

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_j to denote the unknown shadow value of node j , 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_j 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:

$$ren(i) = f(X_i, Y_i) = e^{-(X_i - s(Y_i))^2} \quad (1)$$

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:

$$rel(i, j) = g(X_i, X_j, Y_i, Y_j) = e^{-(X_i - X_j - dif(Y_i, Y_j))^2} \quad (2)$$

where $dif(Y_i=a, Y_j=b)$ 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 dif 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:

$$reg(X) = \prod_{i=1}^N ren(i) \prod_{i=1}^N \prod_{j \in \partial(i)} rel(i, j) \quad (3)$$

In order to find the X value that maximizes (3), we must have the distributions of dif and s available. The simplest solution to estimate these distributions is to use sample pixels. Unfortunately, due to the illumination variations, dif and s distributions estimated from sample pixels may not match the actual dif and s distributions of the individual images. To resolve this problem, an iterative optimization method is proposed: generate the initial distributions for dif and s based on sample pixels, initialize X for a given image, find X that maximizes (3) under the current dif and s distributions and the X value, re-compute the distributions of dif and s under the new X , 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 dif and s . 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:

$$s(a) = \frac{SP(a)}{SP(a) + NSP(a)}, a = 1..M \quad (4)$$

In order to generate the initial distribution of dif , 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:

$$P(dif(a,b)=W_i) = \frac{|\{Y_i=a, Y_j=b, X_i-X_j=W_i\}|}{|\{Y_i=a, Y_j=b\}|} \quad (5)$$

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

$$X_i = s(Y_i) \quad (6)$$

2.3 Iterative procedure

Since $s(a)$ and $dif(a,b)$ are both random variables, we use the EM algorithm [3] to maximize $Ln(reg(X))$ 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:

$$U_{lm} = P(X_i = T_m | Y_i = l) \quad (7)$$

$$V_{lmk} = P(dif(Y_i = l, Y_j = m) = W_k) \quad (8)$$

subject to $\sum_m U_{lm} = 1$ for each l and $\sum_k V_{lmk} = 1$ 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:

$$Q(X | X^n) = E_{U,V} \{ Ln(reg(X) | X^n, Y) \} \quad (9)$$

and the M step is given by:

$$Max_X Q(X | X^n) \quad (10)$$

By simple substitution, (9) becomes

$$Q(X | X^n) = - \sum_{i=1}^N \sum_{l=1}^G (X_i - T_l)^2 U_{Y_i,l} - \sum_{i=1}^N \sum_{j \in \partial S_i} \sum_{l=1}^H (X_i - X_j - W_l)^2 V_{Y_i,Y_j,l} \quad (11)$$

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 s are generated, which leads to the following equations:

$$U_{lm}^{new} = |\{i, Y_i = l, X_i = T_m\}| / |\{i, Y_i = l\}| \quad (12)$$

$$X_i^{new} = (2 \sum_{j \in \partial S_i} X_j + \sum_{m=1}^G U_{Y_i,m}^{new} T_m + \sum_{j \in \partial S_i} \sum_{k=1}^H W_k V_{Y_i,Y_j,k}^{new} - \sum_{j \in \partial S_i} \sum_{k=1}^H W_k V_{Y_j,Y_i,k}^{new}) / (2D + 1) \quad (13)$$

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_j s, 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:

$$RMSE = \frac{1}{N} \sum_i \left(\frac{X_i^{new} - X_i^{old}}{X_i^{old}} \right)^2 < 0.5\% \quad (14)$$

where 0.5 is an empirical threshold.

2.4 Improvements

In the previous procedure, each pixel has an influence on its neighbors' new X_j 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:

$$|X_i - 0.5| > T \quad (15)$$

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_{lm} , 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_{lm} set for all the pixels in one block remains the same. Obviously, B cannot be too large; otherwise, the assumption that the U_{lm} 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_{lm} 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:

$$\left\{ \begin{array}{l} X_i^{new} = X_i^{old}, S_i \in CP \\ X_i^{new} = \frac{I}{2D+I} \left(2 \sum_{j \in \partial S_i} X_j^{old} + \sum_{m=1}^G U_{Y_i, B, m} T_m + \right. \\ \left. \sum_{j \in \partial S_i} \sum_{k=1}^H W_k V_{Y_i Y_j k} - \sum_{j \in \partial S_i} \sum_{k=1}^H W_k V_{Y_j Y_i k} \right), S_i \in UCP \end{array} \right. \quad (16)$$

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:

$$G(p) = \sum_{i \in S} \sum_{I \in G} (C_1 - V_1(I)) p_i(I) + \sum_{i \in S} \sum_{I \in G} \sum_{j \in \partial(i)} \sum_{J \in G} (C_2 - V_2(I, J)) p_i(I) p_j(J) \quad (17)$$

where V_1 and V_2 are, respectively, potential functions incurred by single-site cliques and pair-site cliques. C_1 and C_2 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_{lm} 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., $p_i(I)$, is the conditional probability of the label on the node position, which should not be similar in different images. Besides, the updating of $p_i(I)$ 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_{lmk} and U_{lm} , 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 sceneries 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.

	Correctness (exp.1/exp.2/exp.3)	Accuracy (exp.1/exp.2/exp.3)
ours	95.3%/93.2%/89.4%	91.3%/89.6%/76.8%
[10]	90.3%/79.1%/67.4%	80.7%/68.4%/53.4%

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.

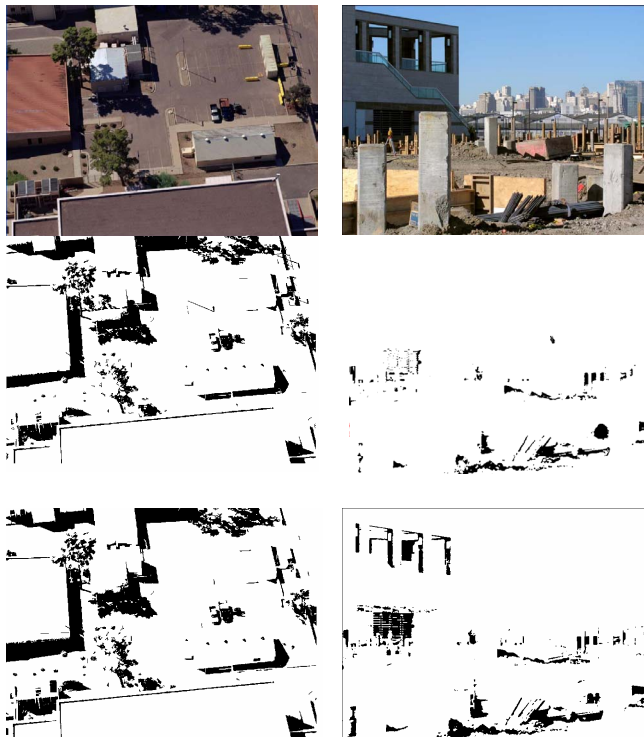


Figure 1: Sample detection results. The first row images are original images; the second row images are detection results of our algorithm; the third row images are detection results of [10].

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

- [1] K.Barnard and G.Finlayson, "Shadow identification using color ratios", Proceedings of the IS&T/SID, pp. 97-101, 2000
- [2] P.B.Chou and C.M.Brown, "The theory and practice of bayesian image labeling", *Int. J. Comp. Vis.*, 4(1990), 185-210
- [3] A.P.Dempster et al, "Maximum-likelihood from incomplete data via the em algorithm", *J.Royal Statist. Soc.*, 39, 1977
- [4] G.D. Finlayson, S.D. Hordley, and M.S.Drew, "Removing shadows from images", *ECCV 2002*, 823-836, 2002
- [5] A.K.Jain, R.P.W.Duin, and J.Mao, "Statistical pattern recognition: A review", *PAMI*, 22(1), 2000, pp4-37
- [6] C.Jaynes et al., "Dynamic shadow removal from front projection displays", *VIS 2001*, 175-182
- [7] C.Jiang and M.O.Ward, "Shadow Identification", *CVPR*, 1992, 606-612
- [8] Stan.Z.Li, "Markov Random Field Modeling in Image Analysis", 2001
- [9] A.Prati, R.Cucchiara, I.Mikic, M.M. Trivedi, "Analysis and detection of shadows in video streams: a comparative evaluation", *CVPR 2001*
- [10] E.Salvador, A.Cavallaro, T.Ebrahimi, "Shadow identification and classification using invariant color models", *ICASSP*, Vol 3, 2001, pp1545-1548
- [11] J.M.Scanlan et al., "A Shadow detection and removal algorithm for 2-D images", *ICASSP*, 1990, 2057-2060
- [12] Y.Weiss, "Deriving intrinsic images from image sequences", *ICCV 2001*, 68-75