists of three stages depending on the information in the image used. In the first stage, we will exploit local information around edges in the image. For this, we compute a set of shadow sensitive features that include the ratios of brightness and color filter responses at different scales and orientations on both sides of the edge. These features are then used with a trained decision tree classifier to detect whether an edge is a shadow or not. The idea is that while any single feature may not be useful for detecting all ground shadows, the classifier is powerful enough to choose the right features depending on the underlying edge region. In order to make the classifier robust to non-shadow edges, a negative training set is constructed from a set of edges not on the ground and those arising due to road markings, potholes, grass/mud boundaries, etc. Surprisingly, this simple procedure yields 80% classification accuracy on our test set of images randomly chosen from Flickr and LabelMe [16].

In the second stage, we enforce a grouping of the shadow edges using a Conditional Random Field (CRF) to create longer contours. This is similar in spirit to the classical constrained label propagation used in mid-level vision tasks [17]. This procedure connects likely shadow edges, discourages T-junctions which are highly unlikely on shadow boundaries, and removes isolated weak edges. But how do we detect the ground in an image? For this, in the third stage, we incorporate a global scene layout descriptor within our CRF, such as the 3-way ground-vertical surface-sky classifier by Hoiem et. al [18]. Since the scene layout classifier is trained on the general features of the scene and not the shadows, we are able to reduce the number of false-positive (non-shadow) detections outside the ground. Our results show that the shadow detection results improve by 5% with this step.

We demonstrate successful shadow detection on several images of natural scenes that include beaches, meadows and forest trails, as well as urban scenes that include numerous pedestrians, vehicles, trees, roads and buildings, captured under a variety of illumination conditions (sunny, partly cloudy, overcast). Similarly to the approach of Zhu et al. [19], our method relies on learning the appearance of shadows based on image features, but does so by using full color information. We found that using color features and incorporating knowledge of the ground location improve classification results as much as 10% on our test set. While our technique can be used as a stand-alone shadow detector, we believe it can also be tightly integrated into higher level scene understanding tasks.

2 Learning local cues for shadow detection

Our approach relies on a classifier which is trained to recognize ground shadow edges by using features computed over a local neighborhood around the edge. We show that it is indeed possible to obtain good classification accuracy by relying on local cues, and that it can be used as a building block for subsequent steps. In this section, we describe how to build, train, and evaluate such a classifier.

2.1 From pixels to boundaries

We first describe the underlying representation on which we compute features. Since working with individual pixels is prone to noise and computationally expensive, we propose to instead reason about *boundaries*, or groups of pixels along an edge in the image. To obtain these boundaries, we first smooth the image with a bilateral filter [20], compute gradient magnitudes on the filtered image, and then apply the watershed segmentation algorithm on the gradient map. Fig. 1(b) shows a close-up example of such boundaries.

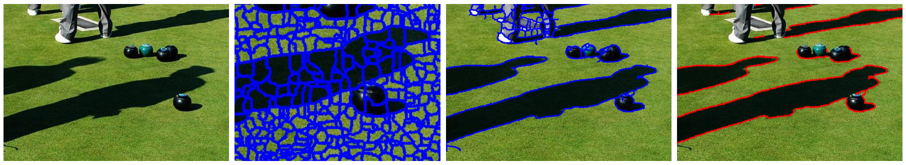

(a) Input image (b) Boundaries (c) Strong boundaries (d) Output

Fig. 1. Processing stages for the local classifier. The input image (a) is over-segmented into thousands of regions to obtain boundaries (b). Weak boundaries are filtered out by a Canny edge detector (c), and the classifier is applied on the remainder. (d) shows the boundaries i for which $P(y_i = 1|\mathbf{x}) > 0.5$. Note the correct classification of occlusion contours around the person's legs and the reflectance edges in the white square between the person's feet.

An undesirable consequence of the watershed segmentation is that it generates boundaries in smooth regions of the image (Fig. 1(b)). To compensate for this, we retain only those boundaries which align with the strong edges in the image. For this, we use the canny edge detector at 4 scales to account for blurry shadow edges ($\sigma^2 = \{1, 2, 4, 8\}$), with a high threshold empirically set to $t = 0.3$. Under these conditions, we verified that the initial set of boundaries contain more than 97% of the true shadow edges in our dataset. For example, Fig. 1(c) shows the set of boundaries on which our classifier is evaluated for that image.

2.2 Local shadow features

We now describe the features computed over each boundary in the image. A useful feature to describe a shadow edge is the ratio of color intensities on both sides of the edge (e.g. min divided by max) [21]. The intuition is that shadows should have a specific ratio that is more or less the same across an image, since it is primarily due to the differences in natural lighting inside and outside the shadow. Since it is hard to manually determine the best color space [22] or best scale to compute features, we use 3 different colors spaces (RGB, LAB, [23]) and 4 different scales, and let the classifier automatically select the relevant features during the training phase. Although color-based features are bound to be affected by camera non-linearities, we found these ratios to work well across a wide range of cameras and capture conditions.

For a pixel along a boundary, we compute the intensity on one side of the edge (say, the left) by evaluating a weighted sum of pixels on the left of the edge. But which pixels to choose? We could use the watershed segments, but they do not typically extend very far. Instead, we use an oriented gaussian derivative filter of variance σ^2, but keep only its values which are greater than zero. We align the filter with the boundary orientation such that its positive weights lie on the left of the boundary and convolve it with the image to obtain f_l. The same operation is repeated with the filter rotated by 180° to obtain the weighted mean of pixels on its right f_r. Color ratios can then be computed at pixel p by

$\frac{\min(f_l(p), f_r(p))}{\max(f_l(p), f_r(p))}$. This is done independently for each color channel of the RGB, LAB, and illumination-invariant [23] color spaces. To account for edge sharpness, we compute each filter at 4 different scales $\sigma^2 = \{1, 2, 4, 8\}$ and size $2\sigma^2$, to obtain 36 ratios in total.

We also employ two features suggested in [19] which capture the texture and intensity distribution differences on both sides of a boundary. The first feature computes a histogram of textons at 4 different scales, and compares them using the χ^2-distance. The texton dictionary was computed on a non-overlapping set of images. The second feature computes the difference in skewness of pixel intensities, again at the same 4 scales.

Finally, we concatenate the absolute value of the minimum filter response $\min(f_l(p), f_r(p))$ computed over the intensity channel to obtain the final, over-complete, 48-dimensional feature vector at every pixel. Boundary feature vectors are obtained by averaging the features of all pixels that belong to it.

2.3 Classifier

Having computed the feature vector $\mathbf{x}_i$ at each strong boundary in the image, we can now use them to train a classifier to learn the probability $P(y_i|\mathbf{x}_i)$ that boundary i is due to a shadow (which we denote with label y_i). We estimate that distribution using a logistic regression version of Adaboost [24], with twenty 16-node decision trees as weak learners. This classification method provides good feature selection and outputs probabilities, and has been successfully used in a variety of other vision tasks [18, 25].

To train the classifier, we selected 170 images from LabelMe [16], Flickr, and the dataset introduced in [19], with the only conditions being that the ground must be visible, and there must be shadows. The positive training set contains manually labelled shadow boundaries, while the negative training set is populated with an equal amount of strong non-shadow boundaries on the ground (e.g. street markings) and occlusion boundaries.

We obtain a per-boundary classification accuracy of 79.7% (chance is 50%, see Fig. 5 for a breakdown per class). See Fig. 1(d) for an example. This result support out hypothesis: while the appearance of shadows on *any* type of material in *any* condition might be impossible to learn, the space of shadow appearances on the ground in outdoor scenes may not be that large after all!

3 Creating shadow contours

Despite encouraging results, our classifier is limited by its locality since it treats each boundary independently of the next. However, the color ratios of a shadow boundary should be consistent with those of its neighbors, since the sources illuminating nearby scene points should also be similar. Thus, we can exploit higher order dependencies across local boundaries to create longer shadow contours as well as remove isolated/spurious ones.

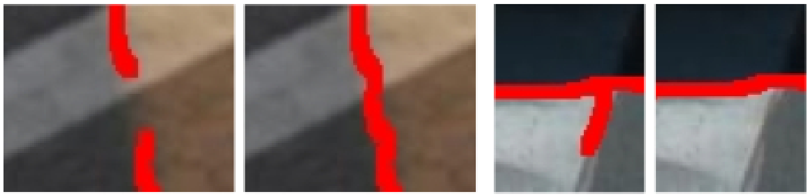

(a) (b)

Fig. 2. Creating shadow contours by enforcing local consistency. Our CRF formulation may help to (a) bridge the gap across X-junctions where the local shadow classifier might be uncertain, and (b) remove spurious T-junctions which should not be caused by shadows.

To model these dependencies, we construct a graph with individual boundaries as nodes (such as those in Fig. 1(b)) and drawing an edge across boundaries which meet at a junction point. We then define a CRF on that graph, which expresses the log-likelihood of a particular labeling $\mathbf{y}$ (i.e. assignment of shadow/non-shadow to each boundary) given observed data $\mathbf{x}$ as a sum of unary $\phi_i(y_i)$ and pairwise potentials $\psi_{i,j}(y_i, y_j)$:

$$-\log P(\mathbf{y}|\mathbf{x};\lambda,\beta) = \lambda \sum_{i\in\mathcal{B}} \phi_i(y_i) + \sum_{(i,j)\in\mathcal{E}} \psi_{i,j}(y_i, y_j) - \log Z_{\lambda,\beta} \ , \tag{1}$$

where $\mathcal{B}$ is the set of boundaries, $\mathcal{E}$ the set of edges between them, and λ and β are model parameters. In particular, λ is a weight controlling the relative importance of the two terms. $Z_{\lambda,\beta}$ is the partition function that depends on the parameters λ and β, but not on the labeling $\mathbf{y}$ itself.

Intuitively, we would like the unary potentials to penalize the assignment of the "shadow" label to boundaries which are not likely to be shadows according to our local classifier. This can be modeled using

$$\phi_i(y_i) = -\log P(y_i|\mathbf{x}_i) \ . \tag{2}$$

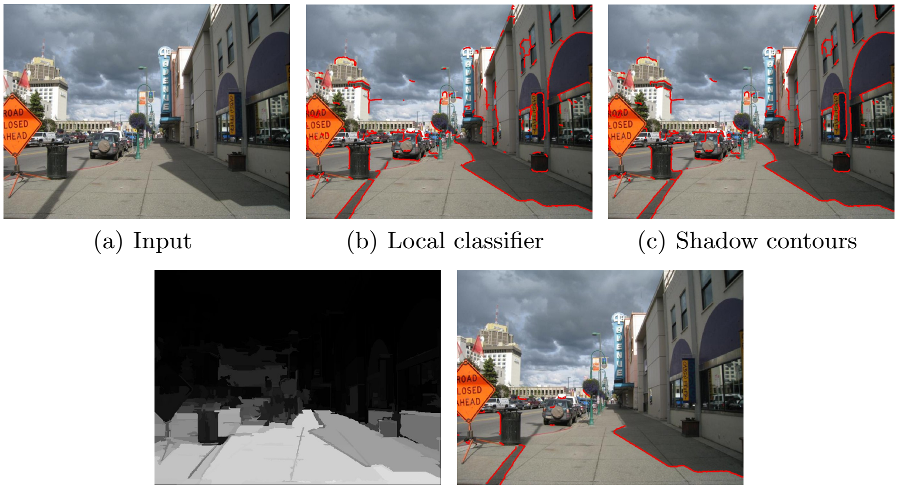
We would also like the pairwise potentials to penalize the assignment of different labels to neighboring boundaries that have similar features, which can be written as

$$\psi_{i,j}(y_i, y_j) = \mathbf{1}(y_i \neq y_j) \exp(-\beta \|\mathbf{x}_i - \mathbf{x}_j\|_2^2) \ , \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function, and β is a contrast-normalization constant as suggested in [26]. In other words, we encourage neighboring shadows which have similar features and strong local probabilities to be labelled as shadows.

The negative likelihood in (1) can be efficiently minimized using graph cuts [27–29]. The free parameters were assigned the values of $\lambda = 0.5$ and $\beta = 16$ obtained by 2-fold cross-validation on a non-overlapping set of images.

Applying the CRF on our test images results in an improvement of roughly 1% in total classification accuracy, for a combined score of 80.5% (see Fig. 5-(b)). But more importantly, in practice, the way the CRF is setup encourages



(a) Input (b) Local classifier (c) Shadow contours

(d) Ground likelihood [18] (e) Combining (c) and (d)

Fig. 3. Incorporating scene layout for detecting cast shadows on the ground. Applying our shadow detector on a complex input image (a) yields false detections in the vertical structures because of complex effects like occlusion boundaries, self-shadowing, etc. (b) & (c). Recent work in scene layout extraction from single images [18] can be used to estimate the location of the ground pixels (d). We show how we can combine scene layout information with our shadow contour classifier to automatically detect cast shadows on the ground (e).

continuity, crossing through X-junctions, and discourages T-junctions as shown in Fig. 2. Since shadows are usually signaled by the presence of X-junctions and the absence of T-junctions [30], this reduces the number of false positives.

4 Incorporating scene layout

Until now, we have been considering the problem of detecting cast shadow boundaries on the ground with a classifier trained on local features and a CRF formulation which defines pairwise constraints across neighboring boundaries. While both approaches provide good classification accuracy, we show in Fig. 3 that applying them on the entire image generates false positives in the vertical structures of the scene. Reflections, transparency, occlusion boundaries, self-shadowing, and complex geometry [30] are common phenomena that can confuse our classifier. This results in image-wide classification results which might not be useful for complex scenes (see Figs. 3(b)-(c)).

The advent of recent approaches which estimate a qualitative layout of the scene from a single image (e.g. splitting an image into three main geometric classes: the sky, vertical surfaces, and ground [18]) may provide explicit knowledge of where the ground is. Since such a scene layout estimator is specifically trained on general features of the scene and not the shadows, combining its output with our shadow detector should reduce the number of false positive (non-shadow) detections outside the ground. We now consider how such high-level scene reasoning can be used within our shadow detection framework.

4.1 Combining scene layout with local shadow cues

To combine the scene layout probabilities with our local shadow classifier, we can marginalize the probability of shadows over the three geometric classes sky $\mathcal{S}$, ground $\mathcal{G}$, and vertical surfaces $\mathcal{V}$:

$$P_{comb}(y_i|\mathbf{x}) = \sum_{c \in \{\mathcal{G},\mathcal{V},\mathcal{S}\}} P(y_i|c_i, \mathbf{x}_i) P(c_i|\mathbf{x}_i) \ , \tag{4}$$

where c_i is the geometric class label of boundary i, $P(y_i|c_i, \mathbf{x}_i)$ is given by our local shadow classifier, and $P(c_i|\mathbf{x}_i)$ by the scene layout classifier (we use the geometric context algorithm [18]). Unfortunately, this approach does not actually improve classification results because while it gets rid of false positives in the vertical structures, it also loses true positives on the ground along the way. This is due to the fact that shadow likelihoods get down-weighted by low-confidence ground likelihoods. Thus, we need a different approach.

4.2 Combining scene layout with shadow contours

Intuitively, we would like to penalize an assignment to the shadow class when the probability of being on the ground is low. When it is high, however, we should let the shadow classifier decide. We can encode this behavior simply by modifying the unary potentials $\phi_i(y_i)$ from (2) in our CRF formulation:

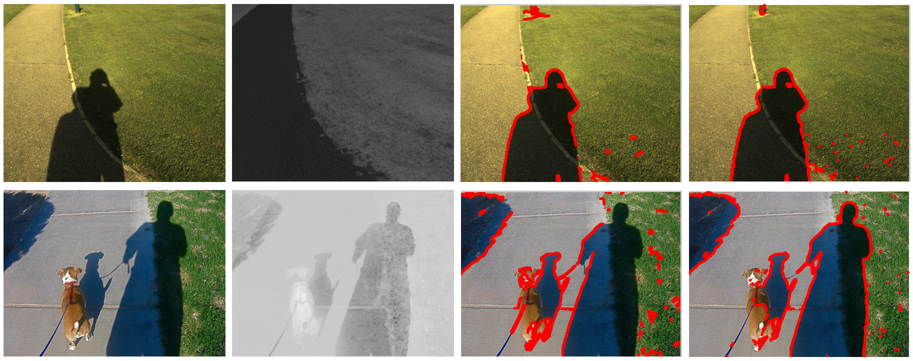
$$\phi_i(y_i) = \begin{cases} -\log P(c_i = \mathcal{G}|\mathbf{x}_i) - \log P(y_i = 1|\mathbf{x}_i) & \text{if } y_i = 1 \text{ (shadow)} \\ (1 - P(c_i = \mathcal{G}|\mathbf{x}_i)) - \log P(y_i = 0|\mathbf{x}_i) & \text{if } y_i = 0 \text{ (non-shadow)} \end{cases} . \tag{5}$$

Here, $\lambda = 0.5$ and $\beta = 16$ was found by cross-validation. They yield a good compromise between local evidence and smoothness constraints.

This approach effectively combines local and mid-level shadow cues with high-level scene interpretation results, and yields an overall classification accuracy of 84.8% on our test set (see Fig. 5) without adding to the complexity of training our model. Observe how the results are significantly improved in Fig. 3(e) as compared to the other scenarios in Fig. 3(b)-(c).

5 Experimental results

We evaluate our approach on 135 consumer photographs downloaded from LabelMe [16], Flickr, and images from the dataset introduced in [19]. In all cases, we have no control over the acquisition settings, so images contain the typical



(a) Original image (b) Invariant img. [12] (c) Result with [12] (d) Our result

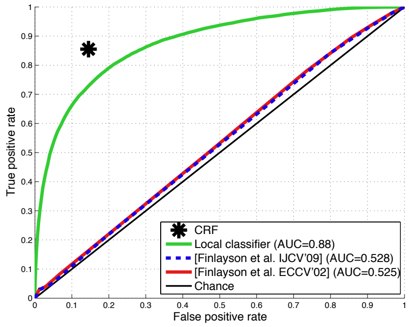
Fig. 4. Comparison with the shadow detection method of Finlayson et al. [12]. *First row*: Using a high-quality linear image as input (a), our implementation of their method successfully recovers a shadow-free 1-D invariant image (b), which is used to detect shadow edges (c). *Second row*: However, if the input is not linear and corrupted by noise or jpeg compression typical of consumer photographs, the 1-D invariant image still contains some shadows (b), making it hard to tell them apart from other types of edges (c). Our method detects shadows both in high and low quality images (d).

sources of distortions [31] such as jpeg compression, sharpening, sampling due to resizing, non-linear response functions, image noise, etc. We first compare our method with the current state of the art [12], then show shadow detection and removal results on several challenging images.

5.1 Comparison with previous work

The current state of the art in color-based shadow detection and removal in single images is the approach of Finlayson et al. [12], which relies on a physics-based model of shadows to compute an illumination-invariant image. Shadows are then obtained by finding edges in the original image which are not present in the invariant image. The first row of Fig. 4 shows that our implementation of their approach successfully recovers a shadow-free invariant image from the same high-quality, linear image used in their original paper [10]. Note that our approach is able to extract similar results to theirs (Fig. 4(d)). When applying their method on an image from our set, the performance degrades because the invariant still contains strong traces of shadows (second row of Fig. 4).

This is also demonstrated in the quantitative comparison shown in Fig. 5-(a), which compares our approach with the original shadow detection technique of [11], and the most recent version [12]. Note here that we cannot generate an ROC curve with the results of our CRF formulation, since the output is binary, so we plot a point at its classification accuracy. To obtain ROC scores for the competing methods, we first estimate the invariant image, and compute the difference of gradient magnitudes between the original and the invariant



(a) ROC curve comparison with previous work

	Shadows	Non-shadows	Combined
Local	78.3%	81.0%	79.7%
CRF	78.7%	82.3%	80.5%
CRF + scene layout	73.1%	96.4%	**84.8%**

(b) Quantitative ground shadow classification results

Fig. 5. Quantitative results. We compare our local classifier with the methods proposed by Finlayson et al. [11, 12] (a). The table in (b) show the results obtained with the approaches presented in Sects. 2 (local), 3 (CRF) and 4 (CRF + scene layout). Integrating scene layout information from [18] results in ground shadow classification accuracy of 84.8%.

images. For fairness, we evaluate this score only at the strong boundaries in the image. The ROC curve shows that our results greatly outperform the previous work. This is most likely due to the use of features which are robust to artifacts common in consumer photographs.

5.2 Ground shadow detection and removal

We summarize the quantitative results obtained by the technique presented in this paper on our test set in Fig. 5-(b), which shows results obtained on the entire image by the local classifier and the boundary CRF (Sects. 2 and 3), and those obtained by combining the geometric context ground likelihoods (Sect. 4). The best performance (84.8% accuracy) is obtained by our CRF formulation which combines the scene layout results with our local shadow boundary classifier.

Fig. 6 shows ground shadow detection results on several images from our dataset. It demonstrates that our method works on challenging outdoor images with varying illumination conditions, ground colors and textures, and clutter.

The typical errors made by our method are shown in Fig. 7. It may fail to detect shadows cast by thin structures like the lamppost in Fig. 7(a). Another failure case arises when the ground has a color that is vastly different from all the other images in the training set, as in Fig. 7(b). This can likely be improved by increasing the size and diversity of the training set. A third failure mode is due to errors in the estimated scene layout probabilities as in Fig. 7(c).

Once we have detected shadow boundaries, we can, as an application, use the technique introduced in [10] to remove them and recover a shadow-free image. There have been improvements proposed since then [32], but we chose the original method for its simplicity. This approach involves setting the derivatives of the image at shadow boundaries to zero, and reintegrating the result by solving the

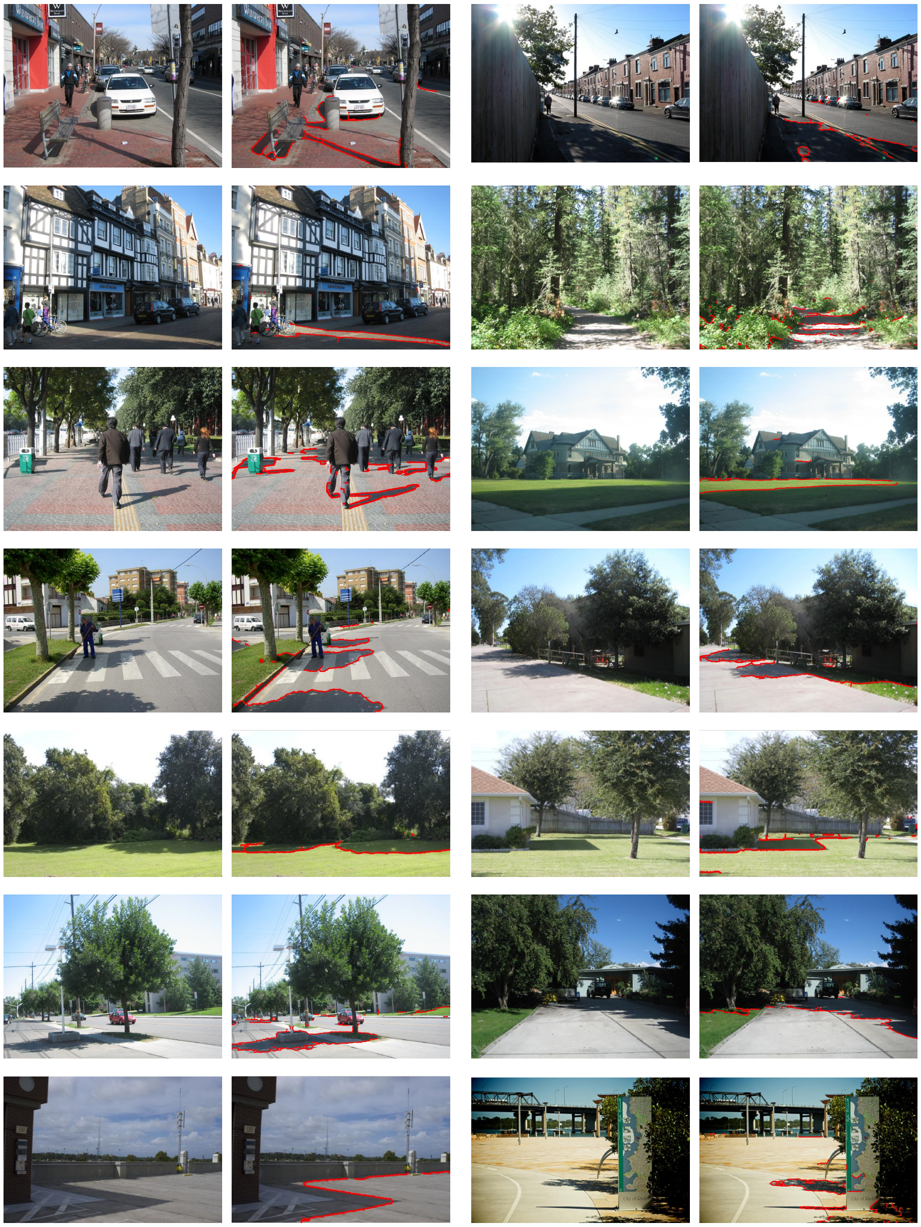
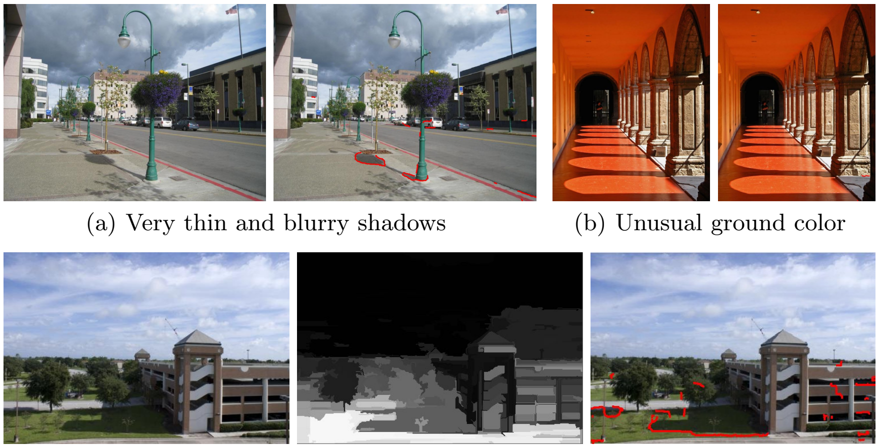


Fig. 6. Ground shadow detection results on images downloaded from the web (Flickr, LabelMe [16]), and the dataset from [19]. First and third columns: input images; second and fourth columns: detected ground shadows. Our approach successfully detects ground shadows in many challenging, real-world conditions.



(a) Very thin and blurry shadows

(b) Unusual ground color

(c) Ground estimation errors

Fig. 7. Failure cases. The downside of using boundaries from an over-segmentation is the trade-off between spatial support obtained from longer boundaries and the size of shadow regions that can be detected. In our current setting, it may miss thin and blurry edges, like the lamppost in (a). Our approach is also sensitive to vastly different ground colors, which have never been seen by the classifier (b). Although our ratio-based features are somewhat color independent, they are not able to compensate for such drastic differences. Increasing the variety of ground colors in the training set would likely improve performance on such extreme cases. (c) Errors in the scene layout probabilities can lead to false positives.

Poisson equation with Neumann boundary conditions. Fig. 8 shows shadow-free images that were computed using the boundaries detected by our method.

5.3 Future work

While our technique detects ground shadows with good accuracy, shadows that are not on the ground exhibit significantly larger appearance variations, so detecting them will be challenging. While our technique can be used as a standalone shadow detector, we believe it can also be tightly integrated into higher level scene understanding tasks. For example, the presence of an object implies that a shadow should be nearby, and vice versa. We will be pursuing these research avenues as future work.

Acknowledgements

This work has been partially supported by a Microsoft Research Graduate Fellowship to J.-F. Lalonde, an Okawa Foundation Grant, ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, as well as NSF CAREER IIS-0643628 and IIS-0546547. We would like to thank Andrew Stein, Derek Hoiem, and Olga Veksler for making their code available online.

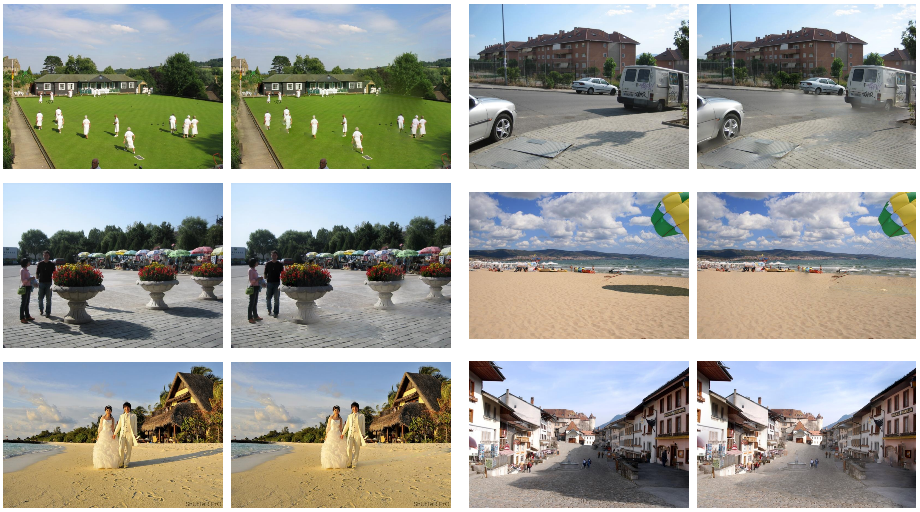


Fig. 8. Automatic ground shadow removal from a single image. We apply the original gradient reintegration method of [10] on the boundaries detected by our method. The shadows are either completely removed or greatly attenuated, with few visual artifacts.

References

1. Lalonde, J.F., Efros, A.A., Narasimhan, S.G.: Illumination estimation from a single outdoor image. In: IEEE International Conference on Computer Vision. (2009)
2. Sato, I., Sato, Y., Ikeuchi, K.: Illumination from shadows. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003)
3. Matsushita, Y., Nishino, K., Ikeuchi, K., Sakauchi, M.: Illumination normalization with time-dependent intrinsic images for video surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004)
4. Finlayson, G.D., Fredembach, C., Drew, M.S.: Detecting illumination in images. In: IEEE International Conference on Computer Vision. (2007)
5. Weiss, Y.: Deriving intrinsic images from image sequences. In: IEEE International Conference on Computer Vision. (2001)
6. Huerta, I., Holte, M., Moeslund, T., Gonzàlez, J.: Detection and removal of chromatic moving shadows in surveillance scenarios. In: IEEE International Conference on Computer Vision. (2009)
7. Wu, T.P., Tang, C.K.: A bayesian approach for shadow extraction from a single image. In: IEEE International Conference on Computer Vision. (2005)
8. Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. ACM Transactions on Graphics (SIGGRAPH Asia 2009) **28** (2009)
9. Shor, Y., Lischinski, D.: The shadow meets the mask: pyramid-based shadow removal. Computer Graphics Forum Journal (Eurographics 2008) **27** (2008)
10. Finlayson, G.D., Hordley, S.D., Drew, M.S.: Removing shadows from images. In: European Conference on Computer Vision. (2002)
11. Finlayson, G.D., Drew, M.S., Lu, C.: Intrinsic images by entropy minimization. In: European Conference on Computer Vision. (2004)

12. Finlayson, G.D., Drew, M.S., Lu, C.: Entropy minimization for shadow removal. International Journal of Computer Vision **85** (2009)
13. Maxwell, B.A., Friedhoff, R.M., Smith, C.A.: A bi-illuminant dichromatic reflection model for understanding images. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008)
14. Tian, J., Sun, J., Tang, Y.: Tricolor attenuation model for shadow detection. IEEE Transactions on Image Processing **18** (2009)
15. Narasimhan, S.G., Ramesh, V., Nayar, S.K.: A class of photometric invariants: Separating material from shape and illumination. In: IEEE International Conference on Computer Vision. (2005)
16. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision **77** (2008)
17. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. International Journal of Computer Vision **40** (2000)
18. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. International Journal of Computer Vision **75** (2007)
19. Zhu, J., Samuel, K.G.G., Masood, S.Z., Tappen, M.F.: Learning to recognize shadows in monochromatic natural images. In: IEEE Conference on Computer Vision and Pattern Recognition. (2010)
20. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the 6th International Conference on Computer Vision. (1998)
21. Barnard, K., Finlayson, G.D.: Shadow identification using colour ratios. In: Proc. IS&T/SID 8th Color Imaging Conf. Color Science, Systems and Applications. (2000)
22. Khan, E.A., Reinhard, E.: Evaluation of color spaces for edge classification in outdoor scenes. In: IEEE International Conference on Image Processing. (2005)
23. Chong, H.Y., Gortler, S.J., Zickler, T.: A perception-based color space for illumination-invariant image processing. ACM Transactions on Graphics (SIGGRAPH 2008) (2008)
24. Collins, M., Shapire, R., Singer, Y.: Logistic regression, adaboost and Bregman distances. Machine Learning **48** (2002)
25. Hoiem, D., Stein, A., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: IEEE International Conference on Computer Vision. (2007)
26. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: IEEE International Conference on Computer Vision. (2001)
27. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. IEEE transactions on Pattern Analysis and Machine Intelligence **20** (2001)
28. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004)
29. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE transaction on Pattern Analysis and Machine Intelligence **26** (2004)
30. Sinha, P., Adelson, E.H.: Recovering reflectance and illumination in a world of painted polyhedra. In: IEEE International Conference on Computer Vision. (1993)
31. Chakrabarti, A., Scharstein, D., Zickler, T.: An empirical camera model for internet color vision. In: British Machine Vision Conference. (2009)
32. Fredembach, C., Finlayson, G.D.: Simple shadow removal. In: International Conference on Pattern Recognition. (2006)