Iterative division and correlograms for detection and tracking of moving objects

Rafik Bourezak, Guillaume-Alexandre Bilodeau

Departement of computer engineering
École Polytechnique de Montréal, P.O. Box.6079, Station Centre-ville
Montréal, QC, Canada, H3C 3A7
rafik.bourezak@polymtl.ca, guillaume-alexandre.bilodeau@polymtl.ca

Abstract. This paper presents an algorithm for the detection and tracking of moving objects based on color and texture analysis for real time processing. Our goal is to study human interaction by tracking people and objects. The object detection algorithm is based on color histograms and iteratively divided interest regions for motion detection. The tracking algorithm is based on correlograms which combines spectral and spatial information to match detected objects in consecutive frames.

Keywords. Motion detection, background subtraction, iterative subdivision, objects tracking, correlograms.

1. Introduction

Nowadays, organizations that need a surveillance system can easily get low priced surveillance cameras but they still need many security agents to keep a permanent look at all time on all the monitors. This approach is not efficient, and in fact, most of the time video tapes or files are replayed to check on a particular event after it has happened. Thus, the automation of these systems is needed as it would allow automatically monitoring all the cameras simultaneously, and advising security agents only when a suspect event is on-going. This makes video surveillance an important and challenging topic in computer vision.

A surveillance system is composed mainly of three different steps. The first step consists in detecting objects in motion. The second step is tracking, although some recent works combine these two first steps [1]. Finally, the third step is usually a high-level interpretation of the on-going events.

In this paper, we present an algorithm for each step. By opposition to previous works, for example [2] where algorithms are based on grayscale sequences and shape analysis, the developed algorithms are based on texture and color analysis to obtain more precise identification of objects.

If we briefly review existing approaches for motion detection, we note that for stationary cameras, most are based on a comparison with a reference image frame on

a pixel by pixel basis [2, 3]. Most important object detection algorithms are listed in [4]. An analysis of these methods show that they are sensitive to local variations and noise, and post-processing is often necessary to filter out erroneously labeled motion pixels. However, post-processing cannot fix detection errors involving large groups of pixels wrongly labeled on a local basis. To tackle this issue, we propose an algorithm that naturally filters out local variations by using color histograms on iteratively divided interest regions. Hence, contrarily to the usual strategy, here we first consider groups of pixels, and then gradually subdivide regions until a given precision is obtained. Motion detection is at the beginning of the process region-based, and at the end it tends to a pixel-based method. However, by starting with regions, small perturbations are ignored and hence the quality of the detection and of the reference background is improved. Furthermore, color histograms are invariant to image scaling, rotation and translation, and hence allow focusing on regions with significant motion. Finally, by controlling the number of subdivisions, our object detection algorithm can be performed at different scale to adjust to the object shape precision needed for a given application. That is, a coarse or precise shape of moving objects can be obtained.

For the tracking step, works are often based on multiple hypotheses analysis [5], other on statistics [2, 3]. We chose to use color and texture combined with some hypotheses analysis to determine new and previous objects in the scene. That is, we track by appearance.

Finally for the third step, we focus on tracking object relationships regardless of their identity. This will allow us analyzing specific behaviors of the detected objects such as the transportation of objects [2], or an illegal entry in a forbidden area [6]. Tracking generically, independently of the identity of objects will allow us to handle a larger set of objects without strong assumptions too early in the scene interpretation process.

The contributions of this paper are a moving object detection algorithm that analyses the video frames based on regions of pixels and the use of correlograms in the HSV color space for object tracking. The first contribution allows a significantly less noisy detection of moving objects, and the second allows a robust appearance tracking of moving objects.

The remainder of this paper is organized as follows. In section 2 preprocessing phase of video sequences is explained, along with the motion detection algorithm and possible post-processing. Section 3 presents the algorithms for the tracking and relationship analysis. Then, section 4 shows experimental results and their analysis. Finally, we conclude and present future works in section 5.

2. Methodology

Preprocessing is used to filter out noise and for color conversion. Then, the object detection algorithm is applied. To remove remaining shadows, it is possible to use some post-processing, though not necessary.

2.1 Preprocessing

Video capture is done in the RGB color space; however this color space is not suited for our application because small changes in the light intensity change significantly an object description. We prefer a color space less sensitive to light intensity. Thus, the HSV color space is used for all the processing because it provides direct control over brightness to normalize the light intensity V. It also focuses on the chromaticity present in Hue and Saturation. That is why H and S are given more importance in the quantization phase. Nonetheless, Hue is considered to be more reliable for color segmentation. So, the colors are quantized in 162 bins (18X3X3). Before converting from RGB to HSV space, a 3X3 median filter is applied to clean the image from acquisition noise.

2.2 Detection of objects in motion

The first step is to detect object in motion. This issue has been addressed by many algorithms [2, 3, 6]. In our application, we specifically need a time efficient algorithm that is not affected by brightness changes and noise. What we consider noise is object shadows, changes in the scene that are not of interest such as tree leaves motion, and also noise resulting from the acquisition which has not been cleaned by the median filter. To reduce the impact of noise naturally without many post-processing steps, we propose a method that is not limited to local pixel change. Instead, we consider groups of pixels to filter out noise by somewhat averaging changes over a window.

The idea is to split empirically the image into squared regions of the same size. At each step, color histograms of both reference frame H_{ref} and the current frame H_{cur} are calculated for each region. Then, the L1 distance metric is used to measure the distance between both histograms.

The L1 distance returns the level of difference, and it is defined as:

$$L1 = \sum \left| H_{ref}(i) - H_{cur}(i) \right| \quad (1)$$

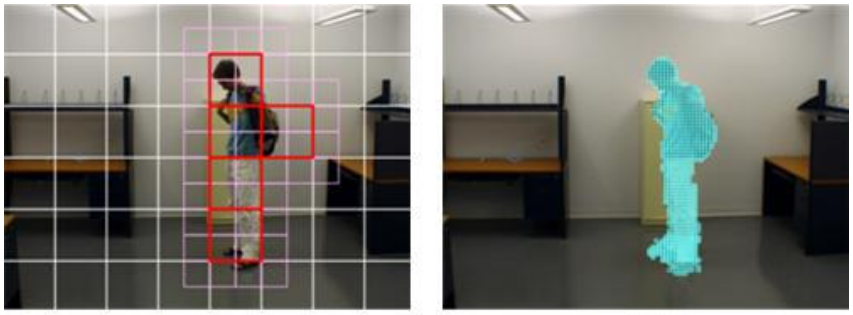

Fig. 1. A) Step one with N=64 (white square), and the first pass of step 2 and 3 (gray square). B) Final result with X_i=4 (y=4).

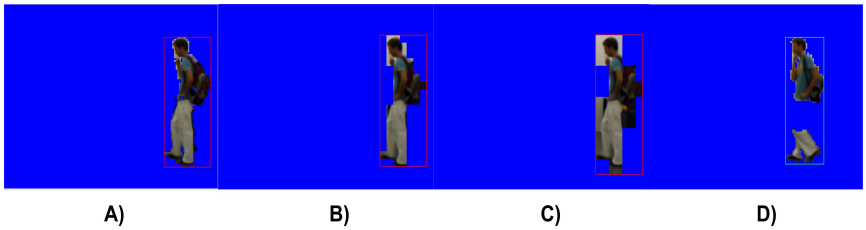

Fig. 2. Detecting and tracking moving objects. A) Object detection at fine scale, at frame 43, B) and C) object detection at coarser scales and D) fragmented object.

If L1 is larger than a specified threshold, than the regions are different, and motion is detected in that region. The threshold is fixed as a percentage of the square size, according to the level of change we want to detect. That is, the smaller is the square size, the larger will be the value of the threshold. Typically, $Th_{i+1} = Th_i + 0.10$ with $Th_0 = 0.20$. These values were set experimentally. Since histograms are normalized, changes in light intensity should not affect the threshold.

More specifically, motion detection algorithm steps are:

1. The image is first split into squares of size X_1 by X_1, the value of X depends on the size of the objects we want to track. The larger the object is relatively to the frame, the larger is X (and vice versa). Let suppose $X_1=N$. (see white squares in Fig.1a.)
2. For each region, the histograms of both the current and reference frame are evaluated using the quantization described in section 2. Then, the L1 distance is calculated between each pair of histograms. If according to L1, the square regions are similar, no motion is detected in that region. Otherwise, we consider that motion is detected and label this region as an interest region to be further processed.
3. The interest regions identified at step 2 are split in four smaller squared region, that is $X_i=X_{i-1}/2$ (see the bold gray squares in Fig. 1a, to have a more accurate segmentation of the objects in motion. Also, to preserve small extremities of objects, we split in four outside boundaries regions and include the bordering quarters to the interest regions (see bordering grey squares in Fig. 1a. Then step 2 is repeated.

The reference frame is updated with the content of the regions where no motion is detected. This way, the gradual changes in lighting is accounted for. This should allow our algorithm to perform reliably for outside scenes observed during extended time.

4. We repeat step 3 until $X_i==N/2^y$, where y is a threshold fixed according to the desired level of precision. Fig. 1b shows the final result for region of 4 by 4 pixels.

At this point, the detection of objects in motion is completed. Compared to standard background subtraction algorithms, our method does not need a statistical background model. It provides efficiency with the control of the detected object shape precision, as coarse or precise shape of objects can be obtained based on the

selected minimum region of interest size (see Fig. 2). Furthermore, the motion of small objects can be naturally filtered.

2.3. Postprocessing

Although, the detection algorithm filters out most of the noise, in some frames noise may persist because of strong shadows. This is caused by the fact that the floor has strong reflections, and these small reflection regions are in the area of squared interest regions containing motion. If such small regions are inside interest area where there is other motion for a number of a subdivision, they eventually end up at a scale where they have a significant impact of their own on the histogram of an interest region. That is why some noisy region may be included in the final segmentation. These noisy regions will be exclusively located near larger segmented region. They are removed using the algorithm proposed by Cucchiara et al. for shadow detection [9]. Note that, the HSV quantization of the frames and histogram difference threshold can be modified to avoid post processing. In our current work, we aim to do that.

3. Tracking of object in motion

The detected objects are tracked using correlograms which have been proven to be a better feature detector than histograms [7, 8]. It is a two-dimensional matrix C that combines color and texture information by quantizing the spatial distribution of color. C (i,j) indicates how many times color *i* co-occurs with color *j* according the spatial relationship given by the distance vector V(d_x,d_y), where d_x and d_y represent the displacement in rows and columns respectively.

Let *I* be the image of width *W* and height *H*, the correlogram is defined as follows:

$$C(i,j) = \left|\left\{(x,y) \in N^2, x \prec W, y \prec H \,\middle|\, I(x,y) = i \wedge I(x+dx, y+dy) = j\right\}\right| \quad (2)$$

For every detected object its histogram and correlogram in the HSV space is calculated.

First, the histogram intersection is computed to verify globally if the objects are coarsely alike. Histogram intersection HI is defined as follows:

$$HI(i,j) = \min(H_p(i,j), H_o(i,j)) \quad (3)$$

Where Hp and Ho represents the color histogram of the object we are looking for and the classified one respectively.

Then correlogram intersection is calculated to compare in more precisely both objects if necessary. Correlogram intersection CI is used generally to check if an image contains another image. Here we used it to compare objects. It is defined as follows:

$$CI(i, j) = \min(C_p(i, j), C_o(i, j)) \quad \textbf{(4)}$$

where C_p and C_o represents the correlograms of the object we are looking for (Obj_p) and the classified one (Obj_o) respectively. Once the correlegram intersection is computed, the distance L1 is calculated between C_p and CI. The closer is the distance to zero; the more likely the object is part of the other one.

The tracking algorithm works as follows:

1. The color histogram of the new object in the current frame and its correlogram are computed.
2. With each object in the previous scene, histograms intersection HI is computed. We assume that the object size does not change more than 15% from frame to frame. We believe that it is a reasonable assumption for human tracking.
3. The distance L1 between H_p and HI is computed. If L1 is larger than the threshold, then Obj_p is not the same as Obj_o. Otherwise, the correlogram intersection CI of Obj_p and Obj_o is computed. The distance L1 between CI and Cp is computed. If it is smaller than a fixed threshold the objects are similar. Otherwise, they are different.
4. Steps 1, 2 and 3 are repeated for each object.

One problem that arises often in motion detection is that the object can be split into at least two parts being considered as two distinct objects. This can be caused by the fact that the background has the same color and texture as some parts of the object. This method overcomes the problem and tells if an object of the current frame has been parts of objects in the previous frame (and vice versa). This allows us to establish if fragments are part of the same object.

3.1. Tracking of objects relationships

Currently, for interpreting a scene, our algorithm cannot identify objects. However it can tell when an object A has been taken or left by another object B. This is also done using the correlograms intersection.

3.1.1 Case when an object is left

When an object A is left, using correlograms intersection we can determine that Object A and Object B were connected as Object C in the previous frame. If object B gets far from A with a distance Z, than the algorithm determine that it was left by B. The distance Z is being used to determine whether A and B are fragments of C, or A

is left by B. Currently, Z is simply represented by the distance between the centroïds of the left Object A and the moving object B.

3.1.2 Case when an object is taken

When an object A is taken, using correlograms intersection we can determine that Object A of the previous frame has been taken by Object B to form Object C. Objects are hence merged into one object. In future work, the system should be able to keep track of the objects separately even if they are temporarily merged.

4. Experiments

The algorithms described in the previous section were implemented on a 2.6 Ghz AMD Opteron(tm) and the presented sequence has been captured in an indoor scene under multiple light sources.

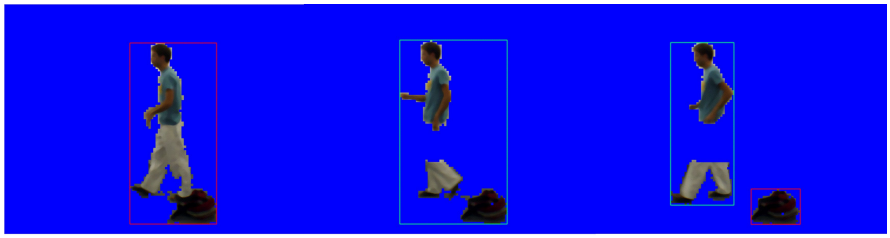

Fig. 3. Objects relationship processing A) at frame 88 B) at frame 91 C) at frame 93.

Fig. 2a shows the result of the motion detection algorithm presented in section 2. Inside the bounding box, the whole moving object is detected. Note that the region of the person includes background, partly because of the 4 by 4 pixels minimum square size. This effect is reduced using smaller square size. Larger square size gives coarser object regions. At some point in the sequence (Fig. 2d), the color and texture of the person legs and the background are similar. As many object detection algorithms this part is not detected. However, as explained in section 3, the tracking algorithm can determine that both parts belong to the same object present in previous frame because both fragment correlogram intersects with the correlogram of the whole object previously in the sequence. The tracking algorithm can also tell if a given object was split some times in the past using again correlogram intersection.

The bag is deposed in Fig. 3a; however it is still connected to the person, so the algorithm still consider them as one object. In Fig. 3b, although the bag and the person are separated; they are still considered as one object because they are close. This is cause by the choice of our distance threshold. A stronger criteria based on motion trajectory of objects is under development to obtain more robust results. Finally, in Fig. 3c the object gets far enough from the bag, so that our algorithm considers that the bag was set down by the person. Note that the body parts are still

considered to be one object, because not only the distance between object is important, but also there appearance. Further work is on the way to segment objects from persons. For the moment, we have concentrated our efforts on assuring that the fragments of a single object are considered as such even if detection fails.

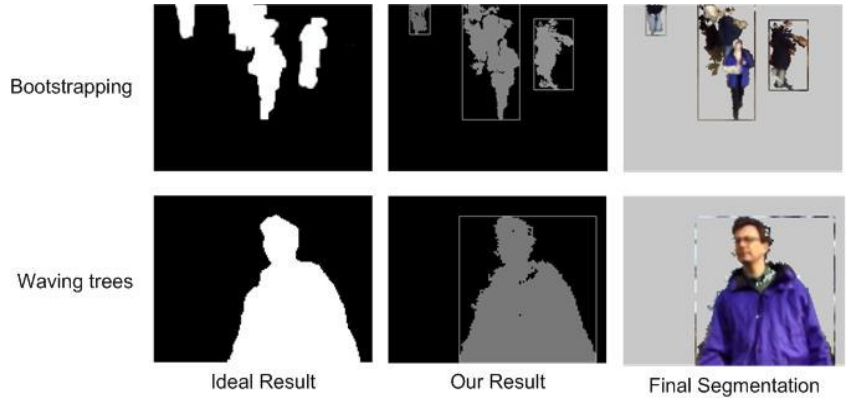

Fig. 4. Test of the segmentation algorithm on the test sequences presented in [10].

Fig. 4 presents performances of the segmentation algorithm on Bootstrapping and Waving trees sequences presented in [10]. The proposed algorithm performs relatively well compared to the methods used on the same sequences. For the Bootstrapping sequence wrong segmentation that occurred is due to the strong reflection on the floor, and the strong lightning present in the scene; however the result is close to the ground truth, and all the present people has been detected. In the waving trees sequence, the waving of the tree leafs does not affect the segmentation algorithm; however some false positives occurred on the shirt of the person because it is green as the color of the background model for the same position in the image.

Fig. 5 c and d is the frame 362 of the PETS 2001 dataset2 (camera2). In this outdoor sequence, objects to be detected are small relatively to the frame, so N is fixed to 32 and Th_0 remains equal to 0.20. The car is detected and tracked correctly since its color and texture are clear enough. When Th_0 is incremented to 0.30, in only 3 frames from the sequence the car is not detected because it is split in four parts, each part being in a different square region. This makes only small changes in these regions, thus the motion is not detected. Note that for the rest of the PETS sequence, where the there are persons walking on the road. They are too small to be detected at coarse scale and if the algorithm is applied directly at fine scale for all the images it looses its advantages since the noise has not been filtered at a coarse scale.

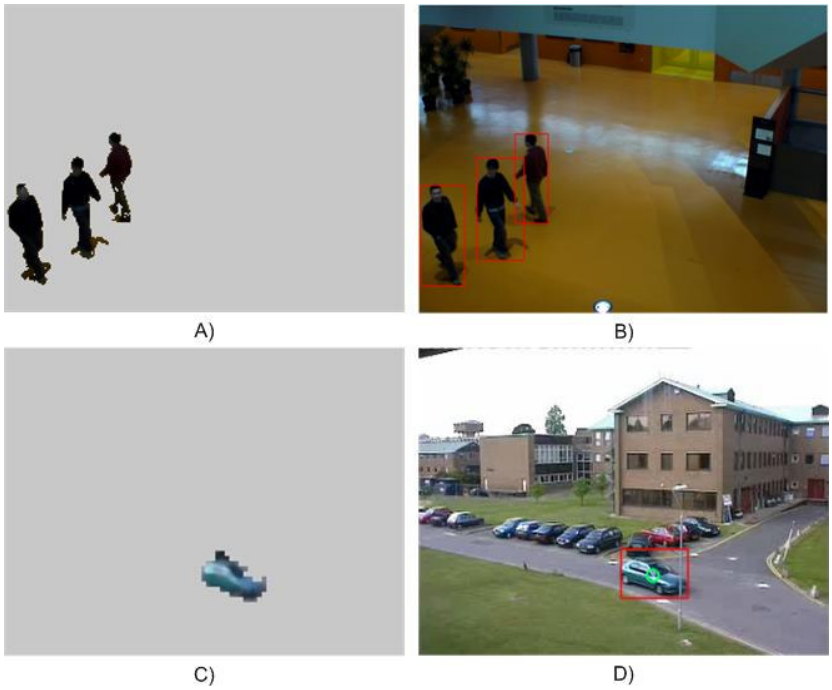

Fig. 5. A), B) Detection and tracking of multiple people walking in an atrium and C) ,D) Detection and Tracking on frame 362 from pets dataset1 2001 camera2.

Table 1 shows quantitative results for the detection algorithm which has been applied to video sequences presented at Fig. 1 (100 frames of size 512X384) and Fig. 5 (frame 257 to frame 416). The algorithm was executed for both sequences. For each frame the false negative/positive motion detected squares of size 4X4 were counted and divided on the total number of squares in the image. Then, the average value for every sequence has been recorded in the table. Also, the frequency of lost objects (false negative objects) and wrongly detected objects (false positive objects) is shown. As we can see, if the objects to be detected are too small relatively to N and Th_0 is too large, motions regions are not detected very well and even some motion objects are totally lost for some frames because they do not make a significant change in the region's histogram. Meanwhile, when Th_0 is too small false positive regions increases creating in some frames false positive objects. Thus, when N is large relatively to motion objects, good results are achieved with a smaller Th_0 and vice versa. Furthermore, the execution time relative to the number of frames shows that the algorithm is time efficient.

Table 1: Quantitative evaluation of the detection algorithm.

	N	Th_0	FP motion region (%)	FN motion region (%)	FP objects (%)	FN objects (%)	Execution time (sec)
Sequence of Fig.5 b c (160 frames)	64	0.10	0.20	0.017	0	0	13
		0.20	0.10	0.23	0	0.27	12
	32	0.20	0.18	0.017	0	0	15
		0.30	0.031	0.14	0	0.019	14
Sequence of Fig.5 a b (100 frames)	64	0.20	0.14	0.04	0	0	20
		0.30	0.07	0.15	0	0.01	19
	32	0.20	0.13	0.017	0	0	16

FP: False positive, FN: False negative

5. Conclusion

In this paper we presented an algorithm for objects motion detection, tracking and interpretation of basic relationships. This algorithm is efficient, fast and does not require a background learning phase. Furthermore, the motion detection algorithm can be performed at different scale to adjust to the object shape precision needed for one application. Also, the motion of small objects can be naturally filtered as focus is only on interest regions. Results have demonstrated that this approach is promising as it performs adequately to detect and track object regions.

In future work, the detection algorithm will be adjusted to get a better segmentation of the object borders. Furthermore, an algorithm to process square regions where no motion is detected will be implemented to predict regions where the detected objects can hide based on color and textures. It will also permit to solve occlusion problems. Finally the algorithm will be tested in an outdoor scene and the object relationship algorithm will be improved using more robust criteria.

Reference

1. Lee, K.C,Ho,J.,Yang, M.H.,Kriegman, D., Visual tracking and recognition using probabilistic appearance manifolds, Computer Vision and Image Understanding, 99 (2005), pp .303–331.
2. Haritaoglu,I., Harwood,D., Davis,L.S., W4:Real-Time Surveillance of People and Their Activities, IEEE Transaction.on Pattern Analysis and Machine Intelligence, Vol. 22, No.8, 2000, pp. 809-830.
3. Shoushtarian,B., Bez, E., A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking, Pattern Recognition Letters 26 (2005), pp. 5–26.

4. Cucchiara,R., Grana,C., Piccardi, M., Prati, A., Detecting Moving Objects, Ghosts, and Shadows in Video Streams. IEEE Transaction Pattern Analysis Machine Intelligence, Vol. 25, No. 10, 2003. pp. 1337-1342.
5. Zhou,Y.,Xu,W.Tao,H.,Gong, Y., Background Segmentation Using Spatial-Temporal Multi-Resolution MRF, in Proceedings of WACV/MOTION 2005, pp. 8-13.
6. Bodor,R., Jackson,B., Papanikolopoulos,P., Vision-Based Human Tracking and Activity Recognition, in Proceedings of the 11th Mediterranean Conference on Control and Automation, June 18-20, 2003.
7. Huang,J., Ravi Kumar,J.,Mitra,M.,Zhu,W.J., Spatial color indexing and applications, in Proceedings of 6th International Conference on Computer Vision, 4-7 Jan. 1998, pp. 602 – 607.
8. Ojala,T., Rautiainen,M., Matinmikko,E., Aittola,M.; Semantic image retrieval with HSV correlograms, in Proceedings of 12th Scandinavian Conference on Image Analysis, Bergen, Norway, 2001, pp. 621-627.
9. Cucchiara,R., Grana,C., Piccardi.M., Prati,A., Sirotti, S., Improving Shadow Suppression in Moving Object Detection with HSV Color Information, in Proceedings Of IEEE International Conference on Intelligent Transportation Systems, August 2001, pp. 334-339.
10. Toyama,K., Krumm, J., Brumitt, B., Meyers, B., Wallflower: Principles and practice of background maintenance. In: Proceedings of IEEE International Conference on Computer Vision, 1999, pp. 255-261.