ADAPTIVE BACKGROUND SUBTRACTION IN H.264/AVC BITSTREAMS BASED ON MACROBLOCK SIZES

Antoine Vacavant
Clermont Université, Université d'Auvergne, ISIT, F-63000, Clermont-Ferrand, France

Lionel Robinault, Serge Miguet
Université de Lyon, CNRS, Université Lyon 2, LIRIS UMR5205, F-69676, Lyon, France

Chris Poppe, Rik van de Walle
Ghent University-IBBT, Multimedia lab, B-9050, Ghent, Belgium

Keywords: H.264 Bitstream analysis, Background subtraction, Adaptive background modeling.

Abstract: In this article, we propose a novel approach to detect moving objects in H.264 compressed bitstreams. More precisely, we describe a multi-modal background subtraction technique that uses the size of macroblocks in order to label them as belonging to the background of the observed scene or not. Here, we integrate an adaptive Gaussian mixture-based scheme to model the background. We evaluate our contribution using the PETS video dataset and a realist synthetic video sequence rendered by a 3-D urban environment simulator. We compare two different background models, and we show that the Gaussian mixture-based is the best and outperforms other techniques that use macro bloc sizes.

1 INTRODUCTION

H.264/AVC (Wiegand et al., 2003) is the newest video coding standard developed by the Joint Video Team (JVT), and overtakes its predecessors (*e.g.* MPEG-4) by higher compression performances and various profiles. Moreover, H.264/AVC permits to deliver high quality videos for local or network interfaces. Nowadays, the analysis of compressed H.264 bitstreams for object detection and tracking is an important challenge for many video surveillance applications. Indeed, they generally need to process large volumes of video streams while transferring or storing them.

Many approaches for moving object detection analyze the motion vectors (MV) encoded in the macroblocs (MB) of a bitstream (De Bruyne et al., 2009; Mehmood et al., 2009; Liu et al., 2007; Solana-Cipres et al., 2009; You et al., 2007). However, MV are useful to make the compression rate optimal, but they do not reliably represent the motion of the moving objects. Moreover, these vectors are very noisy and a post-processing filter is necessary to improve the movement estimation. Poppe *et al.* (Poppe et al., 2009) propose to consider the size of a MB in order to decide if it belongs to the background or to a moving object. They show that their approach is more precise than MV based approaches (by considering recall and precision), and this is a fast algorithm, convenient for real-time applications. In this method, the background model is carried out by computing the maximal size of MB during the learning phase, where it is supposed that no object appears in the filmed scene.

Here, we propose to integrate and to compare the Gaussian mixture model (GMM) and the VuMeter (VUM) in order to build adaptive background models based on MB sizes. We show that the GMM is the best approach to represent the distribution of MB sizes. The first method is a statistical approach that was first introduced by (Stauffer and Grimson, 1999) in the classical image domain to represent several distributions of pixel values. The second model was established by (Goyat et al., 2006) and is based on a discrete representation of the distribution of pixel colors. There exists numerous adaptive background models in the literature. Many authors search to improve the GMM, *e.g.* (Chen et al., 2007; Kaew-

Vacavant A., Robinault L., Miguet S., Poppe C. and van de Walle R..
ADAPTIVE BACKGROUND SUBTRACTION IN H.264/AVC BITSTREAMS BASED ON MACROBLOCK SIZES.
DOI: 10.5220/0003364000510058
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2011), pages 51-58
ISBN: 978-989-8425-47-8

trakulpong and Bowden, 2001; Poppe et al., 2007; Tuzel et al., 2005; XiaHou and Gong, 2008; Zivkovic, 2004), while other introduced new methodologies, like codebooks for example (Kim et al., 2005; Sigari and Fathy, 2008). In this article, we have chosen a very classical model (GMM) and an other approach not based on Gaussian mixtures, VUM, which has been recently demonstrated as an efficient algorithm in (Dhome et al., 2010). In our system, we are thus able to model the size of MB thanks to complex distributions. Moreover, we can handle all frame modes (*i.e.* I, B and P frames) with our method.

This article is organized as follows. In Section 2, we present the two adaptive models we have developed for background subtraction in H.264/AVC bitstreams. Then, we compare these methods in terms of quality of segmentation with related work by using various (synthetic and "real") video data sets.

2 PROPOSED METHODS FOR BACKGROUND SUBTRACTION IN H.264/AVC BITSTREAMS

2.1 Related Work and Motivation

Our method uses the size of a bitstream MB (in bits) to classify it into background or foreground. The main idea is thus that a MB representing a moving object should be more voluminous than if it contains only background. A first idea could be to use a simple threshold to determine if a MB is foreground or not, with its size Z_t at time t:

$$Z_t > T_{STH} \quad (1)$$

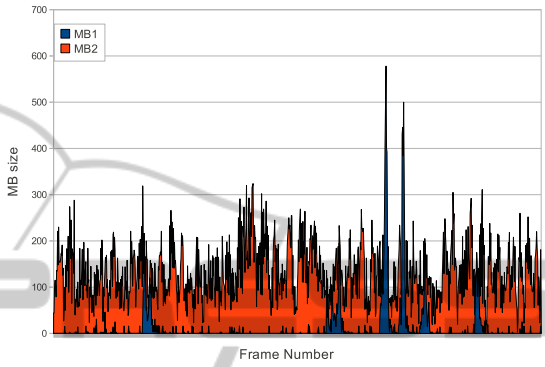
In (Poppe et al., 2009), the authors build the background model Z_{MAX} by computing the maximal value of the size of a given MB during the learning phase. During the detection phase, a MB is first considered as foreground if its size Z_t respects:

$$Z_t > Z_{MAX} + T_{MAX} \quad (2)$$

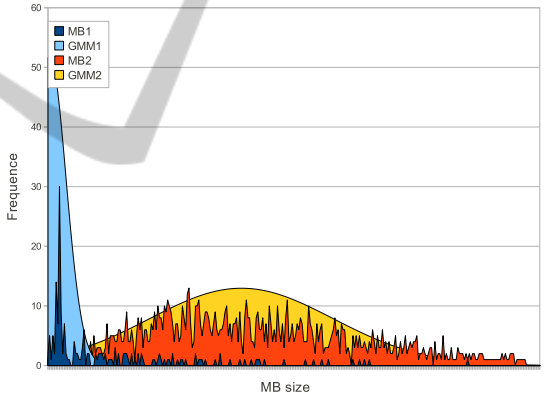
where T_{MAX} is a given threshold for a given MB. This model has several limitations.

It supposes that the background of the observed scene is stationary; *i.e.* no update operation is carried out during the detection phase. Even if this choice may be convenient for several video surveillance applications, an adaptive background model is generally a better solution. Indeed, the model of the MB size should evolve during (maybe very long) stream processing time. We show in Figure 1 the evolution of MB sizes in a real video, where we have chosen two

(a)



(b)



(c)

Figure 1: We consider a real video, where we have chosen two MB (a). Figure (b) represents the size of the chosen MB during acquisition time, and (c) the distribution of MB sizes.

particular MB. In (a), we have depicted them, the first MB (MB1) represents an uniform background, where an object passes through during the video. MB2 is placed on a complex background, where no moving object passes through. In Figure 1-(b), we show the evolution of both MB during acquisition time. We can notice that the MB size may be significantly different if it represents complex environment or not. Moreover, we can see that MB2 size may be greater than the one of MB1 when it contains a moving object. We have measured that new objects in complex MB does not imply necessary a greater augmentation than

in uniform MB. Hence, a single threshold for all MB is not a convenient solution for every H.264/AVC MB configurations. The distribution of sizes of each MB (Figure 1-(c)) may be modeled thanks to one normal law. In this figure, we have shown this result with the MB1 and MB2 sizes distributions (respectively blue and yellow plots). One more time, we show that Equation 2 does not lead to a relevant representation of background MB in video sequences.

Furthermore, only MB size from P frames are really represented in the model chosen in (Poppe et al., 2009). I frames are treated separately, since the MB sizes are sensibly larger than in P frames, and the case of B frames is not discussed. A more complex background model should represent the various frame modes available in H.264/AVC standard. The purpose is to propose a generic approach that could be adapted for both real-time applications using smart cameras and off-line video processing systems.

As a consequence, in our method, we represent the distribution of sizes of a given MB thanks to two adaptive background models. We finally show that the Gaussian-based model is the more efficient, since its representation fits the best to the MB sizes distribution.

2.2 Adaptive Background Models based on H.264/AVC MB Sizes

In the GMM, at time t, we consider that the model M_t generated for each MB from the measures $\{Z_0, Z_1, \ldots, Z_{t-1}\}$ is correct. The likelihood that a MB is a background MB is (Stauffer and Grimson, 1999; Hayman and Eklundh, 2003):

$$P(Z_t \mid M_t) = \sum_{i=1}^{i=N} \alpha_i \, \mathcal{N}(\mu_i, \sigma_i) \tag{3}$$

$$\mathcal{N}(\mu_i, \sigma_i) = \frac{1}{(2\pi)^{1/2}|\sigma_i|^{1/2}} e^{-\frac{1}{2}(Z_t-\mu_i)^T \sigma_i^{-1}(Z_t-\mu_i)} \tag{4}$$

Here, once we have extracted the size Z_t (in bits) of a macroblock MB_t (or simply MB) at time t in the bitstream, we test if Z_t belongs to the GMM by considering each Gaussian distribution G_i (with normal law $\mathcal{N}(\mu_i, \sigma_i)$) (Cheung and Kamath, 2004):

$$Z_t - \mu_i \leq k \, \sigma_i \tag{5}$$

implies that Z_t belongs to the Gaussian distribution G_i. In this equation, k is a constant that is generally between 1 and 4. If Z_t does not belong to any Gaussian in the GMM, then the associated MB is considered as foreground.

For the VUM model, the MB sizes are modeled with a discrete distribution with N bins. A MB can take two states, (ω_1) if the MB is background, (ω_2) if the MB is foreground. This method estimates $P(\omega_1 \mid Z_t)$. The probability density function can be approximated by (Goyat et al., 2006; Dhome et al., 2010):

$$P(\omega_1 \mid Z_t) \approx K_i \sum_{j=1}^{N} \pi_t^j \delta(b_t - j) \tag{6}$$

where δ is the Kronecker delta function, b_t gives the bin index vector associated to Z_t, j is a bin index, and K_i is a normalization constant to keep at each moment $\sum_{j=1}^{N} \pi_t^j = 1$. π_t^j is a discrete mass function which is represented by a bin. After a lot of images, the bins which are modeling the background have a high value. To choose at each moment if a MB is background or not, a threshold T_{VUM} is set. Each new MB with corresponding bins under T_{VUM} will be detected as background, foreground otherwise.

2.3 Global System for Background Subtraction Adapted to H.264/AVC Bitstreams

If the adaptive background model chosen (GMM or VUM) classifies a MB as foreground (Figure 2, test #1), then we check if MB is a skipped MB or not (Figure 2, test #2). This special kind of MB, defined in the H.264/AVC standard (Wiegand et al., 2003), represents a MB where no residual data has been computed during the encoding process. In this case, we have to consider the set of MB according to the $\mathcal{V}_{\text{MB skipped}}$ neighbourhood (see Figure 2, test #3, and also Figure 3). Indeed, these MB are used during the decoding process to reconstruct a valid skipped MB. In this test, we suppose that if surrounding MB are foreground, then the skipped MB is too, otherwise this is a background MB. In this latter case, we apply a spatial filter (Figure 2, test #4); that is, we assume that if at least four MB are foreground within the set of neighbor 8 connected MB, then the current treated MB is assigned as foreground (background otherwise).

In the case where Equation 5 fails or when previous tests #3, #4 return true, MB is finally assigned foreground if the temporal filter (Figure 2, test #5) is verified. This filter allows to keep the MB at time t as foreground if MB_{t-1} or MB_{t+1} are foreground. If this last test fails, then MB is considered as background, and it is possible to update the adaptive model chosen. The waved line represented in Figure 2 means that this operation may be carried out with a lower frame rate, to decrease computational cost and possibly decrease over learning effects (Cheung and Kamath, 2004). If the background model is based on a valid learning phase (with correct conditions of acquisitions, the scene does not contain moving objects

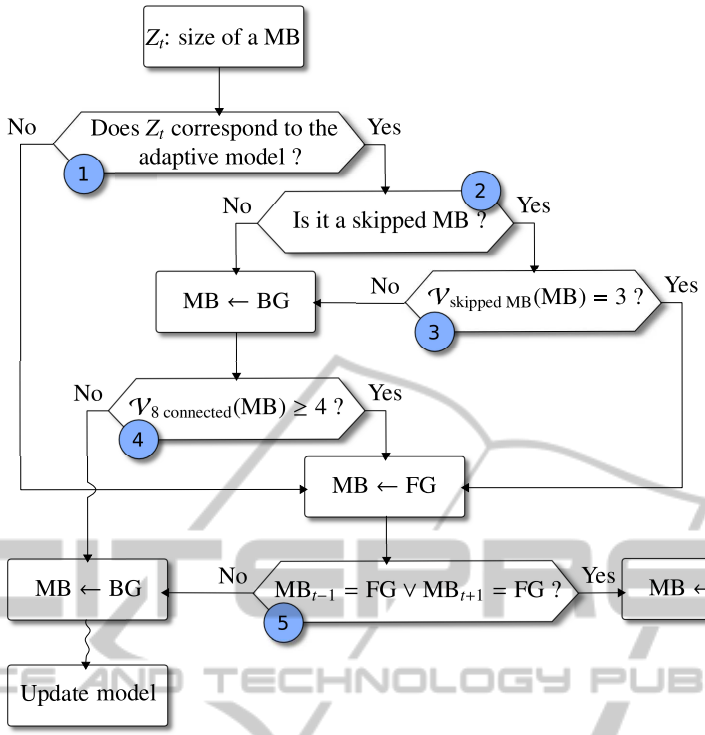


Figure 2: Global flow chart of our method.

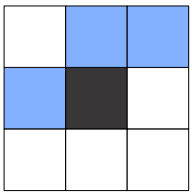


Figure 3: From a given MB (central dark MB), the $\mathcal{V}_{\text{MB skipped}}$ neighbourhood only contains the three foreground MB. The central MB is then assigned to foreground.

during long time, *etc.*), we can also skip this final update procedure during the detection phase.

In the next section, we propose to compare our contribution with related work thanks to various video datasets.

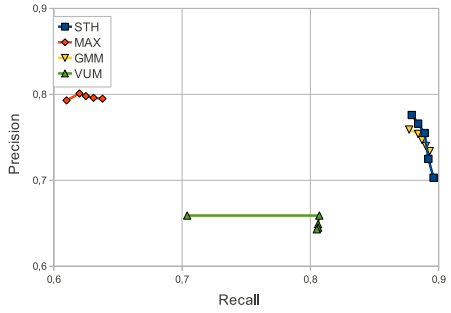
3 EXPERIMENTAL ANALYSIS

We first have to recall that the use of the MB sizes was shown to overcome many MV based algorithms in (Poppe et al., 2009). Hence, we propose to compare our adaptive approaches with related works using MB sizes. If we show that our contribution based on GMM is better than previous work, then by extension, it should overcome these MV based techniques.

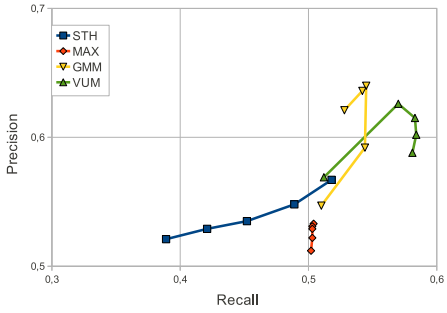
In our experiments, we consider PetsD2TeC2 and Indoor sequences, that were introduced in the PETS 2005 workshop (Brown et al., 2005), and a real video obtained from our visual surveillance activities (FStream). For these videos, we have manually labeled the desired ground truth every 50 frames, which allows us to compare the quality of our background subtraction algorithms, with respect to the one based on Equation 2. We denote our two adaptive algorithms by GMM and VUM respectively and by MAX the previous work based on the maximal size. The simple threshold method we have illustrated through Equation 1 is denoted by STH in the following. Moreover, a complete realist 3D urban environment renderer has been developed using the LIVIC SiVIC simulator (Gruyer et al., 2006) to generate an other synthetic video. Thanks to this software, we are also able to compute exactly the associated ground truth of this sequence (see Figure 4 for an illustration of each video in a gray level color mode).

Since we propose a MB-based output, we consider a ground truth defined on the MB domain. If we consider pixel-based ground truth, our benchmark may be built by assigning a MB as foreground if at least one pixel is foreground inside this MB. Once we have computed the true/false positives TP_i, FP_i and the true/false negatives TN_i, FN_i of each frame i, we can consider the recall and the precision of each class:

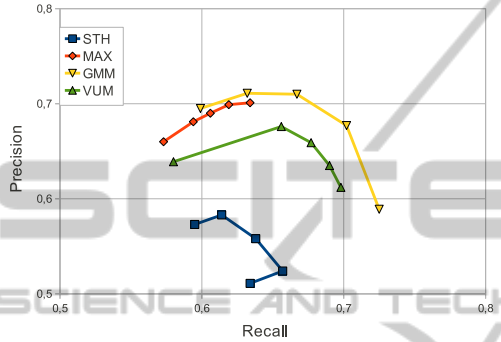
$$Re_i(P) = TP_i/(TP_i + FN_i) \quad (7)$$



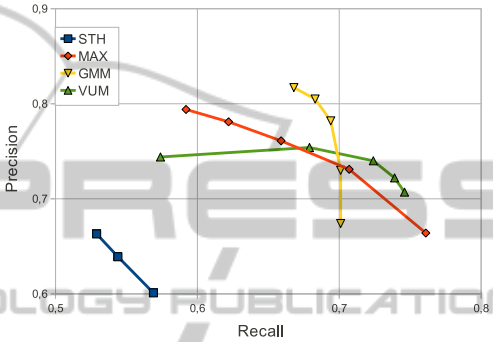
(a) Synth12



(b) Indoor



(c) PetsD2TeC2



(d) FStream

Figure 5: Recall/precision curves for STH, MAX, GMM and VUM algorithms and for each video sequence: (a) Synth12, (b) Indoor, (c) PetsD2TeC2, and (d) FStream. Please note that the X and Y scales are different from one plot to an other.

(a)

(b)

(c)

(d)

Figure 4: One frame from each video sequence we have chosen for our experiments: (a) Synth12, (b) Indoor, (c) PetsD2TeC2, (d) FStream.

$$Re_i(N) = TN_i/(TN_i+FP_i) \quad (8)$$
$$Pr_i(P) = TP_i/(TP_i+FP_i) \quad (9)$$
$$Pr_i(N) = TN_i/(TN_i+FN_i) \quad (10)$$

This leads to the classical recall and precision values that are the means of the following Pr_i and Re_i values:

$$Pr_i = (Pr_i(P)+Pr_i(N))/2 \quad (11)$$
$$Re_i = (Re_i(P)+Re_i(N))/2 \quad (12)$$

In our experiments, we also consider the classical F-measure that was recently used in an other work about the comparison of background subtraction algorithms (Dhome et al., 2010):

$$F = \frac{1}{n}\sum_{i=1}^{n} 2\times\frac{Pr_i\times Re_i}{Pr_i+Re_i} \quad (13)$$

for a video sequence of length n.

Moreover, we compute the PSNR (Peak Signal Noise Ratio), which is defined as:

$$PSNR = 10\log_{10}\left(\frac{mn}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}||I(i,j)-I_r(i,j)||^2}\right) \quad (14)$$

where I and I_r are respectively the tested image and the ground truth image, of size mn.

Figure 5 presents the recall/precision curves associated to each algorithm, and for each video sequence (Synth12, Indoor, PetsD2TeC2, FStream). In the STH method, we have chosen the threshold $T_{STH} = \{50,100,150,200,250\}$. We have also

Table 1: The F measure observed for STH, MAX, GMM and VUM algorithms (a), the mean of the three best F measures (b), and the PSNR (c). For a given measure, we indicate within parenthesis the ranking of each approach, we then depict the global ranking of the algorithms for each measure, and the one for all the measures (d).

(a) Best F

	Synth12	Indoor	PetsD2TeC2	FStream	Global ranking
STH	0.820 (1)	0.521 (3)	0.586 (4)	0.606 (4)	3
MAX	0.701 (4)	0.511 (4)	0.657 (2-ex)	0.715 (3)	4
GMM	0.808 (2)	0.581 (2)	0.676 (1)	0.734 (1)	1
VUM	0.722 (3)	0.594 (1)	0.657 (2-ex)	0.727 (2)	2

(b) 3 best F

	Synth12	Indoor	PetsD2TeC2	FStream	Global ranking
STH	0.797 (2)	0.485 (4)	0.643 (4)	0.594 (3)	3
MAX	0.697 (4)	0.510 (3)	0.648 (3)	0.707 (3)	4
GMM	0.807 (1)	0.574 (2)	0.671 (1)	0.732 (1)	1
VUM	0.713 (3)	0.591 (1)	0.654 (2)	0.725 (2)	2

(c) PSNR

	Synth12	Indoor	PetsD2TeC2	FStream	Global ranking
STH	24.230 (2)	5.753 (4)	19.135 (4)	21.400 (3)	3
MAX	25.034 (1)	23.757 (1)	24.875 (1)	21.657 (2)	1
GMM	23.427 (3)	22.766 (2)	24.749 (2)	22.701 (1)	2
VUM	21.934 (4)	20.557 (3)	23.277 (3)	21.162 (4)	4

(d) Final ranking

	F measure	3 best F	PSNR	Final ranking
STH	3	3	3	3-ex
MAX	4	4	1	3-ex
GMM	1	1	2	1
VUM	2	2	4	2

considered the MAX algorithm with the threshold $T_{MAX} = \{0, 20, 40, 60, 80\}$ for the P frames in Equation 2, and we have imposed another threshold for B frames by supposing that they are 2.5 larger than P frames. For our contribution GMM, we have chosen $k = \{2.0, 2.5, 3, 3.5, 4.0\}$ in Equation 5 and one Gaussian for each mode (I, B, P). VUM is parametrized with $T_{VUM} = \{0.01, 0.05, 0.10, 0.15, 0.20\}$.

We can notice that GMM has better performance, and a more stable behavior than STH, MAX and VUM for the four videos. This fact is also numerically observable if we consider the best F measure and the mean of the three best F measure for each algorithm (see Table 1). A particular case can be observed on the Synth12 video, where STH is better than GMM with F measure, but not with the mean of the three best F. If we now consider the best PSNR of each method, we can notice that the VUM and STH algorithm are always the worst ones. Moreover, the use of the thresholds T_{STH} in STH is a worse solution than using high thresholds as in MAX for real videos Indoor, PetsD2TeC2 and FStream. GMM is the best for FStream video, GMM and MAX are very close for PetsD2TeC2 video, and MAX is the best for Synth12. The differences between F and PSNR results may be explained by the fact that the PSNR is very sensitive to noise, which means that false positive detections of VUM algorithms and false negative detections of STH highly alter this measure for example. We have also illustrated the results of STH, MAX, GMM and VUM algorithms in Figure 6 with output images labeled with different colors (see figure for explanations). One can notice that GMM is the best algorithm on Synth12, Indoor and FStream videos, since its result contains the greatest number of true positive MB. The main problem of VUM model is that it may produce too many false positive detections, whereas MAX algorithm can lead to a high number of false negative MB (see FStream video). Visual experiments show that the GMM approach is the best compromise.

Finally, we have also measured that the mean execution time of STH, MAX, GMM and VUM algorithms is by 45 fps for PetsD2TeC2 and Indoor (of size 384x288), while the one for Synth12 and FStream (of size 640x480) is by 35 fps. These ex-

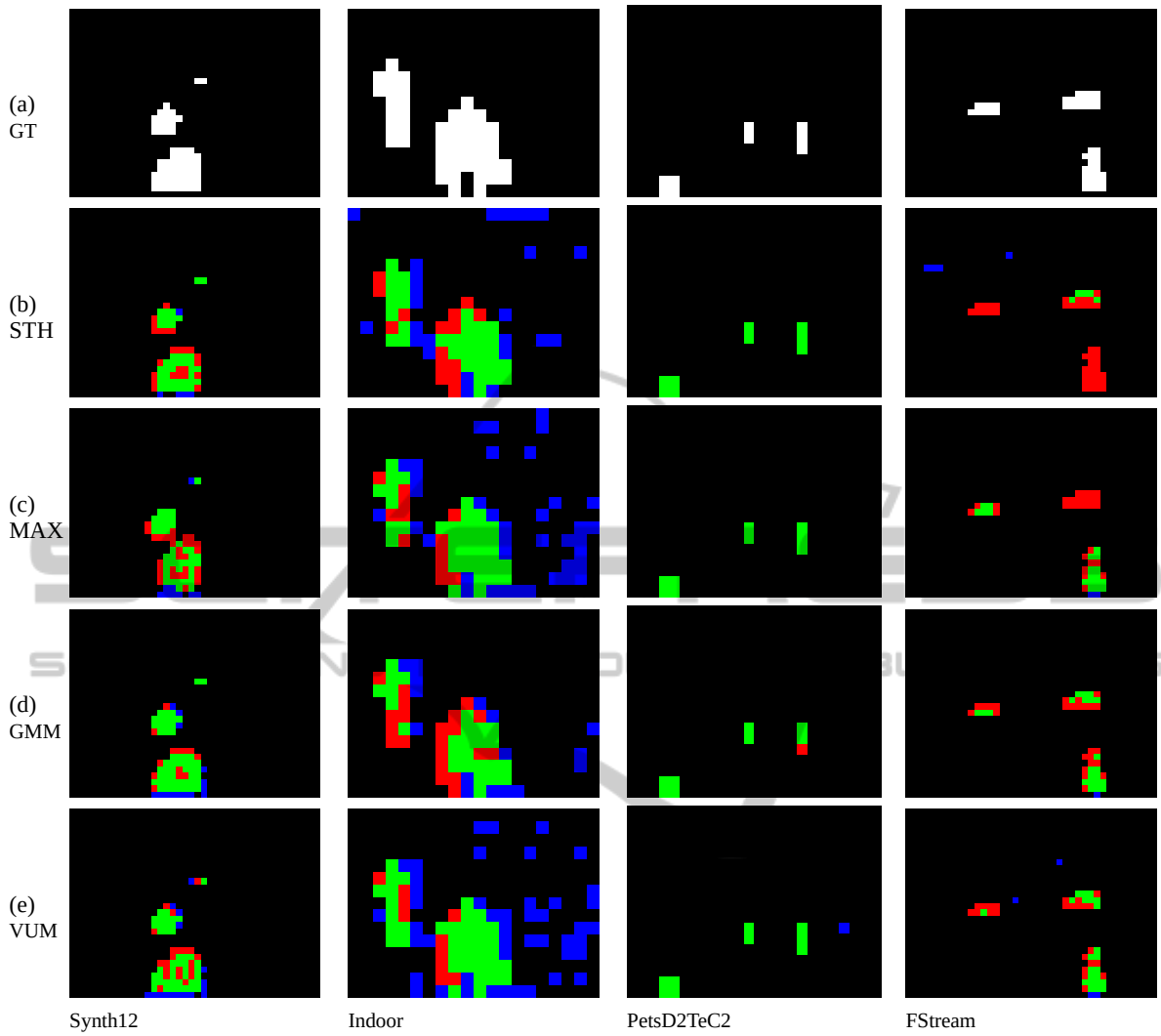


Figure 6: Outputs of the tested algorithms for (from left to right) Synth12, Indoor, PetsD2TeC2 and FStream. We have presented the result of STH algorithm (b) MAX algorithm (c), GMM algorithm (d) and VUM algorithm (e), and the associated ground truth (a). Colors represent: TP/green, TN/black, FN/red, FP/blue.

periments have been driven on a workstation with an 2.2 Ghz Intel® Core™ 2 Duo processor and 2 Gb RAM. Hence, our contributions can be implemented within real-time applications.

4 CONCLUSIONS AND FUTURE WORK

In this article, we have presented a novel algorithm for background subtraction in H.264/AVC bitstreams, by using the size of macroblocs. We have integrated a GMM into the background model for all frame modes (I, B, P). We have shown that the quality of extraction is improved thanks to recall/precision curves and numerical measures, in comparison with previous work and an other adaptive background model (VUM). Moreover, we can still treat videos in real-time applications since the execution time is very competitive.

As a future work, we plan to carry out a complete evaluation of the best parameters of our GMM algorithm w.r.t. various video datasets. We could integrate other adaptive background models to classify macrobloc sizes. Another interesting way could be to refine the result of our algorithm by developing a sub-MB extraction phase, and to test this method for (maybe long term) object tracking applications in video-surveillance. Finally, we are currently developing a real-time on-board application for video processing within a smart camera, which uses our contribution.

REFERENCES

Brown, L. M., Senior, A. W., Tian, Y., Connell, J., Hampapur, A., Shu, C., Merkl, H., and Lu, M. (2005). Performance evaluation of surveillance systems under varying conditions. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*. http://www.research.ibm.com/people vision/performanceevaluation.html.

Chen, Y.-T., Chen, C.-S., Huang, C.-R., and Hung, Y.-P. (2007). Efficient hierarchical method for background subtraction. *Pattern Recognition*, 40(10):2706–2715.

Cheung, S. S. and Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video. In *Proceedings of SPIE*, volume 5308, pages 881–892.

De Bruyne, S., Poppe, C., Verstockt, S., Lambert, P., and Van de Walle, R. (2009). Estimating motion reliability to improve moving object detection in the h.264/avc domain. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 330–333.

Dhome, Y., Tronson, N., Vacavant, A., Chateau, T., Gabard, C., Goyat, Y., and Gruyer, D. (2010). A benchmark for background subtraction algorithms in monocular vision: a comparative study. In *IEEE International Conference on Image Processing Tools, Theory and Applications (IPTA)*. To appear.

Goyat, Y., Chateau, T., Malaterre, L., and Trassoudaine, L. (2006). Vehicle trajectories evaluation by static video sensors. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 864–869.

Gruyer, D., Royere, C., du Lac, N., Michel, G., and Blosseville, J.-M. (2006). SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems. In *World Congress and Exhibition on Intelligent Transport Systems and Services (ITSC)*, pages 1–8.

Hayman, E. and Eklundh, J.-O. (2003). Statistical background subtraction for amobile observer. In *IEEE International Conference on Computer Vision (CVPR)*.

Kaewtrakulpong, P. and Bowden, R. (2001). An improved adaptive background mixture model for real-time tracking with shadow detection. In *European Workshop on Advanced Video Based Surveillance Systems (AVSS)*.

Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-time Imaging*, 11(3):167–256.

Liu, Z., Lu, Y., and Zhang, Z. (2007). Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain. *Journal of Visual Communication and Image Representation*, 18(3):275–290.

Mehmood, K., Mrak, M., Calic, J., and Kondoz, A. (2009). Object tracking in surveillance videos using compressed domain features from scalable bit-streams. *Image Communication*, 24(10):814–824.

Poppe, C., De Bruyne, S., Paridaens, T., Lambert, P., and Van de Walle, R. (2009). Moving object detection in the H.264/AVC compressed domain for video surveillance applications. *Journal of Visual Communication and Image Representation*, 20(6):428–437.

Poppe, C., Martens, G., Lambert, P., and Van de Walle, R. (2007). Improved background mixture models for video surveillance applications. In Yagi, Y., Kang, S., Kweon, I., and Zha, H., editors, *ACCV 2007*, volume 4843 of *LNCS*, pages 251–260. Springer.

Sigari, M. H. and Fathy, M. (2008). Real-time background modeling/subtraction using two-layer codebook model. In *International MultiConference of Engineers and Computer Scientists*.

Solana-Cipres, C., Fernandez-Escribano, G., Rodriguez-Benitez, L., Moreno-Garcia, J., and Jimenez-Linares, L. (2009). Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning. *International Journal of Approximate Reasoning*, 51(1):99–114.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptative background mixture models for a real-time tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 246–252.

Tuzel, O., Porikli, F., and Meer, P. (2005). A bayesian approach to background modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wiegand, T., Sullivan, G., Bjontegaard, G., and Luthra, G. (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576.

XiaHou, X.-J. and Gong, S.-R. (2008). Adaptive shadows detection algorithm based on Gaussian mixture model. In *International Symposium on Information Science and Engineering*.

You, W., Houari Sabirin, M. S., and Munchurl, K. (2007). Moving object tracking in H.264/AVC bitstream. In *Multimedia Content Analysis and Mining (MCAM)*, volume 4577 of *LNCS*, pages 483–492. Springer.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *IEEE International Conference on Pattern Recognition (ICPR)*.