Vol.2016-CVIM-200 No.27
2016/1/22

Random Block Background Modelling for Foreground Detection in UHD videos

Axel Beaugendre[1,a)] Satoshi Goto[1] Takeshi Yoshimura[1]

Abstract: Conventional foreground detection methods can take hours to detect objects in a single 4K Ultra High Definition (UHD) frame and their memory requirement is too high to be used without a huge investment in dedicated hardware systems. The proposed Random Block Background Modelling (RBBM) is a spatio-temporal method designed to update quickly the background image of UHD videos. By dividing the image into Mega-Blocks, themselves containing smaller Sub-Blocks and by using small randomly selected Sub-Blocks at each frame through a Gaussian average, the RBBM can accelerate the background modelling. Then, the RBBM is used in combination with a Block Propagative Background Subtraction method to detect the foreground. The proposed RBBM method has been compared with multiple other state-of-the-art works on 4 categories of UHD 4K scenes. The RBBM shows the best quality performances, the best ratio processing time per pixel/quality and a low memory requirement.

Keywords: UHDTV, background modelling, object detection, foreground,

1. Introduction

Ultra High Definition (UHD) will be the next standard in the near future. With resolutions of 7680×4320 for the 8K UHD and 3840×2160 for the 4K UHD, the quality and level of details have greatly been improved compared to current High Definition (HD). The 4K just arrived in the affordable mass market with television sets and video cameras and the 8K hardware will come soon after in 2020. Sensors see even faster improvement, very recently, a 250 Mega-pixel (19,580 × 12,600 pixels) CMOS sensor has been unveiled by Canon[*1]. Such sensors require adapted tools to exploited.

Background subtraction is one of the key techniques in computer vision. Mainly used for surveillance purposes, the background subtraction has also other specific applications as embedded system in cars to detect pedestrians. It usually consists of subtracting the current frame with a background model. All the remaining elements would be the detected foreground objects. The background image needs to be modeled very accurately to achieve the best results. Most of the efforts of the methods so far have been focused on this taks. State-of-the-art methods have been designed for the process of small resolution videos and all improvement made in the past decades were focused on the improvement on the quality of the detection before the reduction of the computational cost. There was not much need to do so when the dimensions of the videos were small but the situation has changed with the still recent High Definition (HD) and the coming UHD videos, it is now necessary to employ algorithms which take the processing speed into account.

Traditional background modelling methods are based on a full frame process. This approach implies that every pixel is loaded and processed at least once and multiple times in the worst case. This becomes an huge issue with the UHD videos which are more than a hundred times bigger than VGA videos, the computational time can go from seconds to hours per frame depending on the content and the complexity of the method used. In the past years, multiple approaches have been developped such as the statistical methods using one Gaussian [1], multiple Gaussians [2], [3], [4], [5]. Some others are based on the mean and variance over time [6], [7], some are non-parametric methods [8] or even based on colors and textures [9]. What all those methods have in common is that they process the entire frame both for the modelling of the background and for the subtraction of the background with the current frames. There are few spatio-temporal and block based approach works [10], [11] but their methods need to store a lot of spatio-temporal information which increases the ressources required. Also, even though they use Graphics Processing Units, the average speed for a 288 × 352 pixels video is about 40 fps so a processing time per pixel of 2.5×10^{-7} s. The requirements and the restrictions on the image size make those methods far from being able to be applied on UHD videos.

We propose in this paper a new background modelling methods called Random Block Background modelling (RBBM). The purpose of this approach is to divide the process of modelling the background into multiple frames by updating small random blocks at each frame. This way, instead of processing the full frame to update the background model, we compute it little by little thus saving computational time. The requirement for this approach is that the framerate is high enough so the difference between frames is not too important. Coupled with a fast back-

[1] Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan
[a)] a.beaugendre@kurenai.waseda.jp

[*1] http://www.canon.com/news/2015/sep07e.html

ground subtraction method like the Adaptive Block Propagative Background Subtraction (ABPBGS) [12], [13], it is possible to deal more various and difficult situations such as the detection of small or big objects of various shapes, traditional monitoring sequences or even handle illumination variation which are pretty common in surveillance videos.

This paper is organized as follow: Section 2 explains the RBBM process and its integration in the BPBGS. Section 3 presents the different results of the proposed methods compared to the state-of-the-art methods and finally, section 4 closes this paper.

2. Random Block Background modelling

Updating the background model is a costly task which can require a lot of ressources for very high resolution videos. To reduce the processing time, we proposed to divide the area to process into multiple small blocks and spread the updating on multiple frames. The resulting method is the Random Block Background Modelling (RBBM). By randomly selecting small blocks to update we can reduce the processing time required to update the background model. Since we cannot update all the background in only one frame this way, different blocks will be processed in the next frames. With a uniform distribution, all the background will be updated over time. Algorithm 1 presents the general process of the method.

Algorithm 1 Background modelling process

1: **Input:** Input image I_t, preview_map image P, Background image B, MegaBlocks MB; SubBlocks SB
2: **for all** MB **do**
3: **do**
4: Generate random SB index $id(x, y)$
5: **while** pixel value $P(id_x, id_y) > \tau$
6: Load the corresponding SB from the input image and the background image
7: Update the values of block pixels : $\mu_t^i = \rho I_t^i + (1 - \rho)\mu_{t-1}^i$
8: **for all** pixels from the preview_map P(x,y) **do**
9: $P(x, y) = P(x, y) - \alpha$
10: **for all** Selected SB **do**
11: $P(id_x, id_y) = 255$

2.1 Random Block Selection

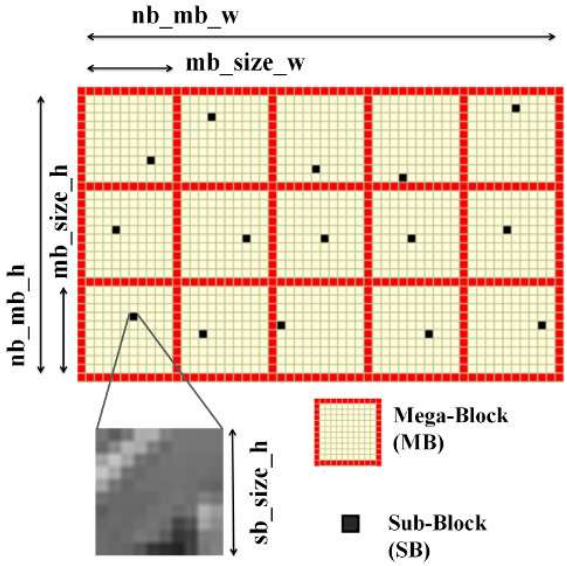
Every frame, a sample of the image is selected to update the background model. In order to obtain a uniformed model, we consider Mega-Blocks (MB) and Sub-Blocks (SB) defined as in Figure 1. The image of size *width* and *height* contains a total $nb\_mb\_w \times nb\_mb\_h$ MB, themselves containing $mb\_size\_w \times mb\_size\_h$ SB. The size of those SB is defined by the following equation:

$$sb\_size\_w = \frac{width}{nb\_mb\_w \times mb\_size\_w}, \quad (1)$$

$$sb\_size\_h = \frac{height}{nb\_mb\_w \times mb\_size\_w}. \quad (2)$$

For each MB, one SB is selected at random by a uniformed random id generator. The id $S_{id}(x, y)$ is made from

(a) Original image with block view



(b) Mega and Sub Block Model

Fig. 1: he Mega-Blocks (in red) contain several Sub-Blocks themselves containing pixels. Each time the background model is updated, one random Sub-Block (in black) per Mega-Block is chosen to update a part of the background model.

the sub block coordinates inside its mega-block as $S_{id}(x, y) \in ([0; mb\_size\_w[; [0; mb\_size\_h[)$. The randomness is necessary in order to not have systematically a temporal difference in the update between two adjacent blocks from difference MB.
Even though the selection is random we want to make sure that every blocks are processed at least once in a while. To decrease the chances of a block which has already been picked recently, we use a *preview map* which is a greyscale image of size $nb\_sb\_w \times nb\_sb\_h$ defined as:

$$nb\_sb\_w = nb\_mb\_w \times mb\_size\_w, \quad (3)$$

$$nb\_sb\_h = nb\_mb\_h \times mb\_size\_h. \quad (4)$$

Each pixel of the map represents one of the sub-blocks. If the pixel value of the selected SB is lower than a threshold value τ, the block can be updated. If the value is higher then a new block id is generated until we find a correct block. Once the chosen SB have been processed, the value of their corresponding pixel on the preview map is set to 255 and all the other pixels on the map have their value lowered by α. This forces a quicker rotation in the choice of the parts to update.

2.2 Block Updating

The region of the background corresponding to the selected blocks are then updated. Each pixel value μ_t from the selected blocks of the background model at time t is updated through the following equation:

$$\mu_t^i = \rho I_t^i + (1-\rho)\mu_{t-1}^i, \tag{5}$$

where ρ is the learning rate and I_t^i the corresponding pixel in the current frame. Since a block is not updated every frame unlike traditional methods such as the Gaussian Mixture Model, the value of ρ must be set to a higher value than the usual one (0.01). This value should now take into account the number of sub-blocks per mega-block:

$$\rho \simeq 0.01 \times mb\_size\_w \times mb\_size\_h. \tag{6}$$

To obtain a quite similar result after 100 frames, we can set $mb\_size\_w \times mb\_size\_h = 100$ leading to a learning rate of $\rho \simeq 0.10$. The number of blocks per MB influences the speed of the update while the number of MB and the number of Sb selected per MB modify the amount of pixels to update at each frame.

2.3 Background Subtraction

The background subtraction part is based on the Adaptive Block Propagative Background Subtraction (ABPBGS) [13] except that we now update the background model after each object detection with the RBBM. Even though both RBBM and the ABPBGS make use of blocks to process a small part of the frame, their respective block numbers and size are not related in any way. The two process are totally independant from each other. The BPBGS is suitable because the background modelling and the subtraction are different processes which is not the case is most of the background subtraction methods.

3. Experimental results

In order to test the our proposed RBBM, we tested the method on 10 custom sequences [14] in 4K@60fps (4K video with a framerate of 60fps) and we added the field sequence from the nebuta festival. We classified the videos into four groups: *Small Objects* (street1, street2, corridor, laser), *Big Objects* (screws, circle, 80percent), *Monitoring* (crossing, field) and *Illumination* (illu, parking). The difficulty is different for each group: very tiny objects (potentially less than 0.01% of the image) are the main target of the *Small Objects* group, whereas *Big Objects* contains objects either very big or multiple occupying a large space in the image (potentially taking all the screen). The videos of the *Monitoring* group contain various objects medium and small objects with surrounding background movements. Last, the *Illumination* category gathers videos in which the illumination can change quickly. All the Ground Truth information have been created manually.

The parameters for the ABPBGS and the RBBM used are as follow: $\Delta_r = 40$, $T = 30$, $w_{MIN} = 30$, $w_{MAX} = 480$, $h_{MIN} = 30$, $h_{MAX} = 480$, $n\_mb\_w = 16$, $n\_mb\_h = 9$, $mb\_size\_w = 10$, $mb\_size\_h = 10$, $\rho = 0.12$. We developed in C++ with the OpenCV library [*2], the computer used is a Quadcore i7@2.83GHz with 16 GB of RAM.

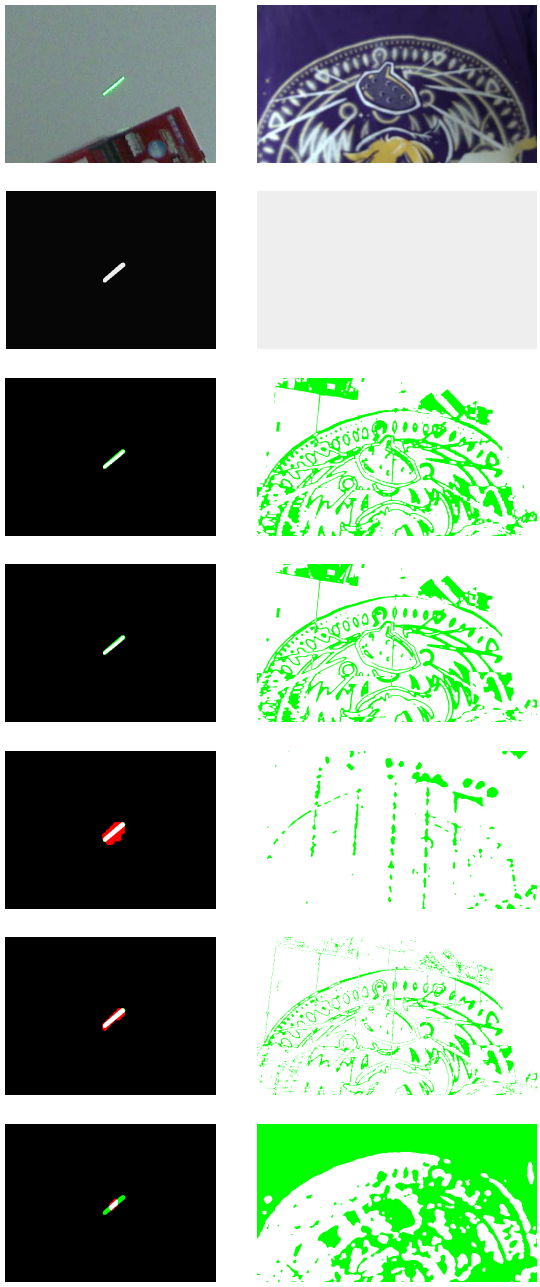


Fig. 2: Left column: example of the *Small Objects* category (*laser* sequence); Right column: example of the *Big Objects* category (*80percent* sequence); From top to bottom: ROI of the original image, ground truth image, RBBM, ABPBGS, PBAS, DPZivkovicAGMM, MultiLayer. Color legend: Black-TN, White-TP, Red-FP, Green-FN.

The proposed method has been compared to the ABPBGS [13], the classic BPBGS [12] and 8 state-of-the-art methods taken from the BGSLibrary [16], [17] for OpenCV. Those methods are: PixelBasedAdaptiveSegmenter (PBAS) [8], DPZivkovicAGMM-BGS [5] (an implementation of the Gauxian Mixture Model), MultiLayerBGS [9], DPAdaptiveMedianBGS [6], AdaptiveSe-

[*2] http://opencv.org

Table 1: Quality comparison of the proposed RBBM to the state-of-the-art methods. Best scores are in bold.

Method ID	Size	Recall	Pre.	F-Meas.	Sim.	SSIM	A-Score
RBBM (Proposal)	**4K**	0.430	**0.553**	**0.454**	**0.355**	**0.955**	**0.588**
ABPBGS [13]	**4K**	0.407	0.482	0.404	0.315	0.909	0.543
BPBGS [12]	**4K**	0.384	0.482	0.386	0.301	0.899	0.529
PBAS [8]	270p	0.394	0.374	0.350	0.287	0.926	0.521
DPZivkovicAGMM [5]	270p	**0.568**	0.336	0.350	0.275	0.871	0.499
MultiLayer [9]	270p	0.264	0.388	0.287	0.224	0.938	0.483
DPAdaptiveMedian [6]	270p	0.343	0.401	0.284	0.207	0.923	0.471
DPPratiMediod [7]	270p	0.445	0.316	0.286	0.211	0.895	0.464
DPWrenGA [1]	270p	0.532	0.275	0.287	0.213	0.880	0.460
AdaptiveSelectiveBGLearning	270p	0.412	0.278	0.267	0.209	0.884	0.453
IndependentMultimodal [15]	270p	0.513	0.198	0.227	0.168	0.867	0.421

lectiveBGLearning, DPWrenGABGS [1], DPPratiMediodBGS [7] and the IndependentMultimodalBGS [15]. Most of those methods are not able to process the videos from the UHD dataset in reasonable times therefore we compared our proposed method on 4K scale to the state-of-the-art processing 270p down-scaled videos. The comparisons are based on the standard quality metrics recall, precision, f-measure, similarity [16] and also on the SSIM [18]. To rank the different methods, the A-Score has been calculated:

$$A\text{–}Score = \frac{Similarity + F\text{–}Measure + SSIM}{3}. \qquad (7)$$

Table 1 and Figure 3 show the average quality metrics of the methods on all the tested sequences. The methods are sorted by rank from highest score to lowest. The RBBM outperforms all the other methods with an average A-Score of 0.588 while the ABPBGS based on a simple static frame subtraction has a score of 0.543 and the best state-of-the-art method, the PBAS, has a score of 0.521. We can observe that if the Recall score of the RBBM is not the best, its Precision score (0.553) is far beyond all the other methods, the highest state-of-the art method having a score of 0.401. This means that the RBBM is less affected by background noise than the other methods.

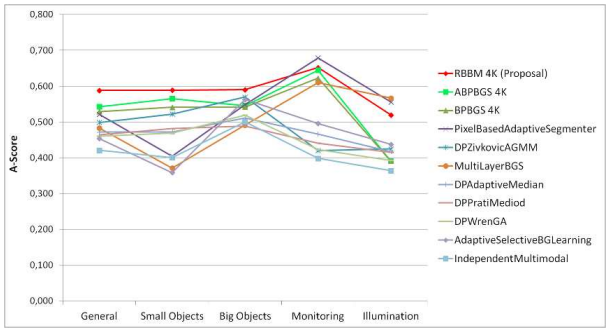


Fig. 3: Graph of the A-Score comparing the results of the RBBM processing 4K videos to the state-of-the-art methods processing 270p videos.

The first challenge was to detect small objects while keeping a very low amount of false positives (Figure 2-left). Indeed, morphological operations such as the erosion can remove some noise but at the condition that the objects of interest are much bigger than the noise to remove. In the sequences belonging to the *Small Objects* category, the objects of interest can be of the same size or even smaller than noise. The RBBM, which subtraction part is based on the ABPBGS can deal with those situations successfully. It obtains an A-Score of 0.589 while some of the best state-of-the-art methods like the PBAS or the MultiLayer are very efficient to get rid of the surrounding background noise but at the cost of missing parts or the entire object of interest if this one is too small. This is reflected on their scores for this category with 0.404 and 0.371 for respectively the PBAS and the MultiLayer. The scores of all the methods for the Big objects category of videos also show that small objects are their real weak point. Indeed, all of them can more or less successfully detect the object which takes all the image since their scores for this category are all much higher and rather close to each other, the main difference being in the ability of dealing with the pattern of the shirt (Figure 2-left). The videos of the Monitoring class contain not only objects of various shape and size but also much more background noise. Some methods like the DPZivkovicAGMM or the IndependantMultimodal show difficulties to deal with these issues and detect more false positives than the other methods (Figure 6-left).

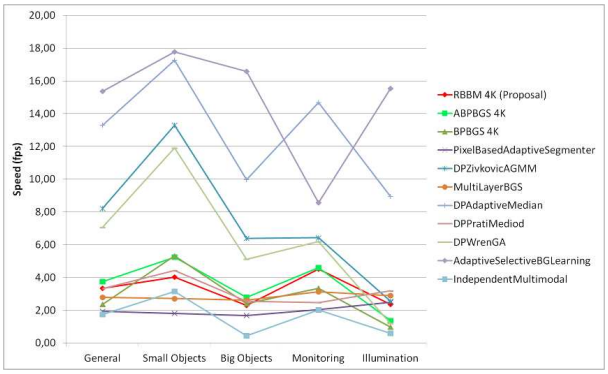


Fig. 4: Comparison of average speed of the different methods.

The last common issue in videos is the modification of the luminosity. Our dataset contains both slow variation and very sudden change of illumination (Figure 6-right). In this particular and very challenging case, we can clearly see which are the methods capable of adapting to the change. The MultiLayer shows very low amount of false positive but at the same time the object of interest is not entirely detected. On the other hand our RBBM can detect more parts of the object but are still affected by the sudden illumination variation. Finally, the other

methods are not able to deal with that kind of situation of very quick modification. It is very visible in Figure 3 in which we can observe 2 groups of methods for the *Illumination* category: the first one composed of the PBAS, the MultiLayer and the RBBM; and the second one with the other methods. The first group is far more performant than the later with a difference of almost 0.1 point of score between the RBBM (0.520) and the AdaptiveSelectiveBGLearning (0.437). Compared to the PBAS and the MultiLayer, the RBBM updating process is much simpler but it still show very satisfactory results on this illumination category of videos. We should also not forget that the RBBM deals with 4K images while the two other only process 270p videos. The variation of illumination is more diffused in lower resolution of images than in UHD images in which detailed parts have their luminosity distorted in each in different ways thus increasing the differences with a background model.

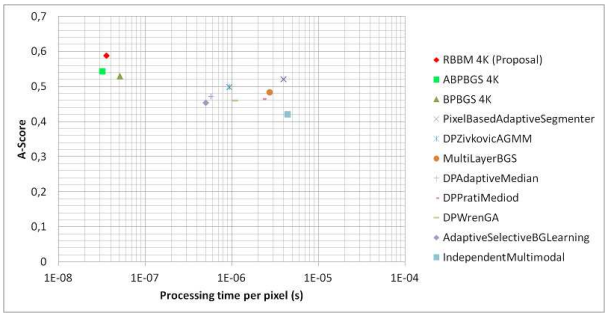


Fig. 5: Comparison of the processing time per pixel of the different methods.

Figure 4 show the average speed for the different categories of videos. In theory, the RBBM should slows down the ABPBGS since it is a background modelling which is absent in the original ABPBGS. This effect is visible in the *Small objects* videos where the ABPBGS is much faster than the RBBM. However, the speed for the illumination category shows that with the proposed RBBM makes the detection much faster than the ABPBGS. Noise and fragmented objects are the main reasons for a slow process. With the RBBM we have much less false positive detected objects which therefore do not need to be labeled thus making the computational time lower. Some methods such as the Adaptive Median or the Adaptive Selective Background Learning seem to reach high speed but it is mainly because they fail to detect the objects. We can also observe that even though the RBBM is processing 4K images, the average speed per frame (3.34 fps) is actually faster than the PBAS (1.94 fps) and the MultiLayer (2.79 fps) which are based on 270p images. In the three best state-of-the-art methods, only the DPZivkovicAGMM achieved a higher speed with 8.21 fps.

Figure 5 shows both the processing time per pixel and the average A-Score of the different methods. The lower the processing time per pixel the better and the best detection quality is achived by the methods with the highest A-Score. In this figure, we can observe that the RBBM is far above all other methods in quality score but also and mainly in its very low processing time per

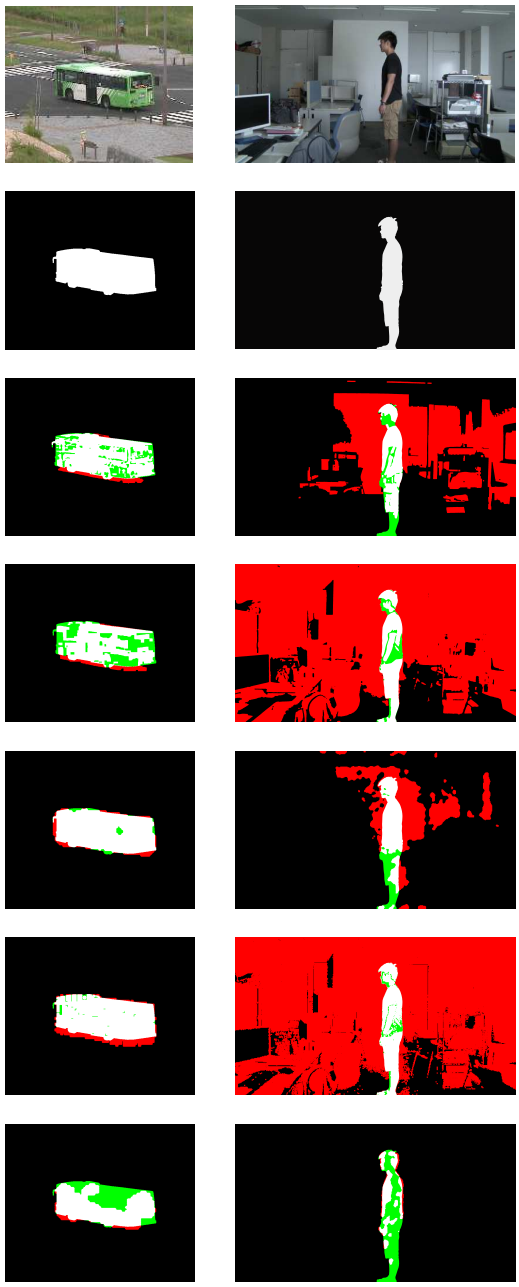


Fig. 6: Left column: example of the *Monitoring* category (*crossing* sequence); Right column: example of the *Illumination* category (*illu* sequence); From top to bottom: ROI of the original image, ground truth image, RBBM, ABPBGS, PBAS, DPZivkovicAGMM, MultiLayer. Color legend: Black-TN, White-TP, Red-FP, Green-FN.

pixel (3.6×10^{-8}s/pixel). In comparison, the PBAS (270p) performs at 4.0×10^{-6}s/pixel and the DPZivkovicAGMM (270p) at 9.4×10^{-7}s/pixel. The final observation is that the RBBM a good compromise with a speed almost equivalent to the ABPBGS for an even better quality.

In figure 7 we compared the memory consumption of the RBBM, the BPBGS and the three best methods: PBAS, DPZivkovicAGMM and MultiLayer. The experience on 8K scale has been done on the first 50 frames only for the state-of-the-art methods because they were not able to process the videos at their original scale. The second thing to mention is that the MultiLayer

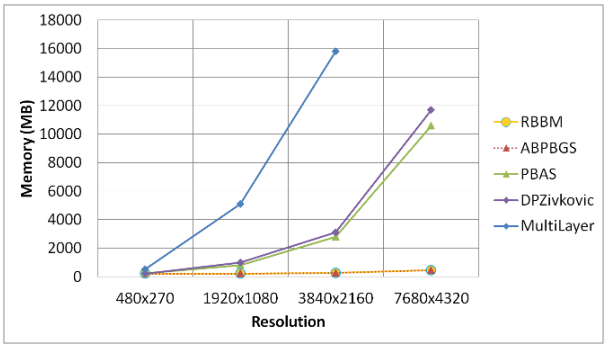


Fig. 7: Comparison of the memory consumption depending on the resolution used, from 270p to 8K UHD.

memory requirements for the 8K scale were above our 16 GB or memory available and completely crashed because of a lack of memory. For the other state-of-the-art methods, the memory consumption increased exponentially with the proportial growth of the dimensions of the video with a peak at 12GB for the 8K scale. The RBBM's memory requirement is very low since it just add the size of the preview map which has a total of few thousand pixels. It is negligeable in comparision to the size of a single 8K image stored in memory. Therefore, its total requirement is the same of the BPBGS for a maximum of about 450 MB for the 8K resolution.

4. Conclusions

With the Random Block Background Modelling, we proposed a method which divides the updating of the background model into multiple random blocks spread uniformly on the image and through time. This allowed us to save computational time which is vital when processing UHD videos. Coupled with the ABPBGS, the quality of the detection is increased with an average A-Score of 0.588. With an average speed of 3.34 fps and a processing time per pixel of 3.6×10^{-8}s/pixel, the RBBM is much faster than any other state-of-the-art method. Finally, by processing few tiny blocks, the RBBM does not require more memory than the ABPBGS with an average 450 MB for an 8K video.

References

[1] Wren, C., A.Azarbayejani, Darrell, T. and Pentland, A.: Pfinder: real-time tracking of the human body, *IEEE Trans. Pattern Ana. Mach. Intell.*, Vol. 19, No. 7, pp. 780–785 (1997).
[2] Stauffer, C. and Grimson, W.: Adaptive Background mixture models for real-time tracking, Vol. 2, IEEE International Conference on Computer Vision and Pattern Recognition (1999).
[3] Kaetrakulpong, P. and R.Bowden: An improved adaptive background mixture model for realtime tracking with shadow detection, *European Workshop on Advance Video Based Surveillance Systems (AVBS)* (2001).
[4] Z.Zivkovic: Improved adaptive Gaussian mixture model for background subtraction, *ICPR* (2004).
[5] Z.Zivkovic and van der Heijden, F.: Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction, *Pattern Recognition Letters*, Vol. 27, No. 7, pp. 773–780 (2006).
[6] McFarlane, N. B. and Schofield, C.: Segmentation and tracking of piglets in images, *Mach. Vis. Appl.*, Vol. 8, No. 3, pp. 187–193 (1995).
[7] Calderara, S., Melli, R., Prati, A. and Cucchiara, R.: Reliable Background Suppression for Complex Scenes, *ACM International Workshop on Video Surveillance and Sensor Networks*, pp. 211–214 (2006).
[8] Hofmann, M., Tiefenbacher, P. and Rigoll, G.: Background Segmentation with feedback: The pixel-based adaptive segmenter, *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 38–43 (2012).
[9] Yao, J. and Odobez, M.: Multi-layer background subtraction based on color and texture, *IEEE Computer Vision Recognition Conference (CVPR)* (2007).
[10] Berjon, D., Cuevas, C., Moran, F. and Garcia, N.: GPU-based implementation of an optimized nonparametric background modeling for real-time moving object detection, *Consumer Electronics, IEEE Transactions on*, Vol. 59, No. 2, pp. 361–369 (2013).
[11] Berjon, D., Cuevas, C., Moran, F. and Garcia, N.: Region-based moving object detection using spatially conditioned nonparametric models in a GPU, *International Conference on Consumer Electronics (ICCE)*, IEEE, pp. 359–360 (2014).
[12] Beaugendre, A. and Goto, S.: Block-Propagative Background Subtraction System for UHDTV Videos, *IPSJ Transaction on Computer Vision and Application*, Vol. 7, pp. 31–34 (2015).
[13] Beaugendre, A. and Goto, S.: Adaptive Block-Propagative Background Subtraction Method for UHDTV Foreground Detection, *IEICE Regular*, Vol. E98-A, No. 11 (2015).
[14] HD, W. G. L. and Dataset, K. U.: http://www.f.waseda.jp/goto/html/uhdtv_dataset.html.
[15] Bloisi, D. and Iocchi, L.: Independent Multimodal Background Subtraction, *Proceedings of the Third International Conference on Computational Modeling of Objects Presented in Images: Fundamentals Methods and Applications*, Rome, Italy, pp. 39–44 (online), DOI: 10.1201/b12753-8 (2012).
[16] Sobral, A. and Vacavant, A.: A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, *Computer Vision and Image Understanding*, Vol. 122, No. 0, pp. 4 – 21 (online), DOI: http://dx.doi.org/10.1016/j.cviu.2013.12.005 (2014).
[17] A., S.: BGSLibrary: An OpenCV C++ Background Subtraction Library, *IX Workshop de Vis£o Computacional (WVC'2013)*, Rio de Janeiro Brazil (2013).
[18] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E.: image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, Vol. 13, No. 4, pp. 600–612 (2004).