A Testing Framework for Background Subtraction Algorithms Comparison in Intrusion Detection Context

Corentin Lallier[1] Emanuelle Reynaud[2] Lionel Robinault[3] Laure Tougne[1]
1. Université de Lyon, CNRS Université Lyon 2, LIRIS UMR 5205
2. Université de Lyon Université Lyon 2, EMC EA 3082
F-69676, France
3. Foxstream. 6 rue du Dauphiné, 69120 Vaulx-en-Velin.
firstname.lastname@univ-lyon2.fr

Abstract

Identifying objects from a video stream is a fundamental and critical task in many computer-vision applications. A popular approach is the background subtraction, which consists in separating foreground (moving objects) from background. Many methodologies have been developed for automatic background segmentation but this fundamental task is still challenging. We focus here on a particular application of computer vision: intrusion detection in video surveillance.

We propose in this paper a multi-level methodology for evaluating and comparing background subtraction algorithms. Three levels are studied: first, pixel level to evaluate the accuracy of the segmentation algorithm to attribute the right class to each pixel. Second, image level, measuring the rate of right decision on each frame (intrusion vs no intrusion) and finally sequence level, measuring the accordance with the time span where objects appear. Moreover, we also propose a new similarity measure, called D-Score, adapted to the context of intrusion detection.

1. Introduction

Identifying moving objects from a video sequence is a critical task in intrusion detection in video surveillance. The intrusion detection generally consists in placing a static camera into an outdoor area to record and/or to be warned of events occurring in the observed zone. The intrusion detection work is guided by three strong constraints: i) a real time processing for being warned as fast as possible, ii) without omissions: areas under surveillance could be very critical, for example airports, railways, or jails and, omissions could lead to important consequences for peoples security; iii) with a few as possible false detections.

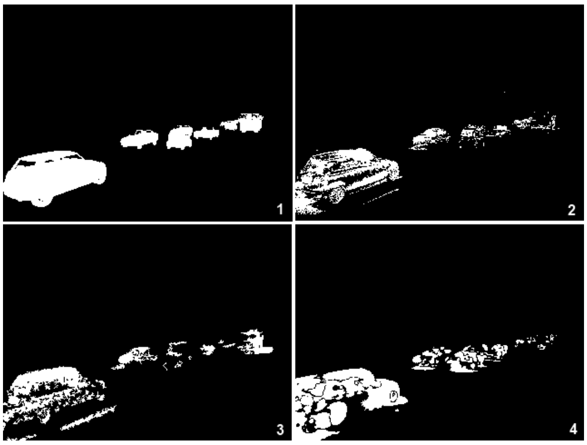

Figure 1. Background subtraction results example. (1) Ground-truth image. (2) Vumeter, D-Score: 0.113 (3) MMG, D-Score: 0.136 (4) DM, D-Score: 0.093

We can find in the literature some previous works consisting in background subtraction algorithms reviews and comparisons. Some of them are very general like [16] [3] or on the contrary, focus on specific application like traffic jam analysis [6]. They use different measures like speed, memory requirement or accuracy and compare different methods [16, 6, 8, 2]. Nevertheless, most of them only compare algorithms on a pixel classification level and none of them deal specifically with video surveillance context and its specific constraints.

The more the background subtraction is accurately performed, the more efficient the further steps of the process (such as tracking or shape recognition) will be. Examples of background subtraction results can be found on fig.1. Finding situations of robustness and weakness is a key problem for selecting the appropriate method for a given application. In this paper, we propose a general framework based on a multi level analysis to compare background extraction algorithms in intrusion detection context. The result of this work

978-1-4577-0845-9/11/$26.00 ©2011 IEEE

is a way to select the most suitable method for an intrusion detection application focusing on real-time processing and avoiding omissions.

Section 2, quickly summarizes background subtraction algorithm principle. Section 3 is dedicated to the proposed multi-level methodology. Section 4 introduces the different measures we intend to compare algorithms. In particular we propose a new measure adapted to the intrusion detection context. In section 5 is presented the built video tests database. Experimental results are given in section 6.

2. Background subtraction

Background subtraction algorithms (BSA) have to classify each pixel as a part of the foreground or the background.

The choice has been made to use background subtraction methods mainly because of the real-time processing assured by their treatment speed. Other well-known methodologies such Watershed[15], Region Growing Method[12], Partition of Graph[10], or Pulse Couple Neural Network[13] are bigger time consumers.

The main assumption for BSA is that a pixel temporal distribution of values will change significantly if it represents the background or the foreground. This could be on a very short time point of view, for example two successive images in the simplest case. In such case, if a pixel presents too much variation it is considered as a part of an object. BSA have to challenge difficulties linked to this hypothesis. In outdoor environment, they have to adapt to various levels of illumination at different daytimes, and handle adverse weather conditions such as fog or snow that modify the background. Changing shadow, cast by moving objects, should be removed so that consistent features can be extracted from the objects in subsequent processing. Moreover they have to handle the moving objects that merge with the background while staying too long at the same place.

BSA are used in several applications as traffic monitoring[17], human detection[18], or gesture recognition[5]. A great number of them exists in the literature. In this paper, we compare various BSA for detecting moving objects in outdoor video sequences, in the way to select the more robust algorithm according to the cases encountered in intrusion detection. We consider approaches varying from simple techniques such as frame differencing to more probabilistic modeling techniques. While advanced techniques often produce superior performance, our experiments show that simple techniques lead to equivalent results with much lower computational time, as we will see in the section 6.2.

3. Multi-level methodology

The evaluation of a video based intrusion detection system leads to three main questions: Was an object correctly extracted? Did the system detect it? Was it detected with a reasonable precision? Each of these questions correspond to a level of the present methodology. The extraction is validated on the pixel level. The detection is evaluated on the sequence level, and the precision of detection is measured on the image level. The rate for each level is calculated comparing the results of segmentation and detection against a ground-truth.

3.1. The pixel level.

The pixel level analysis estimates the accuracy of object extraction. First, the extraction is done by performing the classification of pixels as foreground or background, and second, the agglomeration of 8-connect foreground pixels into objects. As the pixel clustering step is dependent from the classification, the classification is the decisive step. The pixel level measures the classification accuracy comparing each pixel label, with the corresponding pixel in the ground-truth. The ground-truth is a set of binary labeled frames for each video, where each pixel is labeled as foreground or background. True and false positives (noted TP and FP) and negatives (noted TN and FN) are computed in the classical binary classification way. TP and TN are the foreground/background pixels correctly detected. FP are background pixels detected as foreground, FN are foreground one detected as background. This level is generally the only one treated in the other BSA comparisons in literature.

3.2. The image level

It points to the precise rate of object detection for each frame, by comparing the label of each frame with the correspondent ground-truth. Each image is labeled as an alarm or not, depending on the presence of an object in alarm on it. TP means this frame presents at least an object corresponding to an alarm, and it has been detected. TN means there is no alarm on this frame, with the correct decision. FP are frames with no alarm but detected as alarm. FN are alarm frames but not detected.

3.3. The sequence level

The sequence level analysis measures the global rate of detection on the span where object appears. A sequence is a set of following images having the same label. The sequence validation is done by comparing sequences of images output with ground-truths. We consider a sequence as a TP when it is detected in the time span of the ground-truth with a tolerance of one second. TN are the sequences with no alarm classified as not in alarm. FP are sequences with no alarm but detected as alarm. FN are alarm sequences but not detected in the one second span. This one second tolerance is clearly dependent of the application, our intrusion detection application.

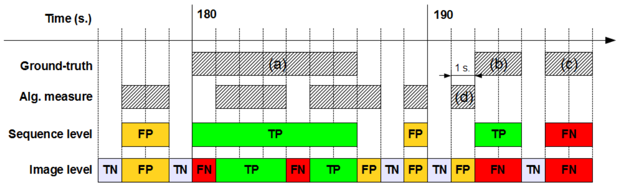

Figure 2. Comparison process for sequence and image levels. TP/FP: true/false positives. TN/FN: true/false negatives.

Fig.2 presents an example of comparison between the image level and the sequence level. On the first line is the ground-truth, on the second is the result of the detection. Third and last ones are the sequences and images levels results. The first sequence on the ground-truth (a) is considerated as TP on the sequence level in the way it matches with at least one detection from the detection result. The same sequence on the image level, is compared frame by frame, and gives a more precise result. The second sequence on the ground-truth (b) is considerated as TP on the sequence level because it matches the sequence (d) from the algorithm output in a span of one second. While on the image level, it is considerated as FP and FN. The third sequence on the ground-truth (c) is considerated as a FN because none of the output of the algorithm matches this sequence.

4. Measures

First we present the classical statistical measures for binary classification used to estimate algorithm performances. Second, a new measure, called D-Score that we developed to take more into account of localization of errors, is introduced.

4.1. Classical measures

The classical measures we choose to use are the following ones.
Precision (Prec) and Recall (Rec). On the image and sequence levels, precision is the probability of an image or sequence to be found as a genuine alarm or not. Recall is the detection rate. On the pixel level, precision and recall permit to evaluate the BSA pixel classification. Precision is the probability of a foreground pixel to be classed as such. Recall is the probability of a background pixel to be classed as a background one. Precision and recall are respectively given by: $Prec = TP/(TP + FP)$ and $Rec = TP/(TP + FN)$.
F-Measure. It is used to measuring the global performance of an algorithm. It is the harmonic mean of precision and recall and is given by: $F_1 = 2(Prec.Rec)/(Prec + Rec)$.
Average error number (Err) and standard deviation (SD). Theses two measures are only used on the pixel level, because on image and sequence level they do not bring more information than previous ones. The SD permits to estimate the general stability of the BSA in time. The smaller the SD is, the more stable through time the algorithm is.
It is worth to notice that all these measures consider errors in the same way whatever their localization is. D-Score, presented next, is a measure developed to take into account the localization and the kind of errors (FP/FN).

4.2. A measure adapted to video surveillance context

The D-Score has been developed to be more qualitative than statistical measures, in the aim of considering localization of errors in relation to real object positions. It is like Baddeley's distance[1], it is a similarity measure for binary images based on the distance transform. D-Score gives a dissimilarity criterion between the ground truth image and the segmentation result. To compute this measure we only consider mistakes in BSA results. Each error cost depends to the distance with the nearest corresponding pixel in the ground-truth. In the object extraction process, short or long range errors are less important than medium range error, because pixels on medium range impact greatly on object shapes recognition. So the penalty applied to medium range is heavier than the one applied to those in a short or large range, as shown fig.4.

Few local/far errors will produce a near zero D-Score, a good D-Score has to tend to 0. The cost is based on the *Distance Transform* of the ground-truth image, defined by:

$$DT(x) = min(d(p, x), \forall x \in X)$$

with X the set of reference pixels (the background or the foreground pixels). $d(p, x)$ the distance from the pixel p to the pixel x. The $DT(x)$ is given by the minimal distance between p and x, for each pixel x from X. To compute $DT(x)$ we use the algorithm proposed in [4].
D-Score for each pixel is given by:

$$D - Score(x) = \exp(-(\ln(2.DT(x)) - \alpha)^2)$$

α is the peak parameter. The more α is low, the more the maximum will be thin and closer to 0. We punish error with a tolerance of 3 pixels from the ground-truth, because these local errors do not really affect the recognition process. For the same reason, we allow the errors that occur at more than a 10 pixels distance. That is why, in the next experiments, we choose $\alpha = 5/2$. $D - Score(x)$ is plotted on fig.3

A first DT map is computed for the foreground of the ground-truth image, noted $EMap_{for}$, and a second for the background, noted $EMap_{back}$. This permits to have different costs for false negatives and false positives. As we want to avoid false negatives, their cost should be higher than the false positive ones. This is done by scaling the $EMap_{for}$ by a scalar s. In experiments we use $s = 5$, in other words, false negatives cost 5 times more than false positives. Next, by summing $EMap_{for}$ and $EMap_{back}$ into a general map

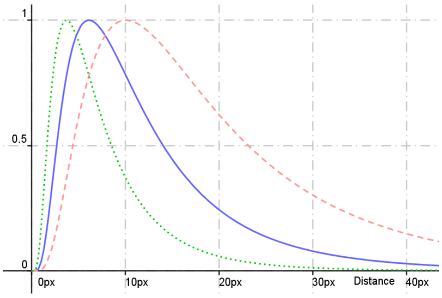

Figure 3. D-Score value based on the distance from ground-truth. Dashed red : $\alpha = 3$. Dotted green : $\alpha = 2$. Solid blue : $\alpha = 5/2$ (actual value used in the next experiments). Errors in [3, 10] are the most punished.

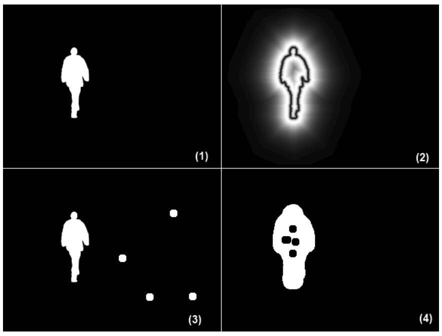

Figure 4. D-Score. (1) Ground-truth. (2) $EMap$ (cost map). (3) Example of long ranges errors. D-Score: 0.003 (4) Omissions with medium range errors. D-Score: 0.058

called $EMap$, we compute the cost of each pixel on a single map, according to distance and error kind (see fig.4 (2)).

The D-Score on a given frame is the mean of errors on every pixels. An example can be found on fig.1. On a given video, the total D-Score is given by the mean of all D-Score on all frames. The more efficient the BSA is, the lower the associate D-Score will be. This new measure is better adapted to the intrusion detection context, by taking globally the position and the shape of objects into account, but also taking care of the type of error made by the BSA. The next section presents the video database built for the evaluation.

5. Videos test database

The test base has been built in order to point out the strengths and the weaknesses of the object detection process in outdoor. Different sets of labeled videos are used to measure *a posteriori* capacities of the video surveillance system. They can be classed according to their durations and their natures: **real long** videos permit to check analysis robustness in time and avoid long time false detection, they could present long time change in luminosity, **real short** videos present a high density of objects in time. They permit to quickly test a range of situations, and avoid object omissions. **Artificial** videos, provided by [7, 11], are short, generally less complex in terms of numbers of events, but are labeled frames by frames and permit to test specifically pixel classification. These video bases allow to test the influence of some difficulties encountered during the object extraction phase. Those difficulties have been sorted according to 1) the ground type, 2) the presence of vegetation 3) casted shadows, 4) the presence of an attendant car flow near to the surveillance zone, 5) the general climatic conditions, 6) fast light changes in the scene and 7) the presence of big objects. All of them could perturb BSA. Each of these videos have been labeled. Next section presents a sketch of this methodology on several BSA.

6. Experimental results

In this section we present the background subtraction methods tested, next experimental results are shown, and finally several points are discussed.

6.1. Implementation and setup

We have chosen to test Gaussian Mixture Model, Modified Mixture of Gaussian, the Vumeter, Simple Image Difference and Weighted Image Difference. Implementation is done using *c++* and *OpenCV*, all these tests have been run on a Intel Core2 Quad Q9550 @ 2.83 GHz. Videos are analyzed in 320*240 resolution.

Gaussian Mixture Model: (**MMG**) from [14], is based on a parametric probabilistic background model. Parameters used are: threshold of mixtures sum: 0.7, variance multiplier: 2.5, historic size: 200, distributions number: 5, initial weights: 0.05, initial variance: 30, min. area: 15. MMG have a processing time (PT) of 19-20 ms.

Modified Mixture: (**MMG***) is based on the previous one, the main difference is that the background is modelized by a unique distribution. Setup is: ED factor: 5, threshold: 5, threshold for background reset: 0.6. PT $\approx$ 8-9 ms.

Vumeter: (**VM**) from [9], is a non-parametric model, based on a discrete estimation of the probability distribution. Parameters used are: bins by components: 16, learning rate: 0.01, decision threshold: 0.1. PT $\approx$ 11-12 ms.

Simple Image Difference: (**Diff**). The state of the pixel is based on the intensity difference between the previous and the current frame. The value used for the difference threshold is 25. PT $\leq$ 1 ms

Weighted Image Difference: (**DM**). In the same way, the state of the pixel depends on the difference between the time-pondered mean of the last values of the pixel, and the present value. The value used are: threshold: 10, weight of the current image: 50. PT$\leq$ 1 ms

6.2. Results

These results were obtained by computing all measures on all videos bases.

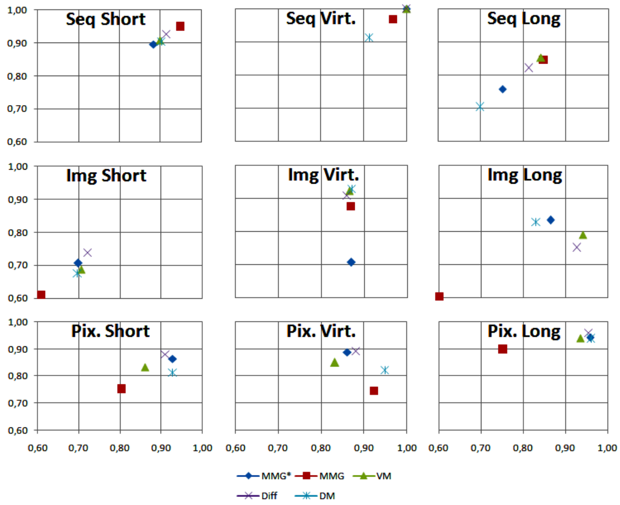

Figure 5. Precision-recall [0.6:1.0] plots for the sequence, image, and pixel levels on the three video bases.

6.2.1 Video bases

The fig.5 shows precision and recall for short, virtual and long videos. The long ones have better scores on pixel level. In the opposite of shorts and virtuals, nothing can append during a long time, facilitating the extraction process. There are also more sequences with objects to be detected in long videos than in others, this explains the difference shown on the sequence level. There is few difference between short real videos and virtual ones. The main difference is on the image level, short videos presents many variabilities, particularly in the illumination conditions whereas virtual are more stable. This have an impact on the precision and recall values. Long videos are suitable to test difficult situations in sequence detection, short videos in image detection and virtual ones for pixel classification process.

6.2.2 Levels

The tables 1, 2 and 3 show respectively the pixel, image and sequence level. MMG make many errors on pixel level, with the highest error rate, highest SD, and lowest F_1. But on the sequence level, it is among the best with the VM, with $F_1 = 0.92$. According to F_1, on the pixel level , BSA order is Diff, MMG*, VM, DM, MMG, on image level: VM-Diff, DM-MMG*, MMG, and on sequence level: MMG-VM, Diff, MMG*, DM. Each level of analysis gives different sorting. This shows levels results are not correlated each others. The sequence level gives information about the effective detection of objects. The image level give information about stability of detection in time. If an algorithm tends to be unstable, it could have a good sequence level but a poor image level.

	MMG	MMG*	VM	Diff	DM
Prec	0.83	0.92	0.88	0.92	0.95
Rec	0.80	0.90	0.87	0.91	0.86
F_1	0.80	0.90	0.87	0.91	0.82
D-Sc	0.019	0.026	0.025	0.024	0.018
Err.(avg.%)	30.96	2.04	2.08	2.09	3.65
SD.(avg.%)	16.5	1.2	2.9	3.0	4.3

Table 1. Results on pixel level. Averaged on all video bases.

	MMG	MMG*	VM	Diff	DM
Prec	0.71	0.81	0.84	0.84	0.80
Rec	0.63	0.80	0.80	0.80	0.81
F_1	0.66	0.80	0.81	0.81	0.80

Table 2. Results on image level. Averaged on all video bases.

	MMG	MMG*	VM	Diff	DM
Prec	0.92	0.88	0.91	0.91	0.84
Rec	0.92	0.88	0.92	0.92	0.84
F_1	0.92	0.88	0.92	0.91	0.84

Table 3. Results on sequence level. Averaged on all video bases.

6.2.3 D-Score

We can notice in table 1, that the F_1 measure, the precision-recall, the SD and the average error rate give a similar BSA order on pixel level. D-Score gives us a different information, the fig.6 compare more precisely D-Score and F_1 on the different bases. For exemple, for MMG, the F_1 is the worst of all the BSA, but it has generally a very good D-Score. This indicates MMG make many errors, but these errors have a low cost and they do not pertub shape recognition. MMG make few omissions and generally make short and long range errors. According to D-Score, BSA order on pixel level is DM, MMG, Diff, VM, MMG*.

To conclude on these results, VM is very good on image and sequence levels, the simple Diff is near the best on every conditions. MMG* are fast, but have a lots of omissions. The two image based difference algorithms, Diff and DM often show interesting performances, but can't be used as is in exploitation: if an object stop moving, it cannot be extracted. In the context of intrusion detection, our results tend to demonstrate that MMG seems to be the best according to the previously seen constrains, in opposition to the classical BSA review literature.

7. Discussion and conclusion

The questions we try to answers in this article are the following ones. First, are measures of literature enough to compare BSA in the specific frame of intrusion detection ? Our answer is no. They do not take into account the error lo-

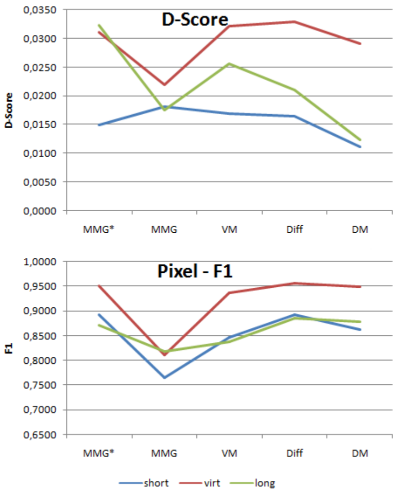

Figure 6. F_1 (down) and D-Score (top) on pixel level for long, short and virtual videos.

calizations. This is the reason why we propose the D-Score. Second, is it sufficient to compare BSA on the pixel level ? Our answer is, once again, negative. Pixel level gives informations about object extractions, image and sequence levels report informations about objects detection, with a specific focus on the stability in time for the image level. These three levels are complementary. Finally, how can we build tests database ? We have to mix virtual videos that allow to test specific conditions and that are pixel labeled on each frame. They are suitable specifically for pixel classification task. Real videos are used to test real light conditions that could be simplified in virtual videos. Long videos are oriented for sequence detections and help to validate robustness in time whereas short videos stand for classification stability.

In this work we propose a methodology to compare background subtraction algorithms in intrusion detection context. For this comparison we presented a dedicated framework based on different levels of analysis, and a specific video base. Notice that this methodology could be extended to other BSA comparisons and more generally to other segmentation algorithms comparison.

References

[1] A. Baddeley. An error metric for binary images. *Robust Computer Vision*, (march 1992):9–11, 1992.

[2] A. Bayona, J. SanMiguel, and J. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. *Advanced Video and Signal Based Surveillance*, pages 25–30, Sept. 2009.

[3] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, Dec. 2008.

[4] G. Borgefors. Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344—-371, 1986.

[5] F. Chen. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, 21(8):745–758, Aug. 2003.

[6] S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. *Proceedings of SPIE*, 5308:881–892, 2004.

[7] Y. Dhome, N. Tronson, A. Vacavant, T. Chateau, C. Gabard, Y. Goyat, and D. Gruyer. A Benchmark for Background Subtraction Algorithms in Monocular Vision: a Comparative Study. *International Conference on Image Processing Theory, Tools and Applications*, 2(2), 2010.

[8] H. Gao and R. Green. A Quantitative Comparison Research on Frame Level Background Subtraction Algorithms. *digital.liby.waikato.ac.nz*, (December):31–34, 2007.

[9] Y. Goyat, T. Chateau, and L. Malaterre. Vehicle trajectories evaluation by static video sensors. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 864—-869, 2006.

[10] D. Greig, B. Porteous, and A. Seheult. Exact maximum A Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society. Serie B*, 51(2):271–279, 1989.

[11] D. Gruyer, C. Royere, N. Du Lac, G. Michel, and J.-M. Blosseville. SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems. *World Congress and Exhibition on Intelligent Transport Systems and Services*, 2006.

[12] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.

[13] J. Johnson and M. Padgett. PCNN models and applications. *Neural Networks, IEEE Transactions on*, (10):410–198, 1999.

[14] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. European Workshop on Advanced Video-Based Surveillance Systems*, pages 1–5, 2001.

[15] F. Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303—-306, 2002.

[16] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics (International Conference on)*, volume 4, pages 3099–3104. Ieee, 2004.

[17] I. Pop, M. Scuturici, and S. Miguet. Common motion map based on codebooks. In *Advances in Visual Computing*, pages 1181—-1190, 2009.

[18] N. Thome, D. Merad, and S. Miguet. Learning articulated appearance models for tracking humans: A spectral graph matching approach. *Signal Processing: Image Communication*, 23(10):769–787, Nov. 2008.