Appears in: 4ICPR 2000

Moving Shadow and Object Detection in Traffic Scenes

Ivana Mikić, Pamela C. Cosman, Greg T. Kogut, Mohan M. Trivedi
Department of Electrical and Computer Engineering, University of California, San Diego, USA

Abstract

We present an algorithm for segmentation of traffic scenes that distinguishes moving objects from cast shadows. Three image features at each pixel site are considered: brightness, and normalized red and blue color components. Each feature is analyzed by an a posteriori probability estimator that computes membership probabilities for the three classes: background, foreground and shadow. The algorithm iterates between the three estimators by using the output of one as the a priori probability input to the next. This approach is used in iterative decoders, a component of turbo codes. Thus, we have named the algorithm turbo segmentation. A fading memory estimator calculates mean and variance of features for background pixels. Given the statistics for a background pixel, simple rules for calculating its statistics when covered by a shadow are used. In addition to the color features, we examine the use of neighborhood information to produce smoother classification. We also propose the use of temporal information by modifying class a priori probabilities based on predictions from the previous frame.

1 Introduction

This work is motivated by the need for a robust segmentation algorithm to be used in a traffic monitoring and incident detection system. Of course, extracting positions of moving objects in image sequences is an important component for many other applications. Background subtraction is a common approach to this problem. The background model is built from the data and objects are segmented if they appear significantly different from the background. Unfortunately, moving shadows are usually extracted along with the objects. This can result in large errors in object localization and can cause serious problems for algorithms that use segmentation results as their basic measurements.

Many algorithms that detect shadows take into account the location of the light source, geometry of the scene and models of moving objects [1]. Our aim was to avoid using any such knowledge in detecting shadows. One such algorithm, proposed by Strauder et al. [2], instead assumes that static edges caused by background texture remain in regions covered by shadows and that shadows have penumbra, a soft luminance transition at the contour of the shadow. However, this is rarely true for outdoor scenes, where shadows usually have sharp edges and background is often non-textured. Even with textured background, our experience is that texture is almost invisible in the shadow regions due to the properties of the imaging process.

Without using scene models and assumptions mentioned above, we can identify three sources of information that can help in detecting objects and shadows. The first is local, based on the appearance of the individual pixels. A point covered by a shadow gets darker, its blue component increases and its red component decreases compared to its appearance when illuminated. The second source of information is spatial: objects and shadows inhabit compact regions in the image, and the third is temporal: object and shadow positions can be predicted from previous frames.

We propose an algorithm that utilizes all three sources of information to classify pixels into the shadow, object and background classes. In Section 2, we describe simple rules for deriving statistics of a point covered by a shadow given its statistics when illuminated. In Section 3, we present the iterative segmentation algorithm based on pixel appearance and we then examine the use of spatial information and propose a way of incorporating temporal information.

2 Color change under shadow

The luminance. The luminance l at image point (x,y) is given by [2]:

$$l(x,y) = E(x,y)\rho(x,y) \tag{1}$$

where $E(x,y)$ is the irradiance, the amount of light power per surface area the object is receiving, and $\rho(x,y)$ is the reflectance of the object surface. Irradiance can be expressed as:

$$E(x,y) = \begin{cases} c_A + c_P \cos\angle(N(x,y), L), & \text{if illuminated} \\ c_A, & \text{if covered by a shadow} \end{cases} \tag{2}$$

where c_P and c_A are intensities of the light source and ambient light, N is the object surface normal and L the direction of the light source.

Since the reflectance of the background does not change with time, the ratio of luminance values of a point when illuminated and when covered by a shadow, $r(x,y)$ is:

$$r(x,y) = \frac{l_{IL}(x,y)}{l_{SH}(x,y)} = \frac{c_A + c_P \cos \angle(N(x,y),L)}{c_A} \quad (3)$$

If the background is mostly flat, we can assume that this ratio is constant for background points and equal to r and estimate it from the example data. If the background is not flat over the entire image, we can divide the image into subregions where this assumption is more likely to hold and model each subregion separately.

With the assumption of constant $r(x,y)$, it is easy to derive the rule for estimating μ_{SH}^B and σ_{SH}^B, the mean and standard deviation of a given pixel brightness when covered by a shadow, given μ_{IL}^B and σ_{IL}^B, the statistics for the same pixel when it is illuminated:

$$\begin{aligned} \mu_{SH}^B &= \mu_{IL}^B / r \\ \sigma_{SH}^B &= \sigma_{IL}^B / r \end{aligned} \quad (4)$$

The color. We are not aware of the quantitative analysis of color changes under shadow similar to the one for brightness changes presented in the previous section. What is common knowledge is that the blue color component increases and red decreases. We, therefore, chose to use normalized blue and red color components as features: $b = B/(R+G+B)$, $r = R/(R+G+B)$.

We empirically design rules to calculate the mean and variance of these features for a pixel covered by a shadow, given the same statistics when it is illuminated. Figure 1 shows plots of illuminated-shadowed pairs for individual pixels and histograms for illuminated and shadowed pixels for normalized blue color component (plots for normalized red look similar with mean value decreased and variance increased for shadowed pixels).

We see from Figure 1a that the ellipse is aligned with the coordinate axes and therefore it is impossible to derive a simple scaling rule similar to the one for brightness. We instead assume that the mean of the normalized blue component is increased by a constant value. We similarly find a constant decrease for the mean of the normalized red component. We observe an increase in variance for both normalized color components and design the following rules for deriving statistics of color features for a pixel covered by a shadow:

$$\begin{aligned} \mu_{SH}^b &= \mu_{IL}^b + \Delta b \\ \mu_{SH}^r &= \mu_{IL}^r - \Delta r \\ \sigma_{SH}^b &= \sigma_{IL}^b \times fb \\ \sigma_{SH}^r &= \sigma_{IL}^r \times fr \end{aligned} \quad (5)$$

where Δb, Δr, fb, and fr are parameters that are easily estimated from the data by subtracting means and dividing variances of values for illuminated and shadowed pixels.

We have also experimented with a few features such as hue, saturation, color purity (R+G+B-3min(R,G,B)) and color component ratios (B/R) and found that using normalized color components yields a slightly better performance of the segmentation algorithm than the B/R ratio and color purity do, but significantly better results than when using hue. More work needs to be done to find and justify good choices of color features.

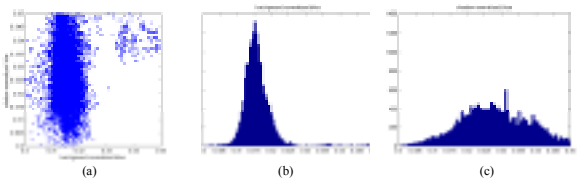

Figure 1 (a) plots of illuminated-shadowed normalized blue color component pairs for individual pixels. (b) histogram of the illuminated pixel normalized blue values. (c) histogram for the shadowed pixels

3 The segmentation based on pixel color change

A fading memory estimator calculates background mean and variance for all pixel locations. Using the rules presented in the previous section, we derive statistics for same pixels when shadowed. Gaussian distributions are assumed for background and shadow pixels, and uniform distribution is assumed for foreground.

We start the segmentation by comparing the luminance of each pixel to the mean luminance at that location in the background model. If not significantly different (difference less than 10% of the mean), the pixel is classified into the background class. Otherwise, we assign to that location the a priori probabilities p_{BG}, p_{SH}, and p_{FG} of belonging to background, shadow and foreground classes, respectively. We have found that the choice of a priori probabilities does not affect the final result unless extremely large or small values are used. Then we enter the iterative estimation process illustrated in Figure 2. Three a posteriori probability estimators are used. The first one computes the a posteriori probabilities of the pixel belonging to each of the three classes (C_1 = background, C_2 = shadow and C_3 =

foreground) based on the luminance of a pixel and on the probability distributions for luminance in our models of three classes. The second one does the same, looking at the normalized blue, and the third one looks at the normalized red color component:

$$p(C_i/v)=\frac{p(v/C_i)p(C_i)}{\sum_{j=1,2,3} p(v/C_j)p(C_j)} \tag{6}$$

where v is the luminance, normalized red or blue value for a given pixel in each of the estimators. We iterate between the three estimators by using the a posteriori probability estimates from the output of one as the a priori probability inputs to the next.

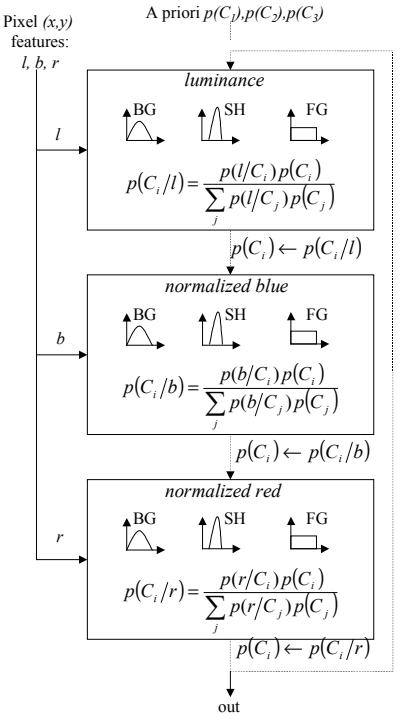

Figure 2 The diagram of the iterative estimation procedure. C_1 is the class background, C_2 shadow and C_3 foreground, and l, b and r are luminance, and normalized blue and red color components for a pixel (x,y)

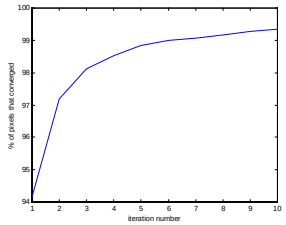

Figure 3 Percentage of pixels for which one of the membership probabilities exceeds 0.9 as a function of the iteration number

Such an iterative approach is used in iterative decoders, a component of turbo codes [3]. Hence we named the algorithm turbo segmentation. There is no guarantee of convergence for this algorithm [4], but empirical results in turbo decoding have been very encouraging. We consider that convergence has been achieved for a pixel

if one of its membership probabilities reaches 0.9 and stays above this value for the number of iterations we examined. In Figure 3 we show the percentage of pixels for which the algorithm has converged as a function of the number of iterations for one frame of a traffic scene. We use two iterations since additional ones result in diminishing improvements. After the iterations are completed, the final decisions are made by thresholding output membership probabilities with a fixed threshold of 0.9. If none of the three probabilities exceeds that threshold, the pixel is assigned to the background.

4. Imposing spatial constraints

The majority of the pixels are classified correctly by the described appearance-based algorithm (around 73%, when compared to the hand-segmented images). However, object and shadow regions are very noisy due to misclassified pixels. The results can be significantly improved by imposing the spatial smoothness. We investigated two approaches. First is simple post-processing by spatial filtering of the segmented images. We eliminate the small gaps in foreground regions by performing one vertical and then one horizontal scan and assigning an encountered small line segment of non-foreground pixels to foreground if it is surrounded by foreground pixels in the direction of the scan. This is followed by morphological opening.

The second approach we investigated was introducing a fourth estimator into the iterative loop. This component assigns a probability of class membership to a pixel based on the probabilities of its neighbors. We have found that the probabilities have to be dramatically changed to change the final classification of a given pixel. This is due to very strong influence of the three color-based estimators. We got the best results by assigning to the central pixel in a window the probabilities associated with its neighbor with the highest membership probability for the class that dominates the neighborhood. This component is inserted into the loop after the luminance estimation step. We have found that the results are slightly improved (around 78% of pixels correctly classified – see Figure 4). However, there is still a need for post-processing that is of similar complexity to the post-processing described in the previous paragraph, which we used on original segmentation results. The final result is very similar (around 90% of pixels classified correctly). Also, adding the spatial estimator into the iterative loop significantly reduces the speed of the algorithm. We therefore conclude that the spatial smoothness is imposed most efficiently by a simple post-processing step.

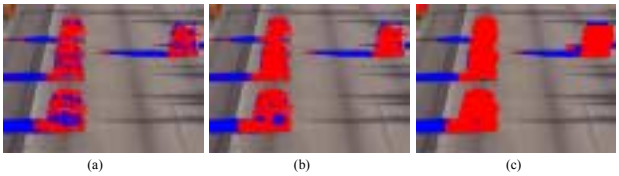

Figure 4. Imposing spatial smoothness. (a) result of the color based segmentation. (b) result of adding a smoothing component to the iteration loop. (c) Result of post-processing of (a).

5 Results

Figure 5 shows segmentation results for one frame from the video of a traffic scene. By correctly classifying shadows and flickering background pixels that simple background subtraction would classify as foreground, the accuracy of the calculated object locations is greatly improved, especially in scenes with long shadows. Note that static shadows are considered to be part of the background. Segmented shadows also provide an important clue for separating objects that are so close that they are segmented as one object. Often in those cases, the shadows of such objects will be distinct and help us separate the objects (see Figure 5d). Figure 6 shows results on several video frames.

.

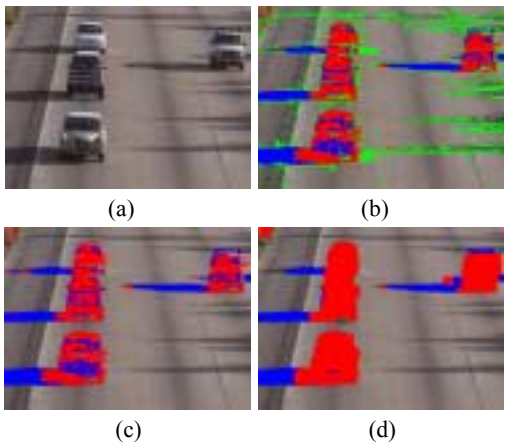

Figure 5. Moving shadow and object detection. (a) the original image frame. (b) Classification results after the second iteration. Red pixels are classified as foreground, blue as shadow and green as background. (c) Same as in (b), with background pixels not shown. (d) final result after post-processing by a spatial filter

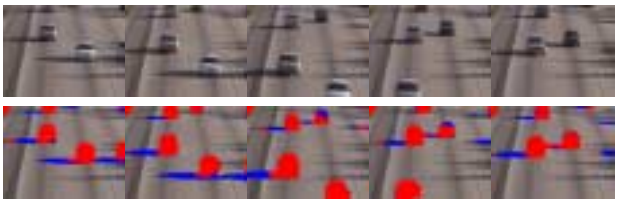

Figure 6. Five frames from the video of a traffic scene. Top row shows the raw video data and the bottom row shows the results of the algorithm

6 Conclusions and future work

We have presented a real time algorithm for segmentation of moving objects and cast shadows in image sequences. Our final aim is to have a deployable algorithm able to operate over extended periods of time and provide robust measurements for a traffic monitoring system.

We propose several improvements and areas of further study. First, including temporal information could significantly improve the performance of the algorithm without much speed degradation. We could use predicted object locations to select a priori probabilities in the current frame. Locations where we expect objects of one class would be assigned high corresponding a priori probabilities.

Another important direction of future work is analysis of the relationship between the scene illumination and the parameters of the model that is used to derive shadow statistics given the statistics of a point when it is illuminated. As the algorithm adapts background statistics to the slow changes in the scene conditions, it could also collect statistics for shadow pixels it identified with high confidence and modify the parameters of the change rules: r, Δb, Δr, fb, and fr. By independently measuring illumination at the scene, we should be able to build a database of these parameters indexed by the scene illumination and use it to recover from sudden changes in scene conditions.

Finally, a systematic study of different color features to be used for this algorithm is important. We have chosen normalized red and blue based a few image sequences. It is possible that some other features could result in better performance

References

[1] D. Koller, K. Danilidis, H. H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes", *Int. Journal of Computer Vision*, 10:3, 257-281, 1993

[2] J. Stauder, R. Mech, J. Ostermann, "Detection of Moving Cast Shadows for Object Segmentation*", IEEE Trans. Multimedia*, 1:1, 65-76, 1999

[3] W. E. Ryan, "A Turbo Code Tutorial", http://www.ee.virginia.edu/research/CCSP/turbo_codes/tcodes-bib/turbo2c.ps

[4] B. J. Frey, F. R. Kschischang, "Probability Propagation and Iterative Decoding*", Proc. 34th Allerton Conf. On Communication, Control and Computing*, 1996