Systematic Static Shadow Detection

Jian Yao
Department of Computer Science, State University of New York at Binghamton, NY 13905, USA
jyao@binghamton.edu

Zhongfei (Mark) Zhang
Zhongfei@cs.binghamton.edu

Abstract

A systematic static shadow detection algorithm for color images is presented in this paper. The image is modeled by an undirected graph and the shadow detection is achieved through maximizing the graph probability using the EM algorithm. Further analysis shows the connection between our model and the relaxation labeling (RL) model. Experiments clearly indicate that our method is superior to a state-of-the-art shadow detection algorithm.

1. Introduction

Shadow occurs when objects totally or partially occlude direct light projected from a source of illumination. Based on whether shadow is moving or not, shadow can be classified as static shadow and moving shadow. Based on whether shadow is generated by projecting, it can be classified as cast shadow and self (attach) shadow [7]. Only static cast shadow detection problem is addressed in this paper.

Existing static shadow detection methods [1,4,6,7,10,11,12] all have the following drawbacks: 1) Failing to present a systematic method, which makes them not scalable to different applications. 2) Assuming prior knowledge on object geometry or background geometry. 3) Depending on the illumination condition heavily. 4) Requiring additional input images or specific training images.

In this paper, a novel systematic method, which does not have any assumption on the illumination condition, object geometry, or background geometry, is presented. We model the image as a graph and define the node probability and the link probability of the graph. We achieve the shadow detection by maximizing the product of all the probabilities, which is the graph probability. Due to the huge dimension of typical aerial images, an optimization method based on the EM algorithm is presented. Further analysis shows the connection between our model and the Relaxation Labeling (RL) model [8].

The paper is organized as follows. The presented method is described in Section 2; the experimental results are reported in Section 3; and the paper is concluded in Section 4.

2. Detection method

In this paper, we assume that the dimension of the image is $N$ (to be concise, we use one dimensional vector to represent two dimensional image). The image is modeled as an undirected graph, with each pixel as a node in the graph and each connected pixel has a corresponding link in the graph. We use $X_i$ to denote the unknown shadow value of node $i$, which indicates the probability of shadow for the pixel, and use $Y_j$ to denote the observed data of the node $j$, which is the color information. $X$ and $Y$ are the vectors whose components are $X_i$ and $Y_j$. The 8-neighborhood [8] is used.

2.1 Graph probability model

For each node $i$, there is a probability that its $X_i$ value matches to its $Y_i$ value, which is called node probability:

$$\text{ren}(i) = f(X_i,Y_i) = \exp(-X_i - s(Y_i))$$ (1)

where $s(Y_i=a)$ is a random variable which is the estimated $X_i$ value provided that $Y_i$ equals $a$.

For each link, there is also a probability that the two nodes connected through this link are neighbors, which is called link probability:

$$\text{rel}(i,j) = g(X_i,X_j,Y_i,Y_j) = \exp(-X_i - X_j - \text{diff}(Y_i,Y_j))$$ (2)

where $\text{diff}(Y_i=a,Y_j=b)$ is a random variable which is the estimated $X_i-X_j$ provided that the corresponding $Y_i$ and $Y_j$ values are, respectively, $a$ and $b$. As it will be shown later, the distributions of $s$ and $\text{diff}$ are estimated using the whole image. Consequently, they are considered as global information.

Assuming that it is independent between different the node probabilities, between the different link probabilities, and between the node probability and the link probability, the graph probability is defined as the product of all the probabilities:

$$\text{reg}(X) = \prod_{i=1}^{N} \text{ren}(i) \prod_{i=1}^{N} \prod_{j \neq i} \text{rel}(i,j)$$ (3)

In order to find the $X$ value that maximizes (3), we must have the distributions of $\text{diff}$ and $s$ available. The simplest solution to estimate these distributions is to use sample pixels. Unfortunately, due to the illumination variations, $\text{diff}$ and $s$ distributions estimated from sample pixels may not match the actual $\text{diff}$ and $s$ distributions of the individual images. To resolve this problem, an iterative optimization method is proposed: generate the initial distributions for $\text{diff}$ and $s$ based on sample pixels, initialize $X$ for a given image, find $X$ that maximizes (3) under the current $\text{diff}$ and $s$ distributions and the $X$ value, and repeat the last two steps until a convergence occurs.
2.2 Initialization

A set of shadow pixels and a set of non-shadow pixels are extracted from sample images. Based on the experimental results, luminance (L) and chroma (C) are chosen as the color features to distinguish shadow pixels from non-shadow pixels. We quantize the L&C space so that we can use histogram to estimate the distributions of the quantized data. Assuming that L&C space are quantized into M slots (1..M), and that SP(i) and NSP(i) denote the number of shadow pixels and non-shadow pixels with quantized data i among sample pixels, we set:

\[ s(a) = \frac{SP(a)}{SP(a) + NSP(a)}, a = 1..M \]  

(4)

In order to generate the initial distribution of \( s(a) \), we first quantize the difference (the \( X_i \) value of the two neighboring nodes) space into H slots (\( W_1..W_H \)), then we have:

\[ P(\text{dif}(a,b)) = W_j = \frac{|Y_i = a, Y_j = b, X_j - X_j = W_j|}{|Y_i = a, Y_j = b|} \]  

(5)

The same quantization procedure is applied to every image before detection. The initial \( X_i \) value is set by:

\[ X_i = s(Y_i) \]  

(6)

2.3 Iterative procedure

Since \( s(a) \) and \( \text{dif}(a,b) \) are both random variables, we use the EM algorithm [3] to maximize \( \text{Ln}(\text{reg}(X)) \) by considering \( Y \) as the incomplete data. Assuming that we quantize \( s(a) \) into G values (\( T_1..T_G \)), we introduce two unknown parameter sets:

\[ U_{ln} = P(X_i = T_m | Y_i = l) \]  

(7)

\[ V_{mk} = P(\text{dif}(Y_i = l, Y_j = m) = W_k) \]  

(8)

subject to \( \sum_m U_{ln} = 1 \) for each \( l \) and \( \sum_k V_{mk} = 1 \) for each \( (l,m) \). \( U \) and \( V \) are decided by the illumination condition and the reflectance property of individual images.

The E step of the EM algorithm is to form:

\[ Q(X | X^*) = E_{Y,i} \{ \text{Ln} ( \text{reg} (X) | X^*, Y) \} \]  

(9)

and the M step is given by:

\[ \text{Max}_{X^*} Q(X | X^*) \]  

(10)

By simple substitution, (9) becomes

\[ Q(X | X^*) = -\sum_{i=1}^{N} \sum_{j=1}^{H} (X_i - T_j)^2 U_{ij} \]  

\[ -\sum_{i=1}^{N} \sum_{j=1}^{H} (X_i - X_j - W_j)^2 V_{ij,j} \]  

(11)

By setting \( \partial Q / \partial X_i = 0 \) for each pixel, we have \( N \) linear equations which contain \( N \) unknowns. Due to the huge dimension of aerial image, it is impractical to solve those linear equations. A plausible solution is to determine one new \( X_i \) value based on the previous \( X \) value and immediately to modify the \( U \) and \( V \) sets. Unfortunately, besides the additional computation time, the experiments also show that it is not stable. Thus, we choose the following strategy: update \( U \) and \( V \) after all new \( X_i \) are generated, which leads to the following equations:

\[ U_{in}^{new} = \{ i, X_i = l, X_j = T_m \} \]  

(12)

\[ X_i^{new} = (2 \sum_{j=1}^{H} X_j + \sum_{m=1}^{G} U_{in}^{new} T_m + \sum_{j=1}^{H} W_j V_{ij,j}^{new} \]  

\[ -\sum_{j=1}^{H} W_j V_{ij,j}^{old} \} / (2D + 1) \]  

(13)

where \( D \) is the neighboring number. Updating \( V \) follows (5). It is clear from (13) that in order to determine the new \( X_i \), not only local information, e.g., \( X_i \), but also global information, e.g., \( U \) and \( V \), are exploited. At the same time, \( U \) and \( V \) are updated based on the whole image, which helps to catch the illumination information of the individual image. Consequently, our method is considered as a local propagation method with global information utilization to speed up the convergence and to generate a better detection result. We set two stopping criteria for the above procedure: it stops either the iteration number reaches a pre-defined maximum iteration number or the relative MSE (RMSE) becomes acceptable, as depicted below:

\[ \text{RMSE} = \frac{1}{N} \sum_i (X_i^{new} - X_i^{old})^2 < 0.5 \% \]  

(14)

where 0.5 is an empirical threshold.

2.4 Improvements

In the previous procedure, each pixel has an influence on its neighbors’ new \( X_i \) values. We do not expect such an influence to have side effect (e.g., an almost certain shadow pixel would change its status to a non-shadow pixel due to the influence of its not-so-certain non-shadow neighbors). We modify the concepts of committed pixels and uncommitted pixels proposed by Chou and Brown [2] to restrict such an influence. In their original version, a committed pixel was allowed to change its status to the status of other committed pixels but not the status of uncommitted pixels. We make a modification so that a committed pixel is not allowed to change its status. By applying such modification, we let a committed pixel to be the pixel which is definitely shadow pixel or non-shadow pixel and should not be influenced by its neighbors. From (4) and (6), we know that a shadow pixel with \( X_i \) value 0(1) is a certain non-shadow (shadow) pixel and that the higher the absolute difference between its \( X_i \) value and 0.5, the more certain that pixel is. Thus, a pixel is a committed pixel if:

\[ |X_i - 0.5| > T \]  

(15)

where \( T \) is a threshold between 0 and 0.5, which indicates a trade-off between the performance and the cost. The higher the \( T \) is, the better the performance is and the more computation time is required.

Initially all the pixels are uncommitted pixels and the final goal is that all the pixels become committed pixels.
At the end of every iteration, we change the status of those uncommitted pixels if they satisfy (15). When the EM stops, if some pixels are still in the uncommitted status, we decide whether they are shadow pixels or non-shadow pixels by checking whether their shadow values are above or below 0.5. Experiments show that this improvement reduces the convergence time.

Another issue is how to determine $U_{im}$, where we consider different pixels with the same $Y_i$ values to have the same $X_i$ distributions. In fact, such distributions should be local instead of global due to the different reflectance properties among different areas of an image. Consequently, we divide the image into $B \times B$ blocks and assume that the $U_{im}$ set for all the pixels in one block remains the same. Obviously, $B$ cannot be too large; otherwise, the assumption that the $U_{im}$ set in one block is the same may not be valid. On the other hand, due to the curse of dimensionality [5], one block must contain enough pixels to generate the correct $U_{im}$ set, which means that $B$ cannot be too small. Experiments show that the satisfactory $B$ values are between 20 and 80 and that the final performance is increased about 6% in average. Let $B_i$ denote the block index for pixel $S_i$, and let $CP$ denote the committed pixel set and $UCP$ denote the uncommitted pixel set, (11) becomes:

$$X_{i, new}^\prime = X_{i, old}^\prime, S_i \in CP$$

$$X_{i, new} = \frac{1}{2D+1} \left( 2 \sum_{j \in \partial S_i} X_{j, old}^\prime + \sum_{m=1}^G U_{Y_{ib}m} T_{im} \right) + (16)$$

$$\sum_{j \in \partial S_i} W_{kj} = \sum_{j \in \partial S_i} W_{kj} V_{Y_{ib}m}, S_i \in UCP$$

Consequently, the final procedure of the algorithm is: generating the initial distribution from sample image using (4)(5); generating the initial configuration for each image by (6); iteratively applying (12)(5)(16) until (14) is satisfied; and classifying each pixel to shadow pixel or non-shadow pixel based on its final $X_i$ value.

### 2.5 Relationship to RL model

The RL model [8] is described as an approach to minimizing an energy $E(f)$ over a discrete space $S$, which may be the posterior energy of MRF model. It has been widely used in image processing and computer vision. It is achieved through maximizing a gain function:

$$G(p) = \sum_{i \in S} \sum_{I \in G} (C_i - V_{ij}(I))p_i(I) + \sum_{i \in S} \sum_{j \in \partial(i) \in G} (C_j - V_{ij}(I,J))p_i(I)p_j(J)$$

(17)

where $V_{ij}$ and $V_{ij}$ are, respectively, potential functions incurred by single-site cliques and pair-site cliques. $C_i$ and $C_j$ are two constants. $G$ is the label set and $p_i(I) \in [0,1]$ reflects the strength with which node $i$ is assigned label $I$, similar to the $U_{im}$ in our model.

Comparing (11) and (17), we note that both functions contain and only contain single-site potentials and pair-site potentials and both potentials have weights. The difference is that (11) is maximized over $X$ while (17) is maximized over $p$. In fact, $p$ can be considered as the fusion of $X$ and $U$. The label of $p$ corresponds to $X$ while the distribution corresponds to $U$. Furthermore, the weights in (17), i.e., $p/I$, is the conditional probability of the label on the node position, which should not be similar in different images. Besides, the updating of $p/I$ exploits only the local information, i.e., values of node $i$ and its neighbors [8]. On the contrary, as we have already shown, the weights in (11), i.e., $V_{im}$ and $U_{im}$, are conditional probabilities of the label on the node value, which we can assume to be similar for different images provided that the illumination variation is not large. Besides, their updating exploits the information of the whole image or whole block, which makes our method fit different illumination conditions.

### 3. Experiments

We conduct several experiments to evaluate and compare our algorithm with a state-of-the-art algorithm [10] under different situation: similar scenery with similar illumination conditions, similar scenery with different illumination conditions, and different scenery.

42 testing images are manually divided into three sets. Set one contains 19 images with similar illumination conditions from similar scenery. Set two contains 9 images, which have different illumination conditions with Set one but are from similar scenes. Set three contains 13 images, which have different scenarios with Sets one and two. Three images from Set one are chosen to be sample images for training. All the images are manually ground truthed.

We modify the evaluation method given by Prati et al. [9], which separates the images into shadow, object, and background, to quantitatively evaluate our method and compare it with the algorithm in [10]. The evaluation metrics are defined at the pixel level using FP, FN, TP, and TN as:

- Correctness: $100 \times TP/(TP+FN)$
- Accuracy: $100 \times TP/(TP+FP+FN)$

The correctness metric is a measure of correctly detected shadow pixels among all shadow pixels. The accuracy reports totally accuracy of the method, which takes both FP and FN into account. For a good shadow detection algorithm, both correctness and accuracy should be high.

The first experiment is to compare the performance among similar illumination conditions between our method and [10]. We use all the images of Set one as the testing images. Table one shows the evaluation results for both methods. It is clear that both correctness and
accuracy metrics show that our method is better. We also note that the correctness difference between ours and [10] is much less than the accuracy difference between ours and [10]. The reason is that in [10] there are many FP shadow pixels. Shadow detection examples for both algorithms are presented in Figure 1. Comparing the two detection results, we note that the two windows in the top left area are detected as shadow using [10]; part of the self shadow of the building in the center is detected as shadow; several shadow of tree are missed, the shadow cast by a tree to the roof of a building is missed using [10]. Both algorithms incorrectly detect the self shadow of the building wall at the lower-left corner and the vehicles in the parking lots as shadow. The reason may be that the texture information for those areas is very similar to that of typical real shadow.

<table>
<thead>
<tr>
<th>Correctness (exp.1/exp.2/exp.3)</th>
<th>Accuracy (exp.1/exp.2/exp.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ours 95.3%/93.2%/89.4%</td>
<td>91.3%/89.6%/76.8%</td>
</tr>
<tr>
<td>[10] 90.3%/79.1%/67.4%</td>
<td>80.7%/68.4%/53.4%</td>
</tr>
</tbody>
</table>

Table 1: Correctness and accuracy comparison between our method and [10]. Three numbers in each cell are the three results of experiments 1, 2, and 3.

Figure 1: Sample detection results. The first row images are original images; the second row images are detection results of our algorithm; the third row images are detection results of [10].

The second experiment is to compare the performance among different illumination conditions between our method and [10]. We use all the images in Set two as testing images. Table 1 shows the evaluation results for both methods. Comparing with experiment one, we note that our performance almost remains the same in both experiments while that of [10] decrease substantially, which proves that our method is illumination and brightness condition independent.

The third experiment is to compare the performance among different scenes between our method and [10]. We use all the images in Set three as testing images. Table 1 shows the evaluation results for both methods. Note that both algorithms show worse performance compared to previous results. However, our method leads to less degeneration than [10]. Right three images of figure 1 show another comparison. It is clear that all the dark regions near the left building are falsely detected as shadow using [10], and on the other hand, one of the building shadow cast to another building is detected by our method. Unfortunately for both algorithms fail to detect some other building shadows cast to other building and there is still FP shadow detected in the left building area.

4. Conclusions

A systematic shadow detection algorithm is presented in this paper. The image is modeled by a graph and shadow detection is achieved by maximizing the graph probability using the EM algorithm. Further analysis shows the connection between our model and the RL model. Experiments clearly indicate that our method is superior to a state-of-the-art shadow detection algorithm.

Reference