

Robust denoising and moving shadows detection in traffic scenes

A. Bevilacqua

DEIS - Department of Electronics,
Computer Science and Systems
University of Bologna
Viale Risorgimento, 2,
Bologna, ITALY 40136

M. Roffilli

DEIS - Department of Electronics
Computer Science and Systems
University of Bologna
Viale Risorgimento, 2
Bologna, ITALY 40136

Abstract

This paper presents a work we have done on the motion detection in the context of an outdoor traffic scene for visual surveillance purposes. Our motion detection algorithm is based both on background subtraction and three frame difference. We propose quite innovative solutions for denoising, blobs filling and shadow detection without exploiting any a priori knowledge. Actually, methods presented here have been fully setup only for the former technique. Target sequence is made of 8-bits grey level still images, taken at 30 fps. This application works off line at 8 fps on a 800 MHz Pentium III computer.

1. Introduction

Visual surveillance applications mostly rely in their first processing step regarding some kind of moving objects detection. Our research focusses on outdoor grey level image sequences taken by one stationary camera, with a fixed focal length and a high depth of field. The final goal is to develop an effective visual surveillance application, in spite being used as general purpose. This should work at a high frame rate and should exploit the simplest image processing operations.

Our application fulfills the following three criteria. First, no a priori knowledge is used in order to detect foreground objects. Second, all foreground objects are detected with high definition, i.e. we use contour lines instead of bounding boxes to highlight them. Third, shadows should be removed without exploiting any a priori knowledge as well, for example their direction.

Typical robust visual surveillance applications use frame differencing techniques in order to detect unusual motion. In this case, we are devising an hybrid algorithm based both on background subtraction and temporal frame difference. The outcome of this algorithm presents a lot of false signals both in terms of noise and non-interesting foreground, such as shadows. Our main interest, during this stage, is

concerned with the *false positive reduction* step, so that the overall computation does not become heavier.

Therefore, we setup two novel methods, actually fully working just on the outcome of the background subtraction technique.

The first algorithm accomplishes in *one step* both denoising and blobs filling. This reduces the final number of false detected blobs by achieving, in the meantime, a very high definition of the correctly detected objects. For this purpose, the algorithm *finds out* and exploits some structural features of the objects.

The second method deals with moving shadow detection and its removal. It is made of two parts, acting independently to each other. By exploiting foreground photometric properties, each of them is able to discover shadows showing different features. The operations are gradient-based and basically rely on the assumption that shadows due to different objects keep some properties across the frames.

Actually, our system is able to detect up to 20 blobs at 8 fps on an entry-level PC. The target sequence has been taken from a daytime (and sunny!) traffic scene. The algorithm, running both on Unix-like and windows OS's, has been fully written in ANSI C.

1.1. Previous Work

There are diverse examples of visual surveillance system for outdoor scenes, involving vehicles or pedestrians. One of the most famous is probably VSAM [1]. That system, anyway, does not deal with shadows. In addition, images that authors use come from a relatively low depth of field. In [2], authors show examples of *one* pedestrian's shadow detection, with no depth of field. The system described in [3], removes shadows of two pedestrians but uses stereo-based detection. At last, [4] use photometric properties we will call transparency and homogeneity. But some of their a priori assumptions which regard shadow regions induce their method to detect only shadows having a quite large area compared with the objects itself.

2. The Visual Surveillance System

Since the purpose of our system is to monitor the traffic, the starting point is to separate vehicles and pedestrians from the background. A common way is to use image difference: pixel-by-pixel difference between subsequent images, or between each image and the background, is performed so as to identify regions where the grey level changes mostly with respect to a threshold value.

In order to perform motion detection, we have developed an hybrid algorithm, through combining background subtraction and three-frame difference techniques. Both these methods suffer from *waving trees* problem. In any case, as discussed in [5], one of the major drawbacks of background subtraction technique is due to the fact of the *waking persons*. On the other side, the main problem with the temporal frame difference rule is the *foreground aperture*.

Nevertheless, since it is the most appropriate input in order to better understand and appreciate the methods we present, at this point here we consider only the outcome of the background subtraction operation. Where it is feasible to have the background itself at disposal, the background subtraction method is suitable in order to detect foreground objects.

We have used a background which has been extracted through exploiting statistical properties of the first 20 frames and have maintained it up to date over the remaining frames. So, this method can be employed even though a training period without foreground objects does not allow us to obtain a “directly measured” background.

Let us consider a sequence from a stationary camera. Let $I_n(x)$ represent the intensity value of the pixel x in the current frame and $B_n(x)$ the extracted background, both at time $t = n$ (see (Eq.1)). Background subtraction is performed by marking any pixel x resulting from the absolute difference that is more than a prefixed threshold, as a foreground pixel $F_n(x)$.

$$F_n(x) = \begin{cases} 1 & \text{if } (|I_n(x) - B_n(x)| > T_F) \\ 0 & \text{else} \end{cases} \quad (1)$$

At the moment, the threshold value T_F is kept constant to 12. Fig.1 at left shows the original frame with the moving objects which have been contoured. The image at right in Fig.1 shows the outcome of the background subtraction operation. It has been obtained by relaxing the threshold operation in (Eq.1). For this reason, it has such a noisy appearance. Therefore, the subsequent stage includes a denoising operation in order to separate interesting signals (namely, foreground regions) from this *salt and pepper* noise. Besides, the remaining regions should be connected in logical objects (*blobs*) and their borders should be extracted.

As shown in the next Section, our system is able to cope with such noisy images through performing denoising and

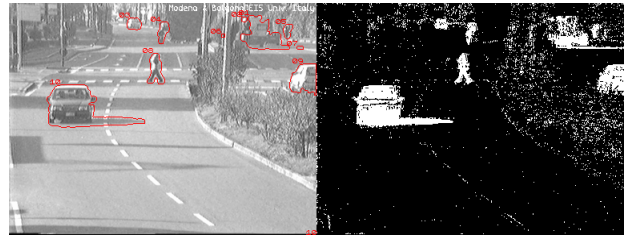


Figure 1: Original frame (left) and binary result after the background subtraction operation (right).

connecting operations in one step. This is achievable by applying a novel algorithm which yields to better results with respect to the outcome of the commonly used *erode* and *dilate* operations.

After denoising and connecting, blobs represent the foreground regions. On the other hand, blobs may be made by objects and their shadows. Therefore, the last step consists in exploiting both some intuitive information and the spatial knowledge collected so far about foreground regions, so as to find out and remove moving shadows.

The traffic sequence we are studying has been taken at 30 fps and is of 210 frames. Images are 8-bits, grey level, with resolution of 384×288 . The camera was mounted on a tripod placed on a bridge, so the background is static. The algorithm has not been optimized till now and it operates off line on an entry-level Pentium III PC, 800MHz, 512 MB RAM, at 8 fps. It has been fully written in ANSI C and works under Windows, Solaris and Linux OS's.

2.1. Structural Analysis for Denoising

Usually, in outdoor environments false detected signals are due to different reasons (Fig.1, right). For example, a random noise which should be homogeneous over the whole frame and in addition, small movements in the scene background, such as moving trees, or camera displacements due to wind load.

One of the most used methods to remove noise consists into applying, one or more times, morphological operators such as erosion, dilatation, opening and closing to the binary image. This approach reveals three drawbacks. First, it is quite difficult to find out the right value for the kernel size and the operations' sequence. Second, small foreground objects may be eliminated. The third and more important disadvantage is that morphological operations may often result into the object's shape and contour distortion.

Our method aims to give a measure of *how much* a pixel belongs to a structural windowed region around it. Then, by means of local thresholding the interesting signal is extracted. The first step is to define the basic structure we intend to find (Fig.2.a). In case of binary image, we perform a logical “AND” between the pixel pointed by the circle and

each one of its three neighborhoods. We then look for the basic structure in each direction (i.e., horizontal, vertical, Fig.2.b). For example, the kernel of Fig.2.b is applied to the 9 different sub-windows of Fig.2.c. Finally, the value of the central pixel (the black pixel in Fig.2.c) is increased by the outcome of all these “AND” operations. In Fig.3 we show

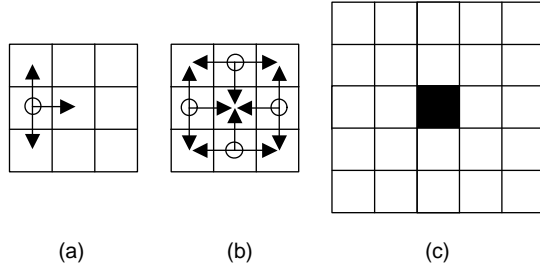


Figure 2: Structuring elements ((a), (b)) used to analyze the neighborhood structure ((c)).

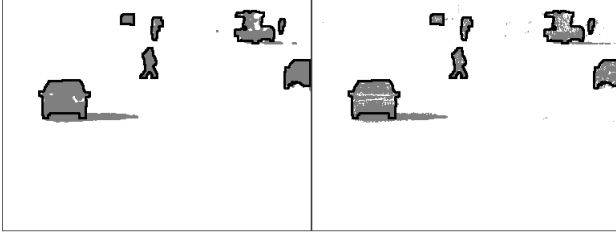


Figure 3: Original frame after our structural analysis (left) and the best opening operation with an appropriate kernel (right).

a comparison between the original frame after our structural analysis (left) and the best opening operation with an appropriate kernel (right). Even though the quality of the printed figures does not allow to fully appreciate results, we see that our method eliminates isolated noise in a better way and in the meantime reconstructs blobs’ inner structure in a better way.

Our method is a sort of region growing algorithm which propagates its structure. This leads to have fuller blobs with *well defined*, and especially *continuous*, edges. At last, this allows to extract the chain code representing borders without any further heavy computational step, such as the N -way connection.

2.2. Shadow detection

Many algorithms detecting shadows take into account a priori information, such as the geometry of the scene and of the moving objects and the location of the light source. We aim to avoid using any such knowledge in detecting shadows.

Nevertheless, we exploit the following sources of information.

First, moving shadows in each frame are connected to their respective objects for the most time - this involves spatial information. *Transparency*: a shadow always makes the region it covers darker - this involves the appearance of single pixels. *Homogeneity*: research in [4], [6] state that the relationship between pixels when illuminated and the same pixels under shadows is roughly linear; namely, this ratio is almost 2 - this also involves the appearance of single pixels. Finally, a *small* and *narrow* strip centered on a pixel belonging to a shadow border always pertains to two adjacent regions referring to the same object - this is concerned with both pixel appearance and spatial information.

We setup two different methods. The first algorithm aims to find quite large shadows, even when they join two different objects belonging to one blob. The second one is able to find out shadows even when they are small and narrow. The outcome of these two algorithms are finally OR’ed.

The first step is common to both methods and consists in calculating for each pixel (within the binary detected blobs) the intensity ratio D_S between the background B_S and the current frame F_S , after they have been smoothed:

$$D_S = B_S / F_S \quad (2)$$

The ratio itself is further smoothed. The former method aims to define the most likely shadow regions. This is accomplished by studying the global histogram of the blobs pixel values in a sample of 20 frames of D_S (2), where blobs are the ones attained by the structural analysis stage. Actually, we multiply the numerator by a constant factor so as to increase the scale sensitivity of result and thus make the threshold operation easier to perform. Inside these blob-defined areas, three gradient operators (i.e., horizontal, vertical, oblique) are applied in order to find roughly homogeneous regions. Again, the global histogram has been studied to find the right threshold for gradient operations. Since shadows are regions with similar values in D_S , whilst the objects are usually composed of significantly different grey levels, the resulting histogram shows a robust peak in correspondence of shadows. This leads to reliable threshold values. The outcome of this method is shown in the image at left in Fig.4.

The second algorithm is based on the edge gradient operation and exploits the last information described above. If we consider a small orthogonal strip of the shadow border, it always divides *one* object. We compute the edge gradient along the blob borders in D_S , by using a 5×2 sized mask, always normal to the edge. By thresholding the histogram of the outcome of this gradient operation, we divide the blob border into many segments of different length. Since the longest continuous lines are in correspondence of shadows, our hypothesis has been confirmed. We have therefore de-

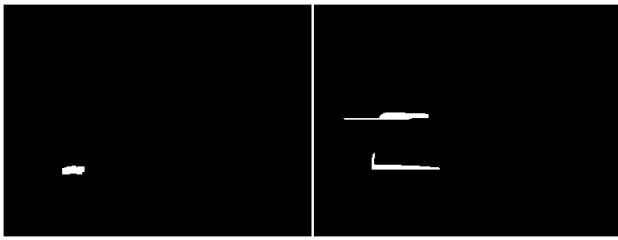


Figure 4: The outcome of the first method (left) and the edge gradient method (right).

veloped an algorithm in order to clean isolated points or small segments. In addition, when segments are sufficiently near to each other, they are joined. Usually, we obtain one open line segment for each shadow. The lines are closed by following the D_S blobs' inner gradient, which defines the boundary between object and shadow. The closed regions so obtained are then filled (Fig.4, right). The area resulting from both methods is finally taken away from blobs previously obtained.

3. Summary and Conclusions

This paper presents the work we are accomplishing for the motion detection regarding visual surveillance purposes. The target sequence is a daytime traffic scene involving both pedestrians and vehicles. The final result of our system is shown in the image at right in Fig.5. A total of 14 blobs is detected, at 8 fps. All moving objects are correctly identified and all shadows (here related to blobs 11 and 12) have been removed.

In spite of the fact that the depth of field makes the threshold operation on the blobs' area quite unfeasible, our structural analysis method achieves a high specificity: for example, in case of frame of Fig.5, all detected blobs are really moving objects. In addition, the high definition fea-

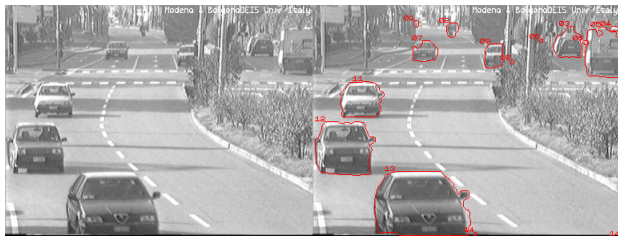


Figure 5: The original frame (left) and the final result of our system (right).

ture of the method allows to detect two blobs separately, which are very close (3 and 4, on the upper right side of Fig.5, right). At last, all blobs are indicated by contour lines, rather than by bounding boxes.

The method we are building in order to detect and with the intention to remove moving shadows is positively working on different kinds of shadows. It exploits simple assumptions and properties and it is based on simple gradient operations. Hence, the overall computation is not heavy.

Actually, we are extending the structural analysis to grey level images with the purpose of exploiting structural (and, now, textural) shadow properties, in order to remove them. We are also finishing the application of these methods to the temporal frame difference. Very soon, we are going to start the tracking stage, convinced that the high definition results of the present step may give us some support.

Acknowledgments

We wish to thank Prof. Giorgio Baccarani and Prof. Riccardo Rovatti for their interest in the present work and for useful discussions. We also desire to thank Mr. Alessandro Antonioli and Mr. Giuseppe Giorgio for their gentle support in developing the C code.

References

- [1] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson, "A System for Video Surveillance and Monitoring," *Robotics Institute, Carnegie Mellon University, Technical Report, CMU-RI-TR-00-12*, 2000.
- [2] A. Elgammal, D. Harwood, L. Davis, "Non parametric model for background subtraction," *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, June*, pp. 37-45, 2000.
- [3] I. Haritaoglu, D. Harwood, L. S. Davis, "W4S: A Real Time System for Detecting and Tracking People in 2.5 D," *Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany, June*, Vol. 1, pp. 877-892, 1998.
- [4] P. L. Rosin, T. Ellis, "Image difference threshold strategies and shadow detection," *Proceedings of the 6th British Machine Vision Conference, Birmingham, UK, September*, pp. 347-356, 1995.
- [5] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," *IEEE Proceedings of the Seventh International Conference on Computer Vision, Corfu, Greece, September*, Vol. 2, pp. 255-261, 1999.
- [6] I. Mikić, P. C. Cosman, G. T. Kogut, M. M. Trivedi, "Moving Shadows and Object Detection in Traffic Scenes," *Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, September*, Vol. 1, pp. 321-324, 2000.