Using Intrinsic Images for Shadow Handling

Robert F.K. Martin, Osama Masoud, Nikolaos Papanikolopoulos
{martin, masoud, npapas}@cs.umn.edu
Artificial Intelligence, Vision and Robotics Lab
Department of Computer Science and Engineering
University of Minnesota

Abstract— **The ability to detect shadows is a critical feature of any intelligent transportation system (ITS). The improper handling of shadows can be the cause of erroneous conclusions in traffic analysis. Because vision-based ITS applications are becoming more popular, it is important to minimize the effects of shadows. Here, we present a novel method for shadow handling in a sequence of images. Following recent work motivated by studies of the statistics of natural images, we show that this new method is a reliable detector of shadows and can be easily implemented in realtime.**

Keywords— **shadow handling, intrinsic images, vehicle detection.**

I. Introduction

A common goal of all intelligent transportation systems is to pass as much information about a scene as possible to later decision-making functions. Whether the information is gathered through magnetic sensors, radar, or vision-based systems, a slight error in measurement can have serious effects "down the road". Reliable traffic analysis requires careful design of any ITS.

Traffic monitoring and analysis have been around nearly as long as the automobile itself. As technology has changed, so has the ability to gather a wider, more complete range of data. With the drop in price and rise in power of computers, many people have turned their interest to vision-based systems. Because of the low computational load, vision-based systems have been used for a variety of such things as intersection management, monitoring weaving sections, and estimating traffic flow [3], [7].

One of the first steps in an ITS application is to separate the foreground from the background. There have been a number of different techniques developed for this but the standard has been background subtraction. In this method, an average background is computed and subtracted from frames of a video sequence. That difference is then thresholded to reveal the foreground. There are two issues with this: how does one compute the average background and how does one decide the threshold. Various algorithms have been developed to determine the average background from a sequence of images and they have been used with success in [3], [4], [5]. However, the threshold generally allows shadows to be included in the foreground and thus background subtraction by itself does not reliably handle shadows.

Kilger extended this method in [4] by assuming that stronger edges exist between cast shadows and their objects than between cast shadows and the background. By narrowing the region of interest to a vehicle and its cast shadow, edge detection can determine where the boundary of the car ends and the shadow begins. While this system works well, it requires the explicit knowledge of the position of the sun. Also, its performance is best when the direction of travel was aligned with the edge filters. Thus, it was not robust enough to handle a variety of traffic scenes.

Another method that is gaining acceptance is using a probabilistic model to determine whether a pixel belongs to a certain group, i. e., background, foreground, or shadow. Friedman introduced this in [2] and used an expectation-maximization algorithm to model a mixture of Gaussians. This approach eliminates the explicit need to find a background image. Because of this, this system does a better job at handling slow moving objects, which tend to get included in the background when computing the average. This system succeeded in robust segmentation and also was able to detect shadows.

One other system worth mentioning, SAKBOT (Statistical and Knowledge-Based Object Tracker), was discussed in [7]. It took advantage of the chrominance information of shadows as well as the effects of shadows in the HSV color space. SAKBOT was then able to use this information to define a shadow. The shadows were then passed to the vehicle detection program which suppressed them to get better detection.

II. Method

Intrinsic images is a term first coined by Barrow and Tenenbaum [1]. It referred to the concept that an image can be decomposed into two parts: a reflectance image, which stays constant under illumination changes, and an illumination image, which represents the lighting of an image. That is, $I(x,y) = L(x,y)R(x,y)$. Working in the log domain, this equation becomes $i(x,y) = l(x,y) + r(x,y)$. They argued that intrinsic images could be extremely useful for such tasks as segmentation, shape-from-shading, and template-matching. However, finding such images is not easy since there are twice as many unknowns as equations. While there have been some solutions to special cases of this problem, it remains generally unsolved.

Weiss [8] considered this problem with a sequence of images from a stationary camera. To find the intrinsic images, he used the fact that natural images have an inherent statistical structure. When derivative filters are applied to natural images, the outputs tend to be sparsely distributed. The outputs are better fit by a Laplacian distribution than

Fig. 1. A frame from our traffic sequence and the resulting histogram of the filter output. The Gaussian has unit variance and has been scaled and shifted to coincide with the peak of the histogram.

a Gaussian one. Therefore, given a set of filter outputs from a sequence of images, we can find the maximum likelihood image by taking the median over the filter outputs. Figure 1 shows a typical scene from our traffic analysis and its resulting histogram from the output of a horizontal derivative filter. A Gaussian curve is shown for comparison.

Given a set of N filters, $f_n,\ 1 \le n \le N$, we denote the filter outputs of an image i as $i_n = f_n \star i$. Then, we can likewise denote the filtered reflectance images, r_n, in the same manner: $r_n = f_n \star r$. Our set of N filtered reflectance images is then given by:

$$\hat{r}_n(x, y) = median_t\ i_n(x, y, t). \tag{1}$$

For each filter, there now exists a median estimate of the true reflectance image.

$$f_n \star \hat{r} = \hat{r}_n \tag{2}$$

Our estimated reflectance image, $\hat{r}$, is then:

$$\hat{r} = g \star \left(\sum_n f_n^r \star \hat{r}_n \right) \tag{3}$$

where f_n^r is the reversed filter of $f_n : f_n(x, y) = f_n^r(-x, -y)$. Solving for g, we get:

$$g \star \left(\sum_n f_n^r \star f_n \right) = \delta, \tag{4}$$

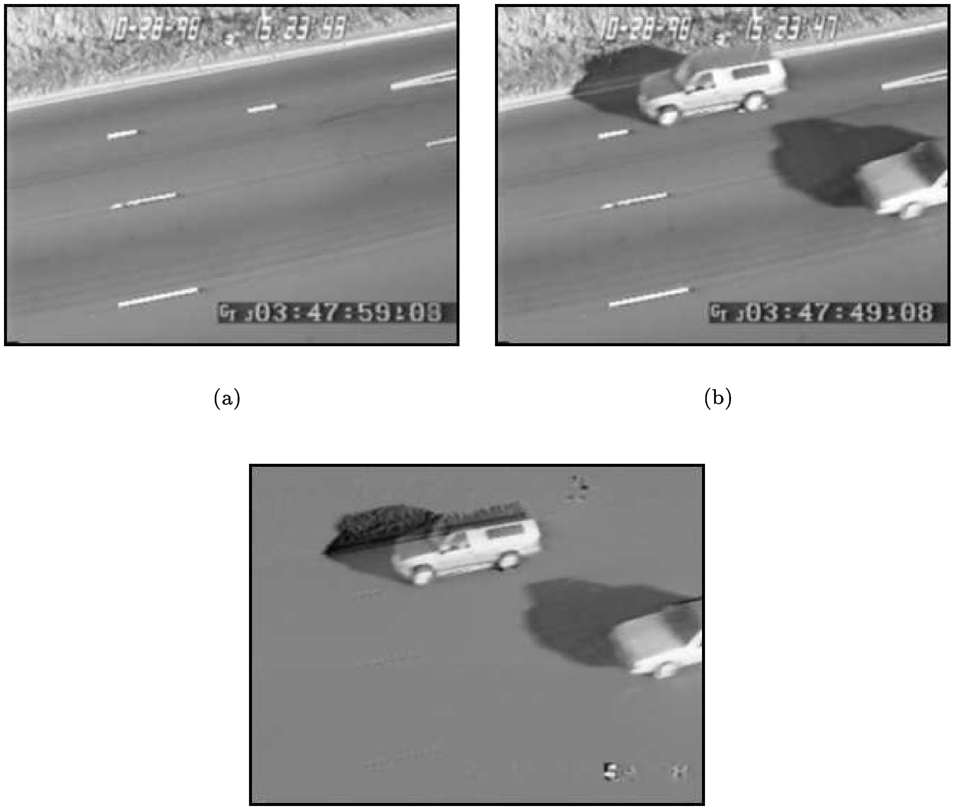
where δ is a the delta function. We can compute g when we define our filters since Equation (4) does not depend on any input images.

We now have a method to recover the reflectance image from our video sequence. In the interest of computational efficiency, we handle the convolutions in the Fourier domain. Since we assume a stationary camera, we only need to use the first few frames of the sequence to determine the reflectance image. In our case, in a sequence of 20 images, any 5 of them worked equally well. We do not need to continue to update our reflectance image since it is invariant to illumination changes.

To perform segmentation, we simply find the illumination difference between the current frame and the reflectance image. Vehicles will not appear in the reflectance image and they will therefore appear in the segmented foreground. To find the illumination difference, we subtract the filter outputs of the current image from the filter outputs of the reflectance image. By applying Equation (3) to our filter difference, we can recover the illumination image. A recovered reflectance image and an image from our input sequence and its corresponding illumination image are shown in Figure 2.

The histogram of the illumination image is peaked like the filter outputs, i. e., its shape more closely resembles a Laplacian distribution. This makes sense since there are either relatively small changes in illumination or only a few foreground objects in the current frame. The majority of pixels in the illumination image represent no change from the reflectance image. The tails of the histogram can be considered the foreground; some darker than the reflectance image, some lighter. For a first approximation, we can consider all pixels with lower intensity than the peak to be shadow, and those with higher intensity vehicles. This may cause some problems since some vehicles are darkly colored although the effects are not as severe as one would anticipate.

To segment the foreground into vehicles and their cast shadows, we use the peak of the histogram as the starting point of our boundary. We then expand our boundaries by some ϵ in both directions. We experimented with a number of values and found that we could get reasonable results when the cutoff points are between 1–20% of the peak value.



(a) (b)

(c)

Fig. 2. a) The computed reflectance image, b) a frame from our traffic sequence, and c) the illumination image of the frame.

III. Results

Figure 3 shows several output frames from our initial sequence. Detected shadows were masked out and appear as large black blobs in the figures. Our program was successfully able to detect and remove shadows in all images in our sequence. We set our segmentation boundary to be 10% of the peak histogram value and having a static limit did degrade our results somewhat. However, even in the difficult situations such as those with overlapping shadows, our system performed without fault. Obviously, a more intelligent selection of the boundary would enhance the performance. Perhaps coupling the output of the shadow detection with the probabilistic methods of background estimation would provide even better results.

IV. Conclusions and Future Work

Shadow handling in video sequences is an important yet extremely difficult process. In this paper we have shown a novel method for handling shadows in traffic scenes. The use of intrinsic images has allowed us to analyze traffic from an arbitrary viewing angle as well as to handle overlapping shadows. Our method robustly finds cast shadows in the scene while allowing the images of objects casting the shadows to remain uncorrupted, which can then be used for further processing.

The computations performed in our algorithms are commonly performed by many commericially available graphics processing boards. It is realistic to expect that this algorithm can be implemented in real-time. It is also hoped to develop this algorithm into a more generic segmentation process which can be used for a larger set of applications.

Acknowledgements

This work has been supported in part by the Minnesota Department of Transportation, and in part by the National Science Foundation through grant # CMS-0127893.

References

[1] H. G. Barrow and J. M. Tenenbaum, "Recovering intrinsic scene characteristics from images", *Computer Vision Systems*, Academic Press, 1978.

[2] N. Friedman, "Image segmentation in video sequences: a probabilistic approach", *Proc. of the 13th Conference on Uncertainty in AI*, 1997.

[3] S. Gupte, O. Masoud, R. F. K. Martin, N. P. Papanikolopoulos, "Detection and classification of vehicles", *IEEE Trans. on Intelligent Transportation Systems*, Volume: 3, Issue: 1, pp. 27-47, 2002.

[4] M. Kilger, "A shadow handler in a video-based real-time monitoring system", *Proc. of IEEE Workshop on Applications of Computer Vision*, pp. 11–18, 1992.

[5] D. Koller, J. Malik, "Robust multiple car tracking with occlusion reasoning", *Proc. Third European Conference on Computer Vision*, pp. 189–196, 1994.

[6] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code", *Nature*, 381: 607–608, 1996.

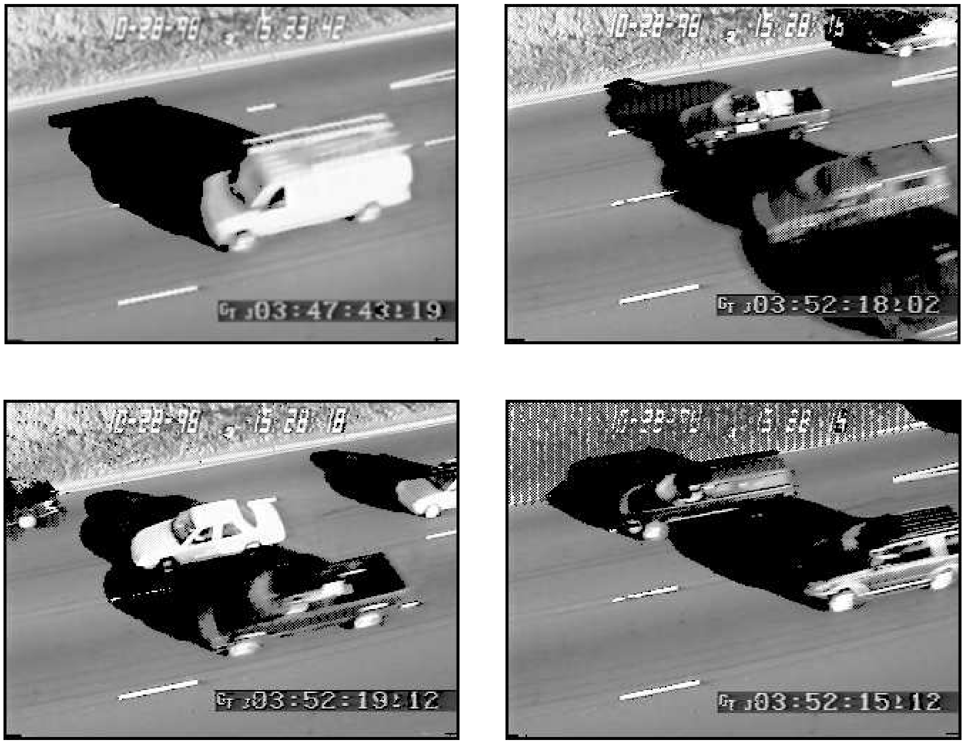


Fig. 3. Results from our shadow removal. Shadows have been shaded a flat black.

[7] A. Prati, I. Mikic, C. Grana, M. Trivedi, "Shadow detection algorithms for traffic flow analysis: a comparative study", *IEEE Intl Conf. On Intelligent Transportation Systems*, Oakland, California, Aug 2001.

[8] Y. Weiss, "Deriving intrinsic images from image sequences",*Proc. of the 8th IEEE International Conference on Computer Vision*, Volume: 2, pp. 68–75, 2001.