Learning to Recognize Shadows in Monochromatic Natural Images

Jiejie Zhu, Kegan G.G. Samuel, Syed Z. Masood, Marshall F. Tappen
University of Central Florida
School of Electrical Engineering and Computer Science, Orlando, FL
jjzhu,kegan,smasood,mtappen@eecs.ucf.edu

Abstract

This paper addresses the problem of recognizing shadows from monochromatic natural images. Without chromatic information, shadow classification is very challenging because the invariant color cues are unavailable. Natural scenes make this problem even harder because of ambiguity from many near black objects. We propose to use both shadow-variant and shadow-invariant cues from illumination, textural and odd order derivative characteristics. Such features are used to train a classifier from boosting a decision tree and integrated into a Conditional random Field, which can enforce local consistency over pixel labels. The proposed approach is evaluated using both qualitative and quantitative results based on a novel database of hand-labeled shadows. Our results show shadowed areas of an image can be identified using proposed monochromatic cues.

1. Introduction

Shadows are one of the most noticeable effects on a scene's illumination. While they can provide useful cues regarding scene properties such as object size, shape, and movement [14], they can also complicate recognition tasks, such as feature detection, object recognition and scene parsing. In recent years, multiple groups have proposed methods on removing the effects of illumination from an image [4, 32, 27, 24] and have also proposed approaches that specifically focus on removing shadows [8, 2, 25, 3].

In systems that focus on shadows, color is the primary cue used to identify the shadow. Finlayson *et al*. [8] located the shadows using an invariant color model. Shor and Lischinski [25] propagated the shadows using a color-based region growing method. Salvador *et al*. [23] used invariant color features to segment cast shadows. Levine [15] and Arévalo [3] both studied the color ratios across boundaries to assist shadow recognition. A number of other approaches have also focused on shadow detection using color-related motion cues [21, 19, 31, 16, 12].

These color-assisted systems rely on the assumption that the chromatic appearance of image regions does not change across shadow boundaries, while the intensity component of a pixel's color does. Such approaches work well given color images where the surfaces are still discernible inside the shadow.

(a) Diffuse (b) Specularity (c) Self-shading

Figure 1. Ambiguity of shadow recognition in monochromatic domain. The diffuse object (swing) in (a) and the specular object (car) in (b) both have a dark albedo. The trees in the aerial image (c) appear black because of self-shading. Such objects are very difficult to be separated from shadows which are also relatively dark.

In this paper, we focus on detecting shadows in a single monochromatic image. This problem is challenging because images in the monochromatic domain tend to have many objects appear black or near black (see in Figure 1). These objects complicate shadow recognition as shadows are expected to be relatively dark. In addition, natural scenes make it even harder because of their complex illumination conditions. We must be very conservative when labeling a pixel as shadow in these situations, because non-shadow pixels are similar to shadow pixels if chromatic cues are unavailable.

Our work on this problem is motivated by two main factors. The first motivation is that color may not be available in all types of sensors, such as sensors tuned to specific spectra, sensors designed for specific usages such as aerial, satellite, celestial images and images captured from very high dynamic scenes. Our second motivation is to understand how monochromatic cues can be explored and used to recognize shadows. While chromatic variations have been shown to aid in the recognition of shadows, humans are still able to find shadows in monochromatic images [10]. Working with monochromatic images makes it necessary to investigate the usual characteristics of shadows.

We present a learning-based approach for detecting shadows in a single monochromatic image. Our approach is novel, with respect to other shadow detection and removal methods, in that our system is trained and evaluated on a database of natural images and uses both shadow-variant and shadow-invariant cues.

This data-driven approach is unique because previous learning-based approaches have been evaluated on limited datasets, such as highways, parking lots or indoor office environments [20]. Gray-scale cues have also been proposed to remove general illumination effects [4, 32, 27], but these systems are limited by their reliance on either synthetically generated training data or data from specialized environments, such as the crumpled paper images from [29]. In addition, these systems mainly focus on the variant cues; it will be shown that improved recognition rates can be achieved by combining them with invariant cues.

Our approach is based on boosted decision tree classifiers and integrated into a Conditional Random Field (CRF). To make it possible to learn the CRF parameters, we use a Markov Random Field (MRF) model designed for labeling [28]. The various design decisions in this system, such as optimization method and feature choice, will be evaluated as part of the experiments. Our results show that shadows in monochromatic images can be found with color-free cues.

The contributions of this paper include: (1) an approach for shadow recognition in monochromatic natural images; (2) several useful features to locate shadows from natural scenes; (3) a database of monochromatic shadows available to the public. We also present a convenient implementation of parallel training to efficiently learn the CRF parameters.

1.1. Monochromatic Shadow Dataset

We have constructed a database consisting of 245 images. In each image, the shadows have been hand-labeled at the pixel level with two people validating the labeled shadows. In this database, 117 of the images were collected by the authors in a variety of outdoor environments, such as on campus and downtown areas, *etc*. To ensure a wide variety of lighting conditions, we also collect images at different times throughout the day. The dataset includes additional 74 257x257 sized aerial images from the Overhead Imagery Research Dataset (OIRDS) [26] and another 54 640x480 sized images from LabelMe dataset [22]. Figure 2 shows several images with shadows from the dataset.

The camera we used to capture part of the images in the database is a Canon Digital Rebel XTi. The camera model used for the shadow images from the Labelme database and from the OIRDS database are not reported.

2. Feature Extraction

Given a single monochromatic natural image, we would like to identify those pixels that are associated with shadows. Without color information, the features in our experiments are chosen to identify illumination, textural and odd order derivative characteristics. Rather than using pixel values alone, we also include the features from homogeneous regions found by over-segmentation using intensity values [17]. To faithfully capture the cues across shadow boundaries, we gather statistics [6] of neighboring pairs of shadow/non-shadow segments from all individual images in the dataset. These statistics are represented as the histograms from shadow and nonshadow segments given equal bin centers.

Figure 2. Example of shadows in the dataset. The images captured by the authors are linearly transformed from raw format to TIF format. The images from LableMe are in JPG format. The images from OIRDS are in TIF format. All the color images are converted to gray scale images.

We propose three types of features: *shadow-variant* features that describe different characteristics in shadows and in non-shadows; *shadow-invariant* features that exhibit similar behaviors across shadow boundaries and *near black* features that distinguish shadows from near black pixels. Our motivation for combining both variant and invariant features is because strong predictions of shadows are observed when these complimentary cues are used together. For example, if the segment has invariant texture with its neighbor but has variant intensities, it is more likely to be in shadows.

Each of the features introduced is characterized by scalar values that provide information relevant to shadow properties.

2.1. Shadow-Variant Features

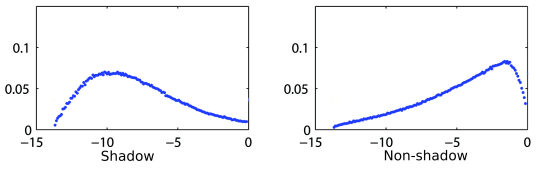
Intensity Difference Since shadows are expected to be relatively dark, we gather statistics (Figure 3 (a)) about the intensity of image segments. In neighboring pixels, we measure the intensity difference using their absolute difference. In neighboring segments, we measure the difference using L_1 norm between the histograms of intensity values. We also augment the feature vector with the averaged intensity value and the standard deviation.

Local Max In a local patch, shadows have values that are very low in intensity; therefore, the local max value is expected to be small. On the contrary, non-shadows often have values with high intensities and the local max value is expected to be large (Figure 3 (b)). We capture this cue by a local max completed at 3 pixel intervals.

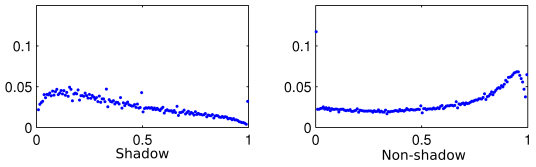
Smoothness Shadows are often a smoothed version of their neighbors. This is because shadows tend to suppress local variations on the underlining surfaces. We use the method proposed by Forsyth and Fleck in [9, 18] to cap-

ture this cue. The method subtracts a smoothed version of the image from the original version. Already smooth areas will have small differences where as highly varied areas will have large differences. To measure the smoothness, we use the standard deviations from neighboring segments.

Skewness We gathered several statistical variables (standard deviation, skewnees and kurtosis), and found a mean value of 1.77 for shadows and -0.77 for non-shadows in skewness. This tells us that the asymmetries in shadows and in non-shadows are different, which is a good cue for locating shadows. This odd order statistic is also found to be useful in extracting reflectance and gloss from natural scenes [1].



(a) Log Illumination



(b) Local Max

Figure 3. Mean Histograms of log illumination and local max. The histograms are generated from neighboring pairs of shadow/non-shadow segments using equal bin centers. The number of bins is set to 150. By gathering the histograms from all individual images in the dataset, we plot their mean histogram in this figure.

2.2. Shadow-Invariant Features

Gradient Similarity In the image-formation model underlying this work, we assume that transforming the image with a pixel-wise log transformation makes the shadow an additive offset to the pixel values in the scene. This leads us to expect that the distribution of image gradient values will often be invariant across shadow boundaries.

To capture this cue, we measure the similarity between the distributions of a set of first order derivative of Gaussian filters in neighboring segments of the image. This similarity is computed using the L_1 norm of the difference between histograms of gradient values from neighboring segments.

Texture Similarity We have observed that the textural properties of surfaces are changed little across shadow boundaries. We measure the textural properties of an image region using the method introduced in [17]. The method filters a database of images with a bank of Gaussian derivative filters consisting of 8 orientations and 3 scales and then apply clustering to form 128 discrete centers. Given a new image, the texon is assigned as the histograms binned at these discrete centers. We also measure the similarity using the L_1 norm of the difference between histograms of texton values from neighboring segments.

As can be seen in Figure 5 (f), where the color indicates the texton index selected at each point, the distributions of textons inside and outside of the shadow are similar. The primary difference is that distortion artifacts in the darker portions of the image lead to a slight increase in the number of lower-index textons, indicated by more blue pixels.

2.3. Near Black Features

As shown in Figure 3, the pixel intensity is a good indicator of the presence of a shadow, as shadows are usually dark. Unfortunately, this heuristic alone cannot be used to reliably identify shadows as it will fail in the presence of dark objects. In our experiments, we found that objects with a dark albedo, are some of the most difficult image regions to separate from shadows. Trees are also difficult because of the complex self-shading caused by the leaves. Due to the complexity of hand-labeling, the self-shading within a tree is not considered to be a shadow. We refer to these objects as *near black objects* and show an example in Figure 1.

To correctly distinguish shadows from near black objects, we introduce two additional features.

Discrete Entropy The first cue we found is that shadows have a different entropy value compared to that of near black objects. We compute the entropy using

$$E_i = \sum_{i \in \omega} -p_i \times \log_2(p_i) \qquad (1)$$

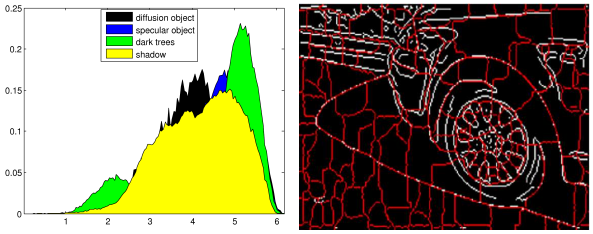
where ω is a 3x3 window, p_i is the probability of the histogram counts at pixel i. We plot the distributions of the entropies in shadows and entropies near black objects in Figure 4 (a). It shows that the entropy of diffuse objects with dark albedo (in black) is relatively small. This is because most black objects are textureless, which is also true in most natural scenes.

The entropy of the specular object (in blue) with dark albedo and the entropy of the shadows (in yellow) both have a mediate value, but appear slightly different at their peaks.

The entropy of the trees (in green) has a large value because of high local variances from self-shading. We also notice that trees have small entropies. We believe this is because some parts of trees are over exposed to the sun, therefore they appear textureless.

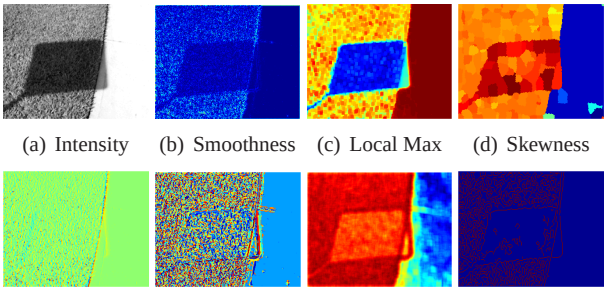
Edge Response Another feature we found important is edge response. Because shadows stifle the edge strong responses, edge responses are often small in shadows. Figure 4 (b) shows such an example where segments in shadows have near zero edge response while that of specular object (the body of the car) have a strong edge response. We compute this cue by summing up edge responses inside a segment.

We take a typical scene and visualize all 8 proposed features in Figure 5.



(a) Entropy (b) Segment-based edge response

Figure 4. Two cues used to distinguish shadows from near black objects. The distributions in (a) are obtained from 40 images including near black objects along with shadows from our dataset. In (b), the edge map is showed in white. The segmentation boundaries are shown in red.



(a) Intensity (b) Smoothness (c) Local Max (d) Skewness

(e) Gradient (f) Texton (g) Entropy (h) Edge Map

Figure 5. Feature visualization. In color figures, pixels in blue have small values, while pixels in red have large values.

3. Learning to Recognize Shadows

From the dataset introduced in Section 1.1, we randomly select 125 images as training data, the remaining 120 images are used as test cases. We pose the shadow detection problem as a per-pixel labeling task, where every pixel is classified as either being inside a shadow or not.

Our goal is to estimate a label for each pixel in an image that denotes whether that pixel belongs to shadows or not. We treat this problem as a binary classification problem, making $\mathbf{y}_i \in \{-1, 1\}$ for all pixels i.

3.1. Learning Binary CRF (BCRF)

We use a pairwise CRF to model this labeling problem. In the standard pairwise CRF model, the conditional distribution over labels $\mathbf{l}$, given an input image $\mathbf{o}$, has a form of

$$P(\mathbf{l}|\mathbf{o}) = \frac{1}{Z}\prod_i \phi_i(l_i|o_i) \prod_{<i,j>} \psi_{i,j}(l_i, l_j|o_i) \qquad (2)$$

$\prod_{<i,j>}$ denotes the product of all pairs of neighboring pixels, both horizontal and vertical. The constant Z is the normalizing constant of the distribution.

We include CRFs because they provide an elegant means for enforcing local consistency and smoothness. Ideally, the parameters of this CRF could be found by maximizing the likelihood of our hand-labeled shadow masks. Unfortunately, the size of the label images and the presence of loops in the CRF make it intractable to compute the expectation computations, which is necessary for computing the gradient of the likelihood function. A number of alternate means for discriminatively training CRF models have been proposed recently, such as [11, 30, 28].

We apply the logistic version of CRF model [28], which builds on quadratic models and logistic functions, to implement the same propagation behavior that we desired from the traditional CRF model, while also making it possible to learn the model parameters.

3.2. Configuring BCRF to Learn Shadows

The BCRF model discriminatively estimates the marginal distribution over each pixel's label. Essentially, a Gaussian CRF model is used to estimate a "response image" which is passed, in a pixel-wise fashion through a logistic function. Formally, the likelihood of pixel i taking the label $+1$, which we will denote as $\mathbf{l}_i$, is expressed as

$$\begin{aligned} \mathbf{l}_i &= \sigma(\mathbf{r}_i^*) \\ \text{where } \mathbf{r}^* &= \arg\min_{\mathbf{r}} C(\mathbf{r}; \mathbf{o}) \end{aligned} \qquad (3)$$

$\sigma(\cdot)$ is a logistic function, and $C(\mathbf{r}; \mathbf{o})$ is a quadratic function that captures the same types of propagation behavior desired from the CRF model. A similar model was also proposed in [7].

The cost function $C(\mathbf{r}; \mathbf{o})$ is based upon interactions between the expected response $\mathbf{r}$ and the current observations $\mathbf{o}$. It expresses the relationship between responses in a neighborhood as well. To recognize shadows, we define $C(\mathbf{r}; \mathbf{o})$ as

$$\begin{aligned} C(\mathbf{r}; \mathbf{o}) = \sum_i w(\mathbf{o}; \theta)((r_i - 10)^2 + (r_i + 10)^2 + \\ \sum_{<i,j>} (r_i - r_j)^2) \end{aligned} \qquad (4)$$

Each term r_i refers to the entry pixel i in the response image $\mathbf{r}$. These two terms pull each pixel to either 10 or +10 in the response image $\mathbf{r}_i^*$. The probability is transformed using the logistic function.

The weight of the first term in Equation 4 at pixel i is given by

$$w(o_i; \theta) = \sum_{j \in N_f} \exp\left(\theta^j f_i^j\right) \qquad (5)$$

where θ^j is the parameter vector associated with feature j and f_i^j is the feature j at pixel i, N_f is the number of features.

The parameters θ can be found by minimizing the negative log-likelihood of the training dataset. We define the

negative log-likelihood of a training image, denoted by $L(\theta)$ as

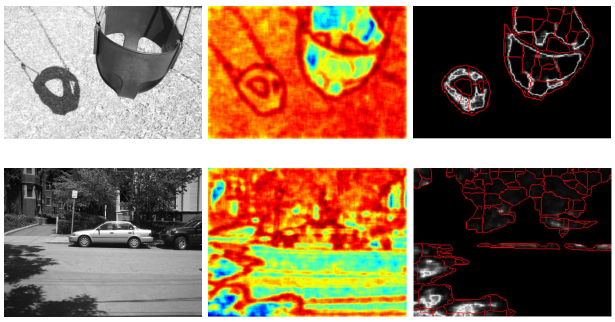
$$L(\theta) = \sum_i \log(1 + \exp(-t_i r_i^*)) + \lambda \sum_{j \in N_f} \theta_j^2 \quad (6)$$

where t_i is the ground-truth probability of each pixel belonging to shadows and the second term is a quadratic regularization term used to avoid overfitting. λ is manually set to 10^{-4}. We use a standard gradient descent method to iteratively update the parameters θ which are all initialized at zero.

The regularizer penalizes the model parameters uniformly, corresponding to imposing a uniform variance onto all model parameters. This motivates a normalization of each type of the feature into [-1,1].

3.3. Feature Boosting

The BCRF works well if the marginal distribution of a pixel belonging to shadows can be defined using the proposed features. Our results show that the linear classifier trained from the BCRF achieve acceptable results (75.2%), but it also missclassified many dark pixels such as near black objects as shadows (see in Figure 8). One reason for this is that the conditional probabilities to distinguish shadows from near black objects are very complex. Figure 6 shows such a case where a pixel likely to be shadow according to the entropy cue varies in different images.



(a) Reference (b) Entropy Map (c) Mask

Figure 6. Examples of conditional probability. In the entropy maps (b), the red colors denote high values while blue ones denote small value. The masks (c) are computed by a soft approach using an exponential function based on two entropy centers clustered from each entropy map. We chose the large center in the first case and chose the small center in the second case. The brightness of the pixel denotes the probability of being a shadow pixel. Part of the segmentation boundaries are shown in red.

Boosted Decision Tree (BDT) [5] builds a number of hierarchy trees by increasing the weights of the misclassified training samples. We sample each segment from the training dataset and learn the BDT estimators. For each pixel in the image, BDT estimator returns a probability distribution over the shadow labels. We sample 40 points from each distribution and output them as new features when training the classifier in the BCRF model. In the experiment section, we show that by integrating BDT outputs into the BCRF model, we achieve the best shadow recognition rate. The technique of combining the labeling distributions from decision trees and random fields has also been introduced in [33], where improved segmentation results are obtained by enforcing the layout consistency.

3.4. Parallel Training

Our requirements to implement a parallel version of learning parameters from the BCRF model has two underlying reasons: the memory requirement for loading all the training data is large and parameter updating requires tremendous computing resources.

Altogether there are 1290 features trained in the BCRF model. The boosting step generates 40 classification results for each pixel. Together with them, a horizontal and vertical derivative, and an one bias feature are used as features in the BCRF model. This gives us totally 43 features. Since BCRF works in pixel-level, we include its 3×3 local patch as a training sample. In this local patch, we include first horizontal and vertical derivatives from 6 different positions on the patch as local smoothness. Referring to Equation (4), we have $43 \times (3 \times 3 + 3 \times 3 + 6 \times 2) = 1290$ features in total. To train 125 images, it requires around 9G of memory without considering additional matrix operations.

In the BCRF model, the most power consuming task is to update the gradient of parameters θ in Equation 6. In each iteration, the gradients are summed over all the training images. This provides an easy way to distribute the gradient computation task into multiple processors as we can assign a portion of training images for each processor.

To accelerate the training speed, we used MatLabMPI [13] which is a set of Matlab scripts that implement a subset of Message Passing Interface (MPI) and allow Matlab programs to be run on multiple processors. One nice feature of MatlabMPI is that it does not require a cluster to execute the parallel commands. We configured three individual PCs including 10 processors and allow them to share 20G of memory. It takes around 10 hours to learn the BCRF parameters in our experiments.

4. Experiments and Results

To evaluate how well the classifiers can locate the shadows, we predict the shadow label at every pixel for 120 test images. The pixels that identified as shadows are then compared with the masks associated with each image. True positives are measured as the number of pixels inside the mask. False positives are measured as the number of pixels outside the mask.

Our results are divided into three groups: comparisons between different types of features, comparisons between different classification models, and comparisons between different levels of over-segmentations.

Overall, our results show that features expressing illumination, textural and odd derivative characteristics can successfully identify shadows. Our results show BDT integerated with BCRF using 2 levels of segments achieves highest recognition rate at 88.7 [1], which is 13.5% higher than BCRF; 11.9% higher than Support Vector Machine (SVM) and 2.9% higher than BDT.

4.1. Quantitative Results

For brevity, we present quantitative results for different features and different models at segment level with a number of 200. The classifiers for different features and different levels are trained from BDT integrated into BCRF. We conclude the numerical results in Table 1.

Comparisons between different types of features

There are two experiments in this comparison. The first experiment has two types of features with only single scalar values. The first type includes only shadow-variant features using illumination, smoothness, local max and skewness. The second type includes all 8 proposed features.

In the second experiment, we augmented these two types of features with their histogram values from segments.

We display all the ROC curves in Figure 7 (a). Our results show that shadow-variant features alone work well on locating shadow, but by augmenting with shadow-invariant features, we can achieve around a 5 percent improvement. We believe this is due to the interaction of variant and invariant features that can aid to train the classifiers and decrease the loss in BCRF.

Comparisons between different classification models

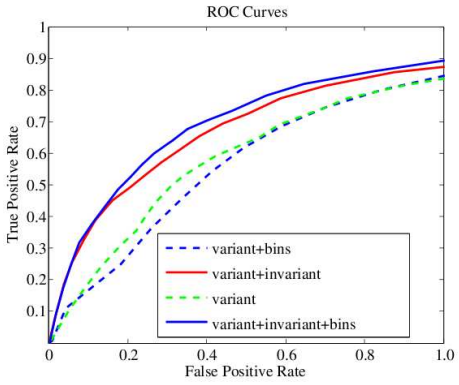
In this comparison, we compared results with Support Vector Machine (SVM), Boosted Decision Tree (BDT), binary CRF models (BCRF) and our combined approach (BDT+BCRF). We obtained around 30,000 training samples (segments) for SVM and BDT, and we trained the classifiers using all the features with their histogram values. We set the number of nodes in a tree to 20 and the initial gain step in the BCRF model to 0.01. It takes around 15 minutes to train the BDT and around 10 hours to train the BCRF in 500 iterations. We set the iteration number to 500 because we found the loss in Equation 6 decreasing very slow afterwards.

We display their PR curves in Figure 7. Our results show that BCRF performs worst with only 75.2% accuracy. This is because linear classifier trained from BCRF performs poorly on complex conditional probabilities, such as distinguishing shadows from near black objects. We can also see SVM using single classification method can only achieve 76.8% accuracy. BDT performs well with 85.8% accuracy. By enforcing the local consistency, BDT+BCRF achieves highest accuracy at 87.3%.

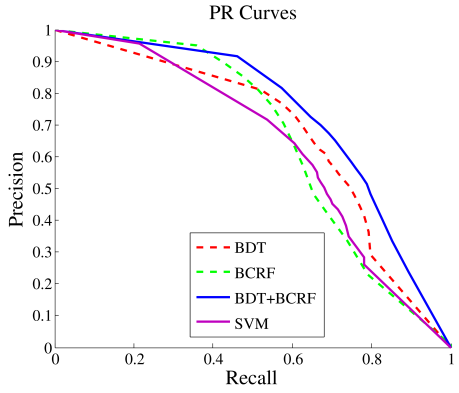
Comparisons between different levels

In this comparison, we did experiments to train the BTD using different number of over-segmentations in an image. We tested number of 200, 500 and their combined experiments. In the combined test, totally 83 features are used in the BCRF, and the number of parameters reaches to 2490. It takes around 24 hours to train the classifer using our parallel version.

From the numerical results, we can see by combining two levels' over-segmentations, the accuracy reaches 88.7% which is the highest accuracy among all the experiments. We believe the single level approaches perform worse than the combined approach because over-segmentation is not perfect in each level. By incorporating multiple levels of segment information, our approach achieves best results but sacrifices in computing recources.



(a) ROC curves from different features



(b) PR curves from different learning models

Figure 7. Quantitative comparison from different features and models. The values in the horizontal axis are normalized to [0,1] which is multiplied by a factor of 5.2. This variable is the ratio between the number of non-shadow pixels and the number of shadow pixels from all the test images.

[1] Note that the number of non-shadow pixels in the test images is larger than the number of shadow pixels.

Table 1. Numerical Comparison of different methods

	Method	Accuracy
Features	Discrepancy	82.3%
	Discrepancy+Bins	80.7%
	Discrepancy+Consistency	86.7%
	Discrepancy+Consistency+Bins	87.3%
Models	SVM	76.8%
	Boosted Decision Tree	85.8%
	Logistic CRFs	75.2%
	BDT+LRF	87.3%
Levels	200 Segments	87.3%
	500 Segments	86.3%
	Combine 200 and 500 Segments	88.7%

4.2. Qualitative Results

We present part of the qualitative results from our test results in Figure 8. We set the threshold to 0.5 when obtaining the binary estimates. For brevity, we only present the results using combined levels of segments.

The first case shows a simple scene with shadows casted on the grass. The probability map from BDT is unsurprisingly inconsistent because BDT treats each segment individually. The result from CRF model alone is neither perfect as we can see some misclassified pixels on the grass and near object boundaries.

The second and third cases show two examples of objects with dark albedo. We can see the probabilities of such objects being shadows from BCRF alone are falsely as high as real shadows. Results from BDT are also not acceptable as thresholding the correct shadows from all the probabilities is very difficult.

The fourth and fifth cases show two complex examples. We can see BCRF model can not distinguish shadows from trees. BDT does good job on separating shadows from trees, but the labeling is inconsistent.

From all these test examples, we can see our combined approach performs better, in terms of both accuracy and consistency, than either of BDT or BCRF model alone.

We show an example from very challenging case in Figure 9. The shadows in this scene are casted on water, which is a transparent media and also appears black. The optical properties of the shadows in the water are not captured well using our proposed features, therefore, both BDT and BCRF perorms poorly.

5. Conclusions

We have proposed a learning-based approach to recognize shadows from monochromatic natural images. Using the cues presented in this paper, our method can successfully identify shadows in real-world data where color is unavailable. We found that while single-pixel classification strategies work well, a Boosted Decision Tree integrated into a CRF-based model achieves the best results.

6. Acknowledgement

This work was supported by a grant from the NGA NURI program(HM1582-08-1-0021),We would also like to thank Dr. Jian Sun for helpful discussions..

References

[1] E. Adelson. Image statistics and surface perception. In *Human Vision and Electronic Imaging XIII, Proceedings of the SPIE, Volume 6806*, pages 680602–680609, 2008. 3

[2] E. Arbel and H. Hel-Ord. Texture-preserving shadow removal in color images containing curved surfaces. In *CVPR*, 2007. 1

[3] V. Arévalo, J. González, and G. Ambrosio. Shadow detection in colour high-resolution satellite images. *Int. J. Remote Sens.*, 29(7):1945–1963, 2008. 1

[4] M. Bell and W. Freeman. Learning local evidence for shading and reflection. In *ICCV*, 2001. 1, 2

[5] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. In *MACHINE LEARNING*, pages 158–169, 2000. 5

[6] R. Dror, A. Willsky, and E. Adelson. Statistical characterization of real-world illumination. *Journal of Vision*, 4(9):821–837, 2004. 2

[7] M. A. T. Figueiredo. Bayesian image segmentation using gaussian field priors. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005. 4

[8] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006. 1

[9] A. D. Forsyth and M. Fleck. Identifying nude pictures. In *IEEE Workshop on the Applications of Computer Vision*, pages 103–108, 1996. 2

[10] A. K. Frederick, C. Beauce, and L. Hunter. Colour vision brings clarity to shadows. *Perception*, 33(8):907–914, 2004. 1

[11] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 76(1), 2009. 4

[12] A. Joshi and N. Papanikolopoulos. Learning to detect moving shadows in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2055–2063, 2008. 1

[13] J. Kepner. Matlabmpi. *J. Parallel Distrib. Comput.*, 64(8):997–1005, 2004. 5

[14] D. Kersten, D. Knill, P. Mamassian, and I. Bülthoff. Illusory motion from shadows. *Nature*, 379:31, 1996. 1

[15] M. D. Levine and J. Bhattacharyya. Removing shadows. *Pattern Recogn. Lett.*, 26(3):251–265, 2005. 1

[16] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multidistribution approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133–1146, 2007. 1

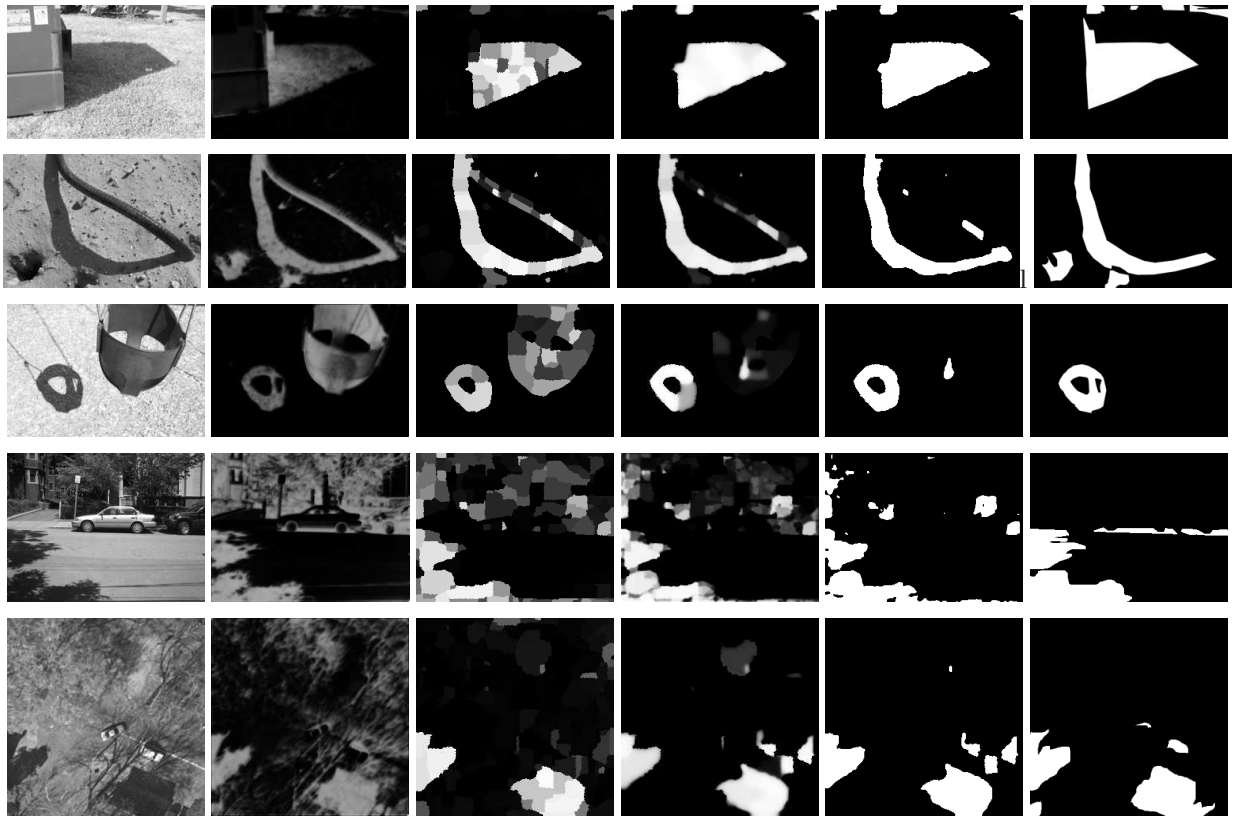
[17] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004. 2, 3

[18] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *CVPR*, pages 973–979, Washington, DC, USA, 2005. IEEE Computer Society. 2

[19] S. Nadimi and B. Bhanu. Physical models for moving shadow and object detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1079–1087, 2004. 1

[20] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003. 2

[21] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *ECCV*, pages 336–350, 2000. 1



(a) Reference (b) BCRF (c) BDT (d) Combined (e) Thresholded (f) Ground truth

Figure 8. Visual comparison of results from Boosted Decision Tree, the BCRF model and our combined approach.

(a) Reference (b) BCRF (c) BDT (d) Combined (e) Thresholded (f) Ground Truth

Figure 9. Result from a very challenging case. The water is flowing from cement. The shadows are the darkest part cast by the cement in the water.

[22] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, 2008. 2

[23] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Comput. Vis. Image Underst.*, 95(2):238–259, 2004. 1

[24] L. Shen, P. Tan, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *CVPR*, 2008. 1

[25] Y. Shor and D. Lischinski. The shadow meets the mask: Pyramid-based shadow removal. In *European Association for Computer Graphics*, 2008. 1

[26] F. Tanner, B. Colder, C. Pullen, D. Heagy, C. Oertel, and P. Sallee. Overhead imagery research data set (oirds) an annotated data library and tools to aid in the development of computer vision algorithms, 2009. 2

[27] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, 2005. 1, 2

[28] M. Tappen, K. Samuel, C. Dean, and D. Lyle. The logistic random field – a convenient graphical model for learning parameters for mrf-based labeling. In *CVPR*, 2008. 2, 4

[29] M. F. Tappen, E. H. Adelson, and W. T. Freeman. Estimating intrinsic component images using non-linear regression. In *CVPR*, pages 1992–1999, 2006. 2

[30] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *The International Conference on Machine Learning*, 2005. 4

[31] Y. Wang, K. Loe, and J. Wu. A dynamic conditional random field model for foreground and shadow segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):279–289, 2006. 1

[32] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, pages 68–75, 2001. 1, 2

[33] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, pages 37–44, 2006. 5