Effective Object Segmentation in a Traffic Monitoring Application

Alessandro Bevilacqua, Member, IEEE
ARCES – DEIS (Department of Electronics, Computer Science and Systems)
University of Bologna, Viale Risorgimento, 2, Bologna, ITALY 40136
abevilacqua@deis.unibo.it

Abstract

The main task of traffic monitoring applications is to identify (and track) moving targets. Thresholding an image resulting from a background difference operation is a common way to detect moving pixels and represents the starting point for all of the subsequent operations. Therefore, a wrong choice for the threshold could afflict final results. The morphological operations utilized within most of the traffic monitoring systems to remove noise dramatically undergo the effect of a "wrong" thresholding. Namely, sometimes wrong thresholded images do not allow morphological operators to reconstruct the whole signal. Besides, a fine tuning of the parameters related to morphological operators is often required.

The segmentation method we present in this work has been used within the traffic monitoring system we are developing. It is based on an original morphological operation which takes advantage of all the true signals which pass through a low threshold, without being heavily afflicted by the inevitable huge amount of noise. The decision of removing, or preserving, a connected component is based on whether a given object is structured according a given pattern. The operator we developed accomplishes this task in an original way. Last but not least, the simplicity required to tune the few parameters related to the morphological operator encourages its use.

1. Introduction

The segmentation technique we describe in this paper is actually used within the traffic monitoring system we are developing ([1], [2]). To have correctly segmented moving *blobs* (a sort of coherent connected regions, sharing common features) represents a key issue in all the visual surveillance system. In fact, a weak segmentation step could affect the subsequent stages of feature extraction and tracking. In our system, after that a background has been generated during a bootstrap phase ([3]) and the arithmetic subtraction between the reference background and the current frame has been performed, one suitable threshold must be chosen and applied. The resulting image will constitute the input for the subsequent stage of segmentation. Therefore, the tasks of choosing a threshold and the proper segmentation method are highly correlated.

The major drawback of threshold-based approaches is that they often lack finding the best separation between true positive signals and false positive signals (noise). Namely, if the threshold is kept too low, a lot of true positive signals maybe are not detected. On the opposite, an excessively high value includes most of the moving pixels together with a lot of noise.

The purpose of the novel operator we setup is to detect connected objects on the basis of the criterion that a given structure must fit inside the object. This is achieved by exploiting at their best the amount of true signals emerged from the threshold operation. In addition, this allows keeping the threshold quite low in order not to miss too many true signals and obtaining a very good definition of extracted blobs. In particular, the decision of preserving a "structured" component is based on a measurement criterion which we called "the fitness" of the operator.

This paper is organized as follows. In the next Section an introduction concerning the most utilized morphological operators is presented. In Section 3 we review some segmentation method used within a few visual surveillance systems. In Section 4 a detailed description of our segmentation method is given, as well as our original structural analysis operation is thoroughly described. Experimental results are shown in Section 5 and Section 6 draws conclusions and future works.

2. Morphological Operators

Before either reviewing some other work or analyzing the segmentation stage we perform, we guess that some basic principle regarding the morphological operations must be explained.

The field of mathematical morphology contributes to a wide range of operators to image processing; they are all based around a few simple mathematical concepts belonging to the Set Theory ([4]). Here, we do not describe this theory, we only give some basic principle of its application.

Actually, we are only interested in handling binary im-

ages. For a binary image, black pixels ("0") are normally taken to represent background regions, while white pixels ("1") denote foreground. Therefore, all of the examples shown in this paper will be based on this assumption.

The two most basic operations in mathematical morphology are *dilation* and *erosion*. These operations can be considered as morphological non-linear filters. Both of the involved operators take two inputs: an image to be dilated (or eroded), and a *structuring element (SE)*(Figure 1). The SE

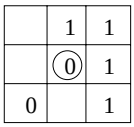

Figure 1: Example of a structuring element, with the origin marked by a circle

is to mathematical morphology what the convolution kernel is to Filter Theory. It consists of a *pattern* specified as the coordinates of a number of discrete points relative to some origin (in Figure 1 the origin is marked by a ring around that point). Normally, Cartesian coordinates are used and so a convenient way of representing the element is as a small image on a square (or rectangular) grid. Actually, a 3×3 grid with its origin at the center is the most commonly seen type. An important point is that not every cell in the grid is part of the SE in general. In fact, we must not forget that the element represents the pattern we are looking for within the image.

When a morphological operation is carried out, the origin of the SE is typically translated to each pixel position in the image in turn, and then the points within the translated SE are compared with the underlying image pixel values. The details of this comparison and the effect of the outcome depend on which morphological operator is being used. Sometimes this operation is performed *like* the convolution operation, thus SE's are also called *kernels*.

Let us go back to the concepts of dilation and erosion. The amount and the way that they grow or shrink depends upon the choice of the SE. Dilating or eroding without specifying the SE makes no more sense than trying to lowpass filter an image without specifying the filter. In addition, other mathematical morphology operators can be defined in terms of combinations of erosion and dilation. The most important are *opening* and *closing*.

To conclude, this work could seem to a perfunctory reader related to the field of statistical morphology ([5]). However, it is worth remarking that while in statistical morphology the center of the SE is considered to be "a winner" according to a probability value, in our approach the winner is deterministically established on the basis of how much a well-defined simple small structure may be considered belonging to a wider complex "physical" structure.

3. Previous Works

Applications of mathematical morphology to object segmentation has been proposed for many years, therefore a huge amount of works exists in this field. However, most of the works accomplished in sequence analyses, within traffic monitoring or visual surveillance applications, only deal with morphological opening and closing operations. This due mainly for the simplicity of these two operations, which have as a drawback a low precision. This is yet more evident in presence of very noisy images, like ours.

The system W^4 described in [6] works on a binary image stemming from a thresholding operation on a background difference image. First, one iteration of erosion is applied to remove one-pixel noise. Then a fast binary connected component operation allows to remove small regions and to find likely foreground regions, which are further enclosed by bounding boxes. In order to restore the original size of the objects, a morphological opening is applied. After reapplying a background subtraction and a size thresholding operation, a morphological closing is performed only to those regions which are enclosed by the bounding boxes. Authors met with great difficulties the right combination for the morphological operations and made this system result in a quite scene-dependent application.

In [7, 8] authors use three frame differencing until a background model is stabilized. After that, the background subtraction technique is used and a thresholding operation permits to obtain moving pixels. In all the cases, moving pixels are grouped by means of a connected component approach. Two iterations of morphological dilation and one erosion step are performed in order to reconstruct incomplete targets. Noise is removed by filtering the size of the pixels' area. As a result, blobs are extracted with a rough definition.

In order to segment moving regions, authors in [9] start with the three frame differencing approach, followed by a thresholding operation. A size filtering operation follows in order to clean small spots due, for example, to small movements of the sensors. Further, the contour enclosing moving objects is defined by means of a morphological closing. Blobs thus extracted show a bad definition.

4. Blob Segmentation

The segmentation step we describe is referred to the traffic monitoring system we are developing. It relies on a stationary video camera or, at most, on a camera moving in a "step and stare" mode. The algorithm processes one frame at a time and it gives the segmented interesting blobs as the final output, which here are made of vehicles, humans, shadows or all of them. The outline of the blob segmentation algorithm is described in Figure 2. This stage takes the *background* and the *current frame* (a sample is shown Fig-

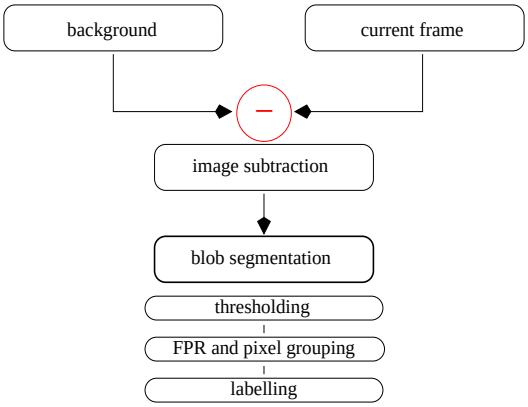

Figure 2: General scheme for the segmentation algorithm

ure 3) as input images. After performing the background

Figure 3: A sample frame extracted from our test sequence

differencing and a thorough analysis in order to determine a correct *threshold* T_B, the algorithm produces a noisy image (Figure 4) which retains most of the true moving pixels together with false signals due to noise and uninteresting moving objects, such as hedges and trees. These signals must be removed and the shape of interesting moving objects must be "extracted". Removing these signals has been often called in the image processing community the *False Positive Reduction (FPR)* step. Here, this has been accomplished by using our morphological operations thoroughly described in the next Section. Basically, we look for a predetermined structure within the scene, so that blobs fitting that structure can be "reconstructed" and in the meanwhile noise can be removed since it does not fit the same structure.

At last, all of the regions must be *labeled* so as to make them feasible to be distinguished from each other.

4.1 Structural Analysis

For every moving blob detected through the image subtraction operation, there are a lot of signals which alter the right

Figure 4: The binary result after thresholding the outcome of the background subtraction operation ($T_B = 12$)

perception of the blob itself.

The method we have developed aims to give a measure of *how much* a pixel belongs to a structural windowed region around it, thus resulting in a very effective FPR step. The operation we perform acts in a slightly different way with respect to the ones employing the "classic" morphological operators described in Section 2. In fact, we introduce the concept of *fitness* of the pixel at the center of the SE in respect of the pattern it should belong to. The first step is to define the basic structure we intend to address. Figure 5(a) shows the basic structure and the *compound* structure (b)

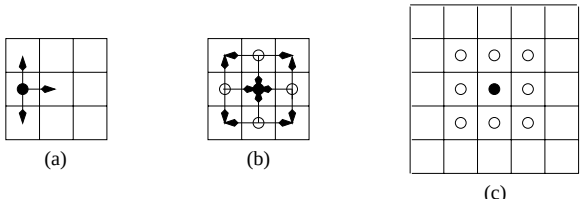

Figure 5: Structuring elements: basic (a), compound (b) and cell-based (c)

we use. The latter is obtained by rotating the former by $90°, 180°$ and $270°$. This is as to say that the basic structure is searched by considering every spatial arrangement. In addition to these two structures, we define a *cell-based* structure (Figure 5(c)). It is built through stemming from the compound structure (b) the same as (b) has been built starting from (a). But (b) is symmetric; thus (c) is formed basically by the set of all possible occurrences of the compound structure. Namely, in the example of Figure 5 the cell-based element (c) is composed by 9 compound (*cell*) elements (b), whose centers are the white circles plus the black circle.

How does this method exactly work? In our implementation, all the pixels of the elements involved in (a), (b) or (c) are assigned "1". In case of the basic structure (Figure 5(a)),

a logical AND between the pixel pointed by the circle and each one of its three neighborhoods is performed. The arithmetic sum of these three partial results represent the fitness of the pixel pointed by the circle (therefore, the fitness maximum value is 3). Further, a hard threshold on this fitness value allows the pixel to be assigned "1" or "0"; this occurs whether the fitness is greater or less than the threshold, respectively. In case of the compound structure (Figure 5(b)), this procedure is accomplished for four times, one for each possible position of the basic element (a) within the compound element (b). Unlike what we have made before, the partial fitnesses computed for the pixels pointed by the white circles are summed to each other instead of being assigned to the pixel. Here, the fitness maximum value can go up to $3 \times 4 = 12$, in case of all the underlying image pixels hold "1". The outcome of the threshold operation performed on the total amount of fitness is finally given to the pixel corresponding to the center of the structure (the black circle). At last, for the cell-based structure (Figure 5(c)), first we compute the fitness for each cell and then the overall fitness is assigned again to the central pixel pointed by the black circle. Here, the fitness maximum value M_F is given by Expression 1:

$$M_F = 12 \times (l_{cb} - l_c)^2 = 12 \times (5-3)^2 = 108 \qquad (1)$$

where l_{cb} and l_c represent the basic and the compound element side length, respectively, and $(l_{cb} - l_c)^2$ gives the number of possible positions of (b) within (c).

There are two ways of performing the above morphological operation. In an its early version ([1]), the operator "switched on" dark pixels belonging to the desired structure, thus resulting essentially in a "smart" dilation. In addition, it "switched off" white non-structured pixels, which are probably nothing else but noise. Often this approach enlarged a blob in correspondence of its borders, thus resulting in an overall loss of resolution.

Actually, we realized that using a more noisy image and only switching off pixels not belonging to any structure better preserves both blob shape and border. In this way, the operator performs a "smart" erosion, namely, only noisy pixels are removed from the image. The improvement achieved can be appreciated when comparing Figure 6 and Figure 12, each one representing a significant output frame of the segmentation stage by applying the former and the latter approach, respectively. When employing the former approach, the loss in terms of blob resolution yields three distinct blobs to be merged and detected as though they were one.

5. Experimental Results

The input of the system is constituted by a 210 frame gray level sequence representing a daytime traffic scene, with 384×288 frame size and working at 10 Hz. In all the experiments we use as a compound element the one shown in Figure 5(b). The overall motion detection algorithm has been written in C and works under Windows, Solaris and Linux OS's.

Figure 6: An output frame of the segmentation stage, where the blobs have been labeled. It is related to an earlier version of the method

Fundamentally, once the basic SE has been defined, two more parameters have to be set. The first is the size of the cell-based element, the second is the threshold T_F for the fitness. The first parameter is strictly related to the threshold T_B applied to the background difference operation. In fact, a very low value for T_B yields zones with a high density of noisy pixels (Figure 7) which could mislead the morphological operator in case it looks for quite small structures. Practically speaking, the *size* of the cell-based element determines the minimum value of T_B that leads the possible detected false blobs not to be comparable *in size* with the smallest true blobs we want the system to reveal. In fact, we see in Figure 8 a lot of small false blobs which have been introduced by a too low value of T_B whit respect to the "small" size (9×9) of the cell-based element utilized.

Figure 7: The binary result of the background differencing operation by using a relaxed threshold $T_B = 8$

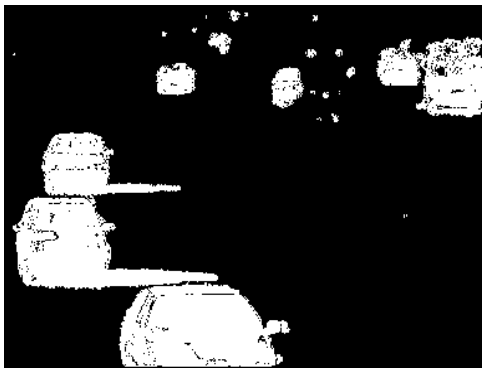

Figure 8: The outcome of the structural analysis performed on the image of Figure 7 by means of a 9×9 cell-based element with $T_F = 205$

As a matter of fact, in Figure 9 a higher size (11×11) for the cell-based element reduces the number of false blobs

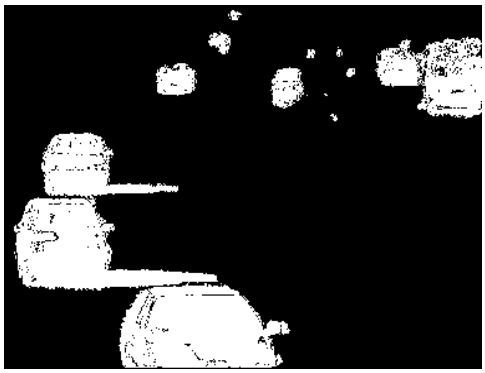

Figure 9: The outcome of the structural analysis performed on the image of Figure 7 by means of a 11×11 cell-based element with $T_F = 320$

present in Figure 8.

The second parameter, the threshold T_F, affects the *density* of the structure. Namely, in order that a pixel may be considered belonging to a structure, its fitness must exceed T_F. Therefore, a low value for T_F yields a dense area to be considered as a structure, whether it is a real structure or also noise. As a direct consequence we attain a high number of false blobs and, above all, dirty blob border. These effects can be fully appreciated when comparing Figure 10 and Figure 11 that represent the outcome of the structural analysis performed on the image of Figure 4 by means of a 9×9 cell-based element, with $T_F = 100$ and $T_F = 205$, respectively.

Let us go back to our application. Figure 11 also represents the true outcome of the FPR step performed in our system. With this value for T_F and T_B all the noisy pixels of Figure 4 have been cleaned. In addition, also the structures

Figure 10: The outcome of the structural analysis performed on the image of Figure 4 by means of a 9×9 cell-based element with $T_F = 100$

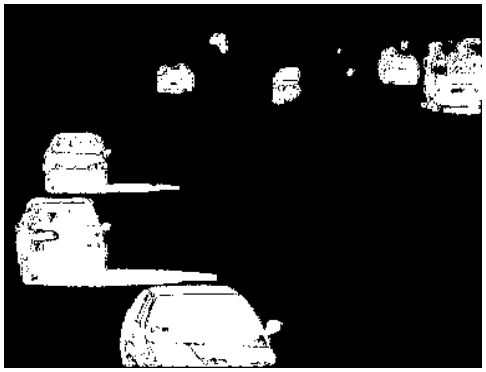

Figure 11: The outcome of the structural analysis performed on the image of Figure 4 by means of a 9×9 cell-based element with $T_F = 205$

not completely visible at a glance have been successfully preserved. It is worth remarking that since a noisy image also contains a lot of useful pixels, the blobs detected are rather dense. This allows not applying heavy morphological operations in order to fill the blobs, because they are basically already connected, thus preserving the blob's border, hence its definition.

Figure 12 shows a significant output frame of the segmentation stage, after that a morphological closing on the image of Figure 11 has been performed by using a 3×3 kernel. We can appreciate how this operation does not introduce any relevant change to the overall shape of each blob. This is yet more evident for the small (red) blob in the upper side of the image.

This result can also be compared with the "ground truth" (Figure 13) we have previously segmented by hand. During this tedious manual extraction, even the smallest regions have been extracted in order to make a quantitative quality measure of the detection method feasible and highly accurate. As a final result, we miss a negligible blob and, on the

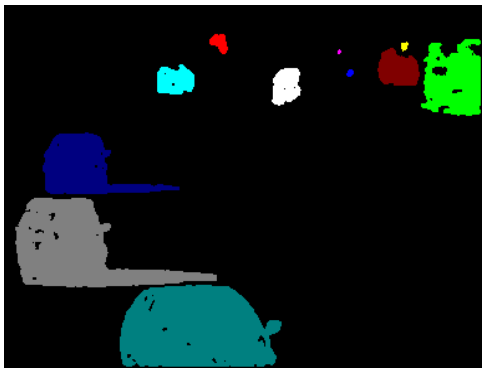

Figure 12: An output frame of the segmentation stage, where the blobs have been labeled

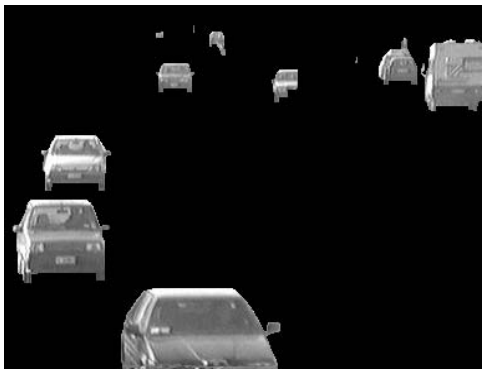

Figure 13: The "ground-truth" related to the frame studied as far

contrary, we introduce two small false blobs, which could be easily cleaned in a further area-based thresholding. It is worth remarking that any size-filtering has not been performed till now in order to better assess the effectiveness of this method.

6. Conclusions and Future Works

In this work a segmentation method utilized within the traffic monitoring application we are developing is described. The original morphological operation we devised in order to perform the FPR step shows up two properties. The first is that the operator works better with very noisy images, since these also retain most of the source signals. The second being someway related to the first; in fact, this operator is able to detect, and *preserve*, the structure of the moving blobs *while* it removes noise. The presented method combines a model-based and a thresholding approach, thus working sensibly and resulting in a human-like behavior.

Let us move to some directions for further improvements. Even though we use our operator only in the presence of binary images, it could be applied to gray level images as well. Besides, different basic SE's could be considered as well as different sizes for each element of Figure 5. A thorough inquiry should concern with the method used to choose a suitable threshold for the "morphological" fitness. At last, an efficient algorithm able to reduce the computational burden is being studied.

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 Prof. Luigi Di Stefano for having offered the sequence to study. At last, a special thank goes to Matteo Roffilli, who devised the method at its early stage.

References

[1] A. Bevilacqua and M. Roffilli. Robust Denoising and Moving Shadows Detection in Traffic Scenes. In *Proc. Technical Sketches of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai Marriot, Hawai, USA*, pages 1 – 4, December 2001.

[2] A. Bevilacqua. *A System for Detecting Motion in Outdoor Environments for a Visual Surveillance Application*. PhD thesis, Department of Electronics, Computer Science, Systems, University of Bologna, Italy, 2002.

[3] A. Bevilacqua. A Novel Background Initialization Method in Visual Surveillance. In *Proc. of the IAPR Workshop on Machine Vision Applications (MVA 2002), Nara-ken New Public Hall, Nara, Japan*, December 2002.

[4] J. Serra. *Image Analysis and Mathematical Morphology, Vol.2: Theoretical Advances*. Academic Press, London, 1988.

[5] R.M. Haralick, E. Dougherty, J. Ha, T. Kanungo, S. Karasu, C.K. Lee, L. Rystrom, V. Ramesh and I. Phillips. Statistical Morphology. In *Proc. of SPIE Conference on Image Algebra and Morphological Image Processing IV Vol.2030*, pages 191 – 202, San Diego, July 1993.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Who? When? Where? What? a Real Time System for Detecting and Tracking People. In *Proc. International Conference on Automatic Face and Gesture Recognition, Nara, Japan*, pages 222 – 227, April 1998.

[7] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in Cooperative Multi-Sensor Video Surveillance. In *Proc. of Darpa Image Understanding Workshop*, pages 3 – 24, November 1998.

[8] R. T. Collins, A. J. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring. In *Proc. American Nuclear Society on the 8^{th} International Topical Meeting on Robotics and Remote Systems, Pittsburgh, PA*, pages 1 – 15, April 1999.

[9] R. Cucchiara, M. Piccardi, A. Prati, and N. Scarabottolo. Real-Time Detection of Moving Vehicles. In *Proc 10^{th} International Conference on Image Analysis and Processing, Venice, Italy*, pages 618 – 623, September 1999.