Shadow free segmentation in still images using local density measure

Aleksandrs Ecins, Cornelia Fermüller and Yiannis Aloimonos
Computer Vision Lab
Department of Computer Science, University of Maryland, College Park
aecins@cs.umd.edu, fer@umiacs.umd.edu, yiannis@cs.umd.edu

Abstract

Over the last decades several approaches were introduced to deal with cast shadows in background subtraction applications. However, very few algorithms exist that address the same problem for still images. In this paper we propose a figure ground segmentation algorithm to segment objects in still images affected by shadows. Instead of modeling the shadow directly in the segmentation process our approach works actively by first segmenting an object and then testing the resulting boundary for the presence of shadows and resegmenting again with modified segmentation parameters. In order to get better shadow boundary detection results we introduce a novel image preprocessing technique based on the notion of the image density map. This map improves the illumination invariance of classical filterbank based texture description methods. We demonstrate that this texture feature improves shadow detection results. The resulting segmentation algorithm achieves good results on a new figure ground segmentation dataset with challenging illumination conditions.

1. Introduction

Shadows are visual phenomena which happen when an area in the scene is occluded from the primary light source (e.g. sun). Shadows are everywhere around us and we are rarely confused by their presence. On the contrary they provide an additional source of information about the geometry of the scene [4], position of the light sources [18], shape of the objects [10] etc. However, shadows proved to be a challenging problem for computer vision algorithms. Object detection, segmentation, tracking and stereo are all confused by shadows because they change the appearance of the scene and move together with the objects.

In the past few decades a lot of work was done on detecting moving cast shadows in fixed camera setups aimed at surveillance applications. An overview and comparison of such approaches were made by Prati *et al.* [16] and Al-Najdawi *et al.* [1]. Notable papers include Horprasert *et al.* [9] who introduce a computational color model that separates pixel RGB value changes into brightness and chromaticity components, and Strauder *et al.* [20] who use the observation that in HSV colour space shadows don't change the hue but decrease the saturation of shadowed surfaces. More recently, Salvador *et al.* [17] combined invariant colour features with geometric information to do cast shadow segmentation in both images and video sequences.

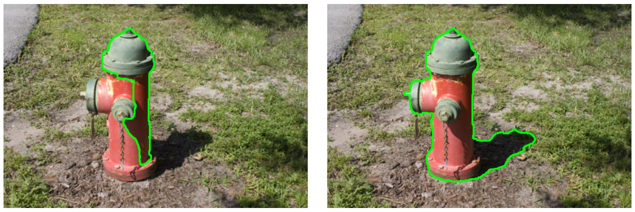

(a) Undersegmentation (b) Oversegmentation

Figure 1: Examples of segmentation errors caused by shadows. (a) Object is undersegmented because of strong self shadow (b) Object is ovsegmented because of strong cast shadow.

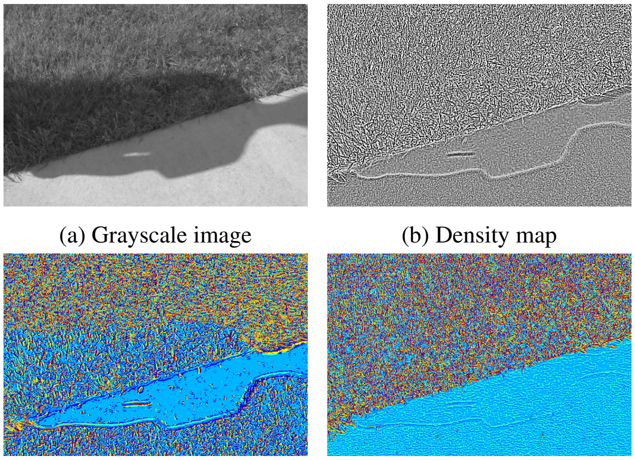
Some approaches leverage image formation theory to derive intrinsic images that are independent of illumination conditions. Weiss *et al.* [22] analyse a sequence of images taken with a static camera to derive an intrinsic representation of the scene. Finlayson *et al.* [7] compute a 2D chromaticity image that is shadow free and find shadow edges by subtracting edges in a shadow free image from edges in the original image.

Most recently, a number of methods were proposed that take advantage of machine learning techniques to detect shadows in single images. Zhu *et al.* [24] use a CRF trained on intensity, texture and odd order derivative features to detect shadow regions in monochromatic images. Lalonde *et al.* [11] detect shadow boundaries by computing intensity, texture and colour difference features across boundary segments obtained from a watershed oversegmentation of the image. Spatial smoothness is then enforced by means of a CRF. Guo *et al.* [8] predict shadow probability for individual image patches and combine them with observations about material and illumination similarity of nearby patches.

In this paper we address the problem of foreground background segmentation of objects affected by shadows in still images. A large amount of existing foreground segmentation algorithms rely on strong image gradients to find the potential object boundaries. As a result, they are prone to be confused by shadows because shadows not only appear as strong boundaries in the image, but also attenuate true object boundaries. There are two ways how a shadow can corrupt a segmentation: either a cast shadow is erroneously included in the segmentation, or a strong self shadow causes the object to be undersegmented (Figure 1). We use a modified version of the segmentation approach proposed by Mishra *et al*. [14]. This algorithm takes as input a user specified fixation point and uses the graph cut algorithm [5] to find a closed contour surrounding a region containing the fixation point. To increase the robustness of the algorithm to the presence of shadows, we first segment the object and then analyse the resulting segmentation boundary to see if it contains strong shadow boundaries. If that is the case we modify the segmentation parameters and resegment again. The main advantage of this active approach is that it saves us the burden of detecting shadow boundaries everywhere in the image, which is both computationally expensive and not well defined. Foreground segmentation naturally defines a boundary of interest so we can focus on a handful of boundary segments that are potentially causing trouble.

To classify shadow boundaries, we take a statistical approach similar to [11]. We first extract intensity, colour and texture dissimilarity features from image boundaries and then feed them to an SVM. For colour and intensity we use features previously proposed in the literature. The difference is in the way we treat texture. Most of the recent approaches use the texture description method introduced in [12]. An image is filtered with a filterbank, and responses are clustered to form a dictionary of prototypical responses called textons. Each pixel is then assigned to its closest texton and the texture of an image patch is described by a histogram of textons. Texture similarity is estimated as the χ^2-distance between texton histograms. Although this approach works well in situations where illumination changes are not significant it is not adequate for images with strong shadows (Figure 2). To remedy this problem we introduce a novel image preprocessing operation to produce the image density map. This map preserves the textural details of the image and is invariant to multiplicative illumination changes. A filterbank technique can be applied to this map instead of the grayscale image, thus giving a rich texture description that is robust to illumination changes.

The main contributions of this paper are: (1) a new image preprocessing technique that improves illumination invariance of filterbank based texture description techniques; (2) a quantitative evaluation of the technique in the context of shadow detection (3); a novel method to improve figure ground segmentation results in the presence of shadows; (4) a quantitative evaluation of the segmentation algorithm on a new dataset of objects in challenging illumination conditions[1].



(a) Grayscale image

(b) Density map

(c) Grayscale textons

(d) Density textons

Figure 2: Visual comparison of textons computed on grayscale image and corresponding density map. Intensity textons differ significantly between the shadowed and sunlit regions of the same material. This effect is reduced for density textons. Note that density map is mapped to $[0, 1]$ and contrast normalized for visualization purposes.

2. Shadows and texture

A widely used scene luminance model [2] describes intensity of an image pixel $\mathcal{I}(\mathbf{x})$ as

$$\mathcal{I}(\mathbf{x}) = \mathcal{R}(\mathbf{x}) \cdot \mathcal{L}(\mathbf{x}) \cdot \mathcal{C}(\mathbf{x}), \tag{1}$$

where $\mathcal{R}(\mathbf{x})$ is the reflectance field, $\mathcal{L}(\mathbf{x})$ is the illumination field and $\mathcal{C}(\mathbf{x})$ is the illumination reduction term which models the effects of shadows. Appearance changes due to shadows are modeled as multiplicative changes to intensity. $\mathcal{C}(\mathbf{x})$ can be smoothly varying to model penumbra or soft shadow effects. However, we are interested in strong shadow boundaries which have no penumbra region, thus we assume that $\mathcal{C}(\mathbf{x})$ is piecewise constant. According to this model the texture of a surface under cast shadows will not change i.e. the ratios of intensities of adjacent pixels are preserved. It is important to note that this model works best for shallow surfaces with small depth variation. A surface with prominent 3D structure (like grass or gravel) exposed

[1] Code and dataset are available at `http://www.umiacs.umd.edu/~aecins/`.

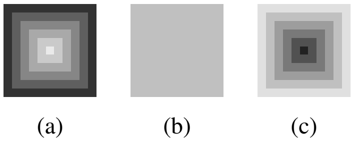


Figure 3: Local density of intensity at center point (a) 1.83; (b) 2; (c) 2.7.

to a directional lightsource will create additional smaller self shadows due to irregularities in the 3D surface. When such a surface falls under cast shadow, the self shadows disappear thus changing the texture appearance of the surface. These effects are difficult to model and we will ignore them in this discussion.

The most common texture description method used for shadow detection is based on the texton approach introduced by Malik *et al.* [12]. A grayscale image is filtered with a filterbank consisting of even and odd symmetric Gaussian derivative filters at multiple orientations and scales. The resulting responses are clustered using k-means to form representative filter responses called textons. Each pixel is then mapped to its closest texton. The downside of this method is that it is not robust to illumination changes. Due to an associative property of convolution, the filter response of a shadowed image patch will be attenuated (i.e. $f * (c \cdot I) = c \cdot (f * I)$). This results in shadow pixels being mapped to lower energy textons. Thus the texton distribution in shadowed region may differ significantly from the same non-shadowed region (Figure 2c).To alleviate this problem, we propose to apply the texton approach on the density map instead of the grayscale image.

2.1. What is a density map?

The notion of density comes from fractal theory. Density measures locally the variation of a quantity over a number of scales. It has been used before for texture classification [23] [21]. The idea is that at small scales naturally occurring objects (here the texture) change over scale in a way, that can be modeled as an exponential function of the scale. Thus the exponent of the function is a good statistical descriptor of the object which is invariant to a wide range of spatial transformations. In this work we focus on the illumination invariance property of the density map.

Let $I(\mathbf{x})$ be a grayscale image and let $\mu(\mathbf{x}, r)$ be a measure on $\mathbb{R}^2$. For our purposes we choose $\mu(\mathbf{x}, r)$ to be the sum of image intensities in a disk of radius r around point $\mathbf{x}$ i.e. $\mu(\mathbf{x}, r) = \sum_{\|\mathbf{y}-\mathbf{x}\| \leq r} I(\mathbf{y})$. We use the power law to express μ as a function of r:

$$\mu(\mathbf{x}, r) = k r^{d(\mathbf{x})} \tag{2}$$

$$\log(\mu(\mathbf{x}, r)) = \log k + d(\mathbf{x}) \log r \tag{3}$$

$$d(\mathbf{x}) = \lim_{r \to 0} \frac{\log(\mu(\mathbf{x}, r))}{\log r} \tag{4}$$

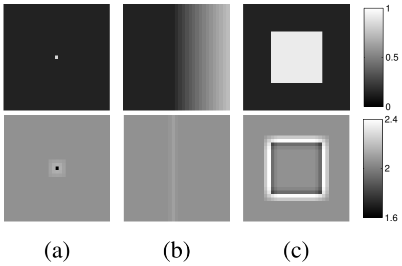


Figure 4: Grayscale images (top row) and corresponding density maps (bottom row). Density map is invariant to smooth changes in intensity but responds to sharp discontinuities.

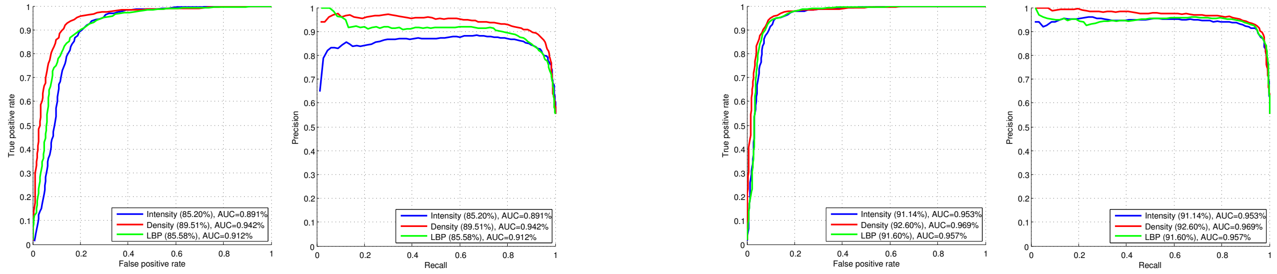
We define the exponent $d(\mathbf{x})$, also known as Hölder exponent, to be the local density function of image $I(\mathbf{x})$ at point $\mathbf{x}$. Intuitively, it measures the degree of regularity of intensity variation in a local neighbourhood around point $\mathbf{x}$. This is illustrated in Figure 3. For a patch where intensity is decreasing from the center, the density value is less than 2, for a patch of constant intensity, the density is equal to 2 and for a patch of increasing intensity the density is greater than 2.

The above definition applies to continuous images. In practice when dealing with discrete images the summation is taken over square regions of several radii (1 to 5 pixels in our implementation). The value $d(\mathbf{x})$ is then obtained by fitting a straight line to the points in the $\log(\mu(\mathbf{x}, r))$ vs $\log r$ plot and estimating the slope of that line.

Figure 4 shows examples of density maps computed on various grayscale images. As can be seen, the density map values are the same within regions of different intensity as well as within regions of smoothly varying intensity. At the same time, the edges of the square show up in the density map. In essence, the density map preserves important textural features by responding to abrupt intensity discontinuities and avoiding smoothly varying regions. Another important property is its invariance to multiplicative changes in intensity. A multiplicative change in intensity corresponds to an additive change of the logarithm of the measurement function μ, which does not affect the slope of the line in $\log(\mu(\mathbf{x}, r))$ vs $\log r$ plot. Thus the density map is well suited to deal with shadows as defined in the model in equation (1). Hence we propose to use the density map as a preprocessing technique before applying the texton approach for texture description in images with shadows.

2.2. Shadow detection

Given a boundary in the image, we want to decide whether it is a shadow boundary by comparing intensity, colour and texture measurements across the boundary. The intuition is that texture and colour can be used to decide whether both sides of the boundary belong to the same ma-



(a) ROC curve for grayscale (b) PR curve for grayscale (c) ROC curve for colour (d) PR curve for colour

Figure 5: ROC and PR curves for shadow boundary detection on grayscale and colour images.

terial while intensity indicates if one side is darker than the other. For each boundary pixel we align a disc of fixed radius r split in two parts, such that the two half discs lie on different sides of the boundary, and we compute the difference of appearance features in the half discs. The orientation at a pixel is found as the direction perpendicular to the tangent of the boundary curve.

For intensity, we calculate the mean intensities in the two half discs and take the ratio of the bright half disc to the dark half disc $\bar{I}_b/\bar{I}_d$. The higher this ratio, the more likely it belongs to a shadow boundary. Similarly, to compare colour, we use RGB colour ratios. We take the ratios of the bright half disc to the dark half disc $\bar{R}_b/\bar{R}_d, \bar{G}_b/\bar{G}_d, \bar{B}_b/\bar{B}_d$. We expect each colour channel to be attenuated by the same amount due to shadow, hence the colour channel ratios should be approximately equal to each other. We experimented with various other colour features (ratio of L*a*b channels, histogram distance of L*a*b), but found the simple RGB colour ratios to give the best results. Finally, for texture we compare a variety of different texture description methods.

Intensity textons: as a baseline, we use the texton method computed on grayscale images [13]. In our implementation, we use a filterbank of a single scale and we don't use the center surround filter because it is sensitive to changes in the average intensity. We use $k = 128$ textons.

Density textons: the second method is our proposed method of computing textons on a density map. The same filterbank parameters are used as in the case of intensity textons.

LBP: the third method is Local Binary Patterns [15] (LBP). This method is widely used in texture classification because of its simplicity and invariance to rotation and illumination. Each pixel is characterized as a binary pattern, where each digit corresponds to one of the adjacent pixels being of either greater or smaller intensity value than the center pixel. Thus each pixel in the image can be mapped to one such binary string.

To compute texture dissimilarity, we compute the normalized histogram of textons (or binary patterns for LBP) in the two half discs and take the χ^2 distance between them.

Once features are extracted for each boundary pixel they are averaged to get a single feature vector describing the boundary. These features are then used to train an SVM classfier. Images from the shadow dataset introduced by Zhu *et al.* [24] are used for training. To collect groundtruth, we applied the *Pb* edge detector [13] to find strong boundaries and then manually labeled them as shadow, and non-shadow. The positive set contains shadow boundaries while the negative contains occlusion boundaries, boundaries between different materials, texture edges and surface orientation boundaries.

2.3. Experiments

We conducted two experiments: one on grayscale images, where only intensity and texture features were used (2D feature vector), and one on colour images, which used all features (5D feature vector). For each experiment we trained three classifiers, each with one of the three different texture features. An SVM classifier with Gaussian RBF kernel was used for classification. SVM parameters were selected on a small development set and a 10 fold cross-validation was run on the rest of the dataset. SVM scores were calibrated to give probabilistic output. Figure 5 shows the ROC and PR curves, and Table 1 shows the confusion matrices for the two experiments.

Referring to figure 5, on grayscale images the density textons achieved an accuracy of 89.51%, which is more than 4% higher than the intensity textons. The improvement comes from higher accuracy on non-shadow boundaries. LBP has improved non-shadow boundary classification compared to intensity textons, but has reduced recognition rate on shadow boundaries, leading to a non-significant overall improvement. This can be explained by the fact that binary patterns only capture the sign of the intensity change around the pixel whereas density textons also preserve information about the magnitude of the change, which makes them more discriminant. Similar results are seen in colour images. Density textons achieved best classification results, however by a noticeably smaller margin. This is due to the fact that both colour and texture are material cues, hence using the colour cue leaves less room for improvement by the texture cue. Most importantly, in both experiments density textons achieved the highest precision in low to medium recall rates. This is important for applications that require high confidence in classification results.

	Grayscale		Colour	
Intensity	Shadow	other	Shadow	other
Shadow	0.911	0.089	0.949	0.051
other	0.219	0.781	0.133	0.867
Density	Shadow	other	Shadow	other
Shadow	0.927	0.073	0.946	0.054
other	0.143	0.857	0.099	0.901
LBP	Shadow	other	Shadow	other
Shadow	0.896	0.104	0.950	0.050
other	0.192	0.808	0.122	0.878

Table 1: Confusion matrices for shadow boundary classification.

3. Shadow free segmentation

This section describes our baseline segmentation algorithm, the effects of shadows on the segmentation process and ways of dealing with them.

3.1. Segmentation algorithm

We use a segmentation algorithm similar to the Interactive Graph Cuts algorithm by Boykov and Jolly [6]. Given an image $I(x, y)$ and some user input, we want to find all pixels belonging to the object. This is done by formulating a graph-based segmentation problem and finding a labeling f that assigns a binary label $f_p \in \{0, 1\}$ to all image pixels, such that the following energy functional is minimized:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}} V_{p,q} \cdot \delta(f_p \neq f_q) \qquad (5)$$

where $\delta(\cdot)$ denotes an indicator function, $D_p(f_p)$ are the unary weights measuring how likely is pixel p to have label f_p and $V_{p,q}$ are the binary weights that measure the likelihood that two adjacent image pixels have same labels. A well known shortcoming of such energy minimization approaches is the tendency to prefer smaller segmentations or shorter contours [19]. To deal with this problem, Misra *et al.* [14] introduced an algorithm that uses a single point belonging to the object as user input. This fixation point is used as a seed for the foreground, while image boundary pixels are used for the background. The energy function is transformed to the Polar coordinate system centered at the fixation point. All circles of different radii centered at the fixation point in Cartesian space have the same boundary cost in Polar space, hence energy minimization becomes independent of boundary length. The downside is that binary and unary weights in Polar space need to be constructed by interpolation from Cartesian space, which leads to unwanted distortions. To avoid this problem, we propose to keep the minimization in Cartesian space, and instead "simulate" the coordinate transformation by reweighing each edge weight $V_{p,q}$ by the distance between points p and q in the desired coordinate space. The binary term becomes:

$$V'_{p,q} = w_{p,q} \cdot V_{p,q} \qquad (6)$$

We choose to simulate the Log-polar space. We have found that in practice, it has the same scale invariant properties as the Polar space and has a much simpler expression for $w_{p,q}$.

Without loss of generality, let's assume that our Cartesian coordinate system is centered at the fixation point. Given a point (x, y) in Cartesian space, the corresponding point in the Log-Polar space is

$$\begin{aligned} \rho &= \ln \sqrt{x^2 + y^2} \\ \theta &= \arctan(\frac{y}{x}) \end{aligned}$$

Denoting $dx = x_p - x_q$, $dy = y_p - y_q$, $d\rho = \rho_p - \rho_q$ and $d\theta = \theta_p - \theta_q$, the distance between points p and q in the Log-polar space is $w_{p,q} = \sqrt{d\rho^2 + d\theta^2}$. Using the transformation:

$$\begin{pmatrix} d\rho \\ d\theta \end{pmatrix} = \begin{vmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} \end{vmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \qquad (7)$$

we obtain that

$$w_{p,q} = \frac{1}{r} \sqrt{(dx^2 + dy^2)} \qquad (8)$$

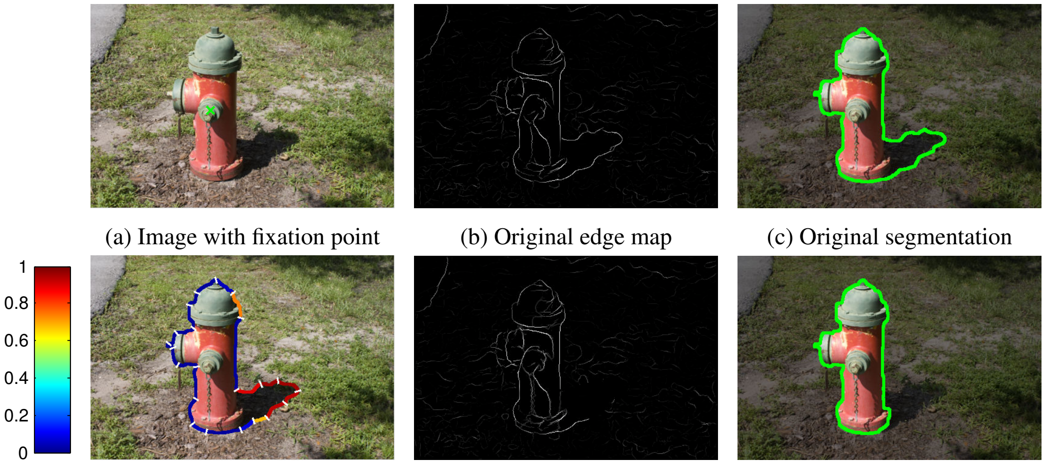
where r is the distance between point p and the fixation point in the Cartesian space. Since we are using discrete images $dx = dy = 1$, hence the weighting function simplifies to $1/r$.

The rest of the segmentation is setup similarly to [14]. The binary weights are

$$V_{p,q} = \begin{cases} \exp(-\eta E_{pq}) & \text{if } E_{pq} \neq 0 \\ k & otherwise \end{cases} \qquad (9)$$

$$E_{pq} = \frac{Pb(p) + Pb(q)}{2} \qquad (10)$$

where $Pb(x)$ is the probabilistic boundary map [13], $k = 20$, $\eta = 5$ and $\lambda = 1000$. The segmentation process works in two iterations. The first one only uses the edge information. The unary weights are set for the fixation point $D_p(f_p = 1) = D$ and the pixels at the image border $D_p(f_p = 0) = D$ with $D = 1000$. The rest of the unary



(a) Image with fixation point (b) Original edge map (c) Original segmentation

(d) Shadow boundary detection (e) Modified edge map (f) Fixed segmentation

Figure 6: Segmentation process.

weights are set to zero. After the first segmentation iteration a preliminary segmentation mask is available and we use a GMM model with 5 components in L*a*b space to construct a colour model of the foreground $F_{BG}(p)$ and background $F_{BG}(p)$. The unary weights for all points except the fixation point and image boundary become:

$$D_p(f_p) = \begin{cases} -\ln(F_{FG}(p)) & \text{if } f_p = 1 \\ -\ln(F_{BG}(p)) & \text{if } f_p = 0 \end{cases} \quad (11)$$

The image is resegmented again with the updated unary terms to give the final segmentation.

3.2. Shadow free segmentation

Presence of shadows in the scene affects both unary and binary weights of the energy function. Unary weights are modified due to changes to the colour model of the shadowed region. Binary weights are affected due to changes of image gradients. Strong gradients are introduced at the boundaries of shadow region, and existing gradients are attenuated inside shadow regions. Combinations of these effects lead to the two most common segmentation errors shown in Figure 1. The first one is undersegmentation, when only part of the object is segmented due to strong self-shadow. The second is over segmentation, when shadow cast on the ground is included in the segmentation.

To mitigate changes to the unary weights, we use an illumination invariant L*a*b colourspace. To deal with changes in binary weights, we need to attenuate shadow edges and reinforce the edge between the object and cast shadow. An obvious approach is to find and attenuate all shadow edges in the image. However, it is not clear how to split the edge map into a set of non-overlapping contours. A contiguous edge in the edge map may belong to several surfaces in the scene, while some edges that belong to the same boundary in the scene may appear disconnected in the edge map. The presence of t-junctions complicates the task even further. Classifying individual pixels produces unreliable results since adjacent pixels are not available to provide orientation information. An alternative approach is to segment the object first and then examine the segmentation boundary for the presence of shadows. Since the segmentation boundary is a closed curve, there are no t-junctions or missing pixels, which makes the task of splitting the segmentation boundary easier. Moreover, the shadow boundaries causing segmentation errors are guaranteed to be included in the segmentation contour since both of the mistakes discussed above involve segmenting along a shadow boundary.

Our approach proceeds as follows. First, the segmentation contour is split into segments by computing 2D contour curvature and splitting at the local curvature maxima. A shadow boundary detector presented in section 2.2 is applied to contour segments. Shadow edges $\mathcal{H}$ with high shadow probability $P(shadow) > 0.8$, are attenuated:

$$Pb(p \in \mathcal{H}) \quad = \quad Pb(p \in H) * 10^{-4} \quad (12)$$

This ensures that the segmentation boundary is less likely to go along the shadow boundaries. We know that the missing edge between cast shadow and object will be inside the initial segmentation. We also know that the shadowed region is likely to have a different colour from the object. Thus, we oversegment the foreground region using the mean shift algorithm [3] and add the edges between segments $\mathcal{T}$ to the probabilistic edge map:

$$Pb(p \in \mathcal{T}) \quad = \quad Pb(p \in \mathcal{T}) + 0.1 \quad (13)$$

Finally, contour segments $\mathcal{L}$ that have low shadow probability $P(shadow) < 0.2$ are likely to belong to the object, hence we increase the corresponding unary weights which forces these pixels to be included in the segmentation:

$$D_{p\in\mathcal{L}}(f_p = 1) = 10 \tag{14}$$

Once these changes to the energy function are made, the segmentation algorithm is run again to give the final shadow free segmentation.

3.3. Experiments

To evaluate the performance of our algorithm we composed a dataset of 53 outdoor images with objects influenced by shadows. 14 images were taken from the dataset of Zhu *et al*. [24], while the other 39 were collected by the authors with a conventional digital camera. The ground truth segmentation mask was manually created for each object, and a fixation point was selected randomly within the mask. To measure segmentation accuracy, we used the F-measure defined as $2PR/(P + R)$, where P and R are precision and recall of foreground mask relative to ground truth. We compare the performance of four algorithms. For a baseline, we use the segmentation algorithm described in section 3.1. and we compare it against our shadow-free algorithm using 3 different texture descriptors (intensity textons, LBP and density textons). All three detectors are trained on Zhu's dataset (images overlapping with our dataset were not used for training). Table 2 shows the F-measure scores. It is clear that our proposed method improves segmentation accuracy over the baseline. Cast shadow is removed completely in most cases shown in Figure 7. Intensity texton and LBP classifiers give a moderate improvement while the density texton classifier increases it even further achieving 7% performance boost over the baseline. This result is consistent with the shadow classification results. Since we are only using high confidence shadow detection results, the classifier with highest precision achieves best performance. There are however several examples where our procedure fails to remove shadow from the segmentation or even degrades the baseline segmentation (Figure 8).

Algorithm	F-measure
Baseline	0.77 ± 0.027
Intensity textons	0.81 ± 0.036
LBP	0.82 ± 0.039
Density textons	0.84 ± 0.033

Table 2: Segmentation quantitative evaluation.

4. Conclusions

We have introduced a new image preprocessing technique that improves illumination robustness of the filterbank based texture descriptors. We applied this technique in the context of shadow boundary detection and showed that it improves performance over the classical filterbank texture descriptor as well as the LBP texture descriptor. We used the resulting shadow boundary detector in the proposed shadow free segmentation approach. Instead of finding shadows everywhere in the scene we take a human inspired approach and only analyse the boundaries that are relevant to the segmentation task. This enables our algorithm to fix segmentation errors due to shadows reliably. In the future we plan to extend this active approach to video sequences where shadows are cast by the objects moving in the scene.

Acknowledgements

The support of the European Union under the grant Poeticon++ in the Cognitive Systems program, and the support of the National Science foundation under an INSPIRE grant in the Science of Learning Activities program (SMA 1248056) are gratefully acknowledged.

References

[1] N. Al-Najdawi, H. E. Bez, J. Singhai, and E. A. Edirisinghe. A survey of cast shadow detection algorithms. *Pattern Recognition Letters*, 33(6):752–764, 2012.

[2] H. G. Barrow et al. Recovering intrinsic scene... In *Computer vision systems*. Citeseer, 1978.

[3] K. Bitsakos, C. Fermüller, and Y. Aloimonos. An experimental study of color-based segmentation algorithms based on the mean-shift concept. In *Computer Vision–ECCV 2010*, pages 506–519. Springer, 2010.

[4] J.-Y. Bouguet, M. Weber, and P. Perona. What do planar shadows tell about scene geometry? In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1. IEEE, 1999.

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

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

[8] R. Guo, Q. Dai, and D. Hoiem. Single-image shadow detection and removal using paired regions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2033–2040. IEEE, 2011.

[9] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV*, volume 99, pages 1–19, 1999.

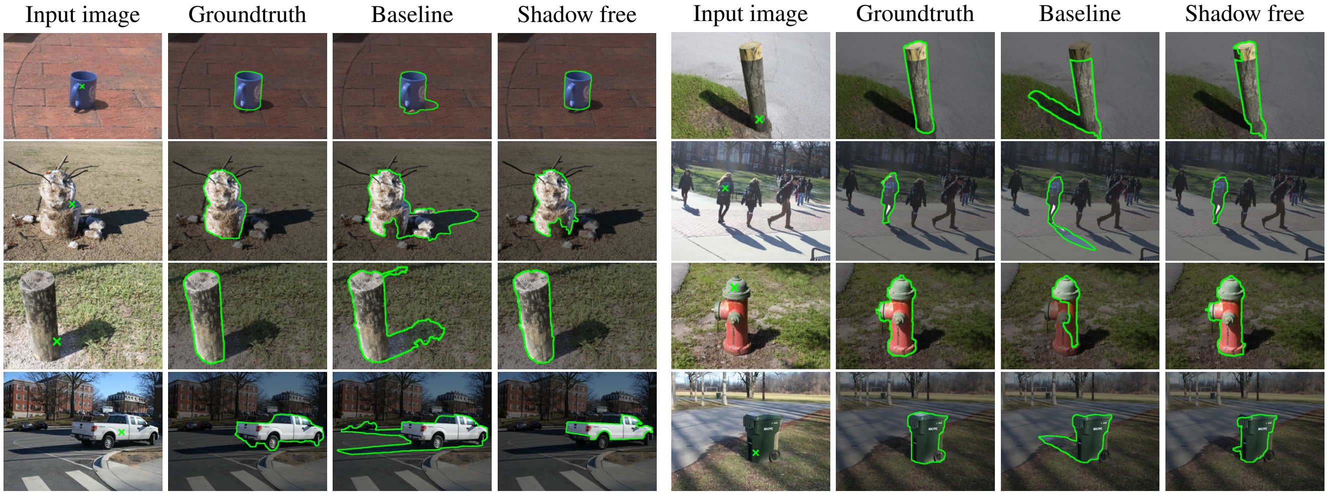


Figure 7: Segmentation examples.

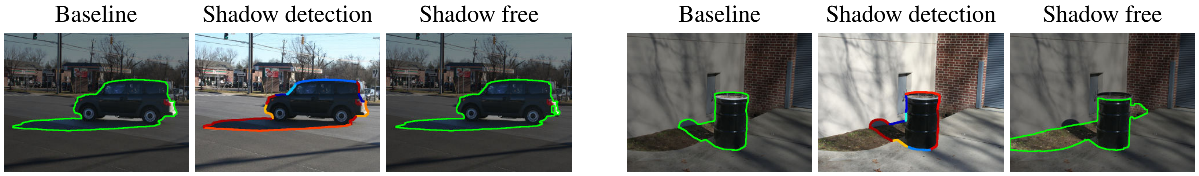


Figure 8: Examples of segmentation failures. In left column even though the shadow boundaries are detected correctly, the similarity of the object and shadow colour models forces the shadow to be included in the segmentation. In right column incorrect shadow detection leads to segmentation boundary expanding beyond the object boundaries.

[10] D. J. Kriegman and P. N. Belhumeur. What shadows reveal about object structure. In *Computer VisionECCV98*, pages 399–414. Springer, 1998.

[11] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *Computer Vision–ECCV 2010*, pages 322–335. Springer, 2010.

[12] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.

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

[14] A. K. Mishra, C. Fermuller, and Y. Aloimonos. Active segmentation for robots. In *IROS*, 2009.

[15] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

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

[17] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Computer vision and image understanding*, 95(2):238–259, 2004.

[18] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(3):290–300, 2003.

[19] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[20] J. Stander, R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *Multimedia, IEEE Transactions on*, 1(1):65–76, 1999.

[21] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[22] Y. Weiss. Deriving intrinsic images from image sequences. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 68–75. IEEE, 2001.

[23] Y. Xu, H. Ji, and C. Fermüller. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, 83(1):85–100, 2009.

[24] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen. Learning to recognize shadows in monochromatic natural images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 223–230. IEEE, 2010.