Fast UHD Background Modelling with Mixed Order Block Updates

Axel Beaugendre [1] and Satoshi Goto [1] and Takeshi Yoshimura [1]
[1]Waseda University IPSRC
2-7 Hibikino, Wakamatsu-ku, Kitakyushu city, Fukuoka, 808-0135, Japan

Abstract: Up to now, the majority of the background modelling methods are pixel-based and rely heavily on hardware optimizations to process standard definition videos in real-time. There are far too complex and require too much resources (time, memory, etc.) to be applied to Ultra High Definition (UHD) videos. We propose a fast background modelling method which divides the complexity through time via the update of multiple small blocks. By using mix of linear and random selecting order we insure that all blocks get updated in a known number of frames. Preliminary results show that we can reduce by at 80 times the modelling time compared to a full frame modelling and reach an average processing speed of 77 fps for 4K videos.

Keywords—**UHD, 4K, Background Modelling, Detection, MBBM**

1. Introduction

In the past years, the video resolution used in mass market products such as cameras or televisions has increased a lot with the venue of the Ultra High Definition (UHD) 4K (3840×2160 pixels). In the close future, the 8K will arrive with an even bigger resolution of 7680×4320 pixels. In computer vision and especially in background subtraction researches, most of the focus has been on videos with very small resolutions like standard definition (SD) VGA (640x480) or even smaller [1]. Some very few datasets contain UHD videos such as the [2] but the content is focused on encoding-decoding purposes thus unusable for object detection. Indeed, the quality of videos intended for television broadcasts is much higher and objects of interest are much bigger than in most of surveillance videos. Tools designed for SD videos are not suitable for UHD as they got more and more complex and need lots of resources in time and memory to deal with various scenes [3][4][5][6]. Some spatio-temporal block-based works [7][8] started to look for a way to speed up the detection process using spatio-temporal approach works but they require to store a lot of information and even if they use Graphic Processing Units they could only achieve 40fps for a 288 × 352 pixels video.
The biggest issues with existing methods is that they either do not correspond to UHD needs or to some situations such as modifications of the illumination or long stay objects. We present here an improvement of [9]. The previous version was based exclusively on a random order of the block selection. The proposed Mixed Block Background Modelling (MBBM) is an improvement which relies a double block selection, one using a linear order and the second using a pseudo random order. Unlike [9] we can insure that all part of the background is updated and boost the chances of blocks which would have been updated last otherwise.
Section 2 decribes the proposed system and section 3 presents the experimental results. Finally, section 4 concludes this proposal.

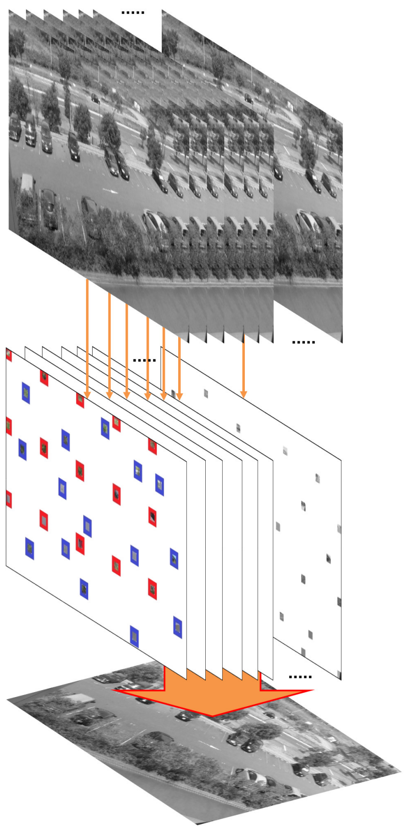


Figure 1: Method general overview.

2. Mixed Linear-Pseudo Random Block Background Modelling (MBBM)

With a high framerate (∼60 fps) and considering scenes where the background gets modified gradually, it is possible to divide the update of the background little by little by updating sparse small blocks SB in a region MB. Selecting blocks in a linear order guaranties that all blocks will be updates once after a maximum of T_u frames which corresponds to the total number of blocks in a region. A potential issue of processing blocks linearly is that the last block updated will always be next to a block which has not been updated for T_u frames therefore creating a constant temporal difference band which moves through time. On the other side, if blocks are selected using a strictly random order that issue can be avoided but then there is no guaranty that a block will be updated any time soon and a block could also be processed twice in a row. The method we propose here is a double pass block update

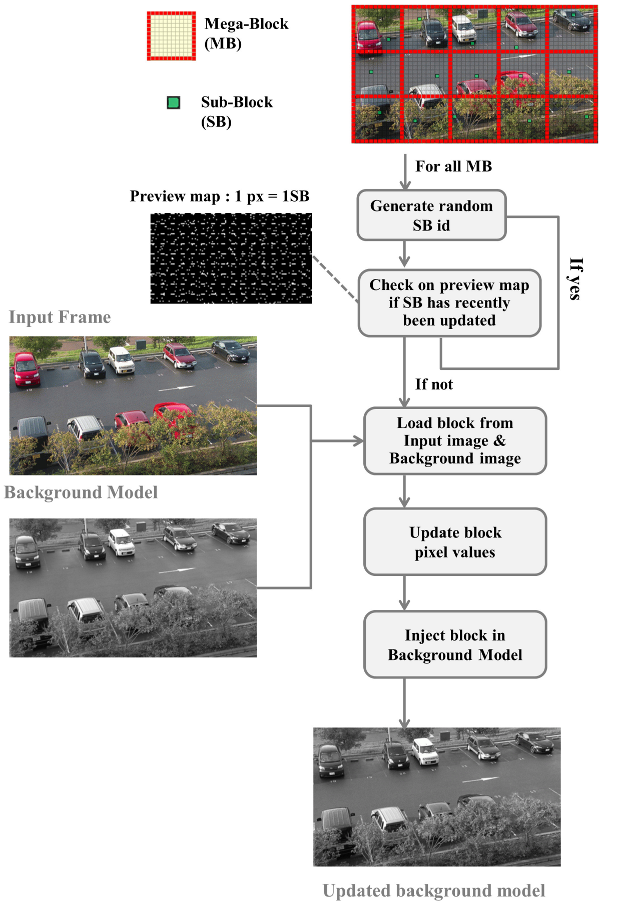


Figure 2: Pseudo-Random Block selection and background update.

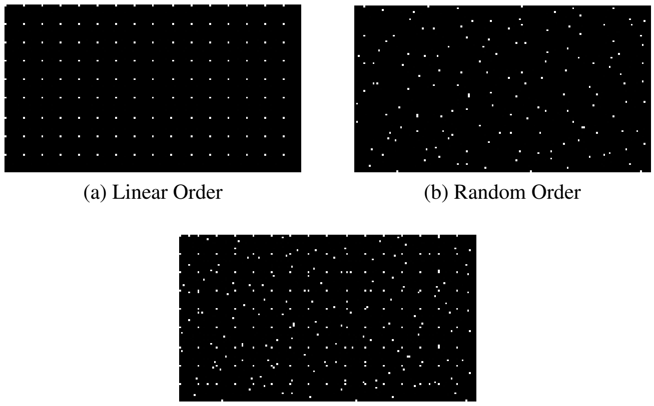
using a mix of linear and a pseudo-random selecting order (Figure 1. The pseudo randomization provides more chances to blocks which have not been updated recently and avoids re-updating those which have been processed in the last frames. Each frame is divided in regions MB as

$$MB = nb\_mb\_w \times nb\_mb\_h, \tag{1}$$

themselves containing small blocks SB with

$$SB = mb\_size\_w \times mb\_size\_h. \tag{2}$$

Since one SB is processed each frame, the total number of SB also defines the number of frame T_u required to update all the blocks at least once.
Figure 2 presents a more detailled flowchart of the random order part of the MBBM. The linear order follows the same principle except that the selection of the SB is done linearly. Moreover, the linear order does not require to check if the selected block had been processed recently. This verification step is done by using a *preview map*, an image of size $mb\_size\_w \times mb\_size\_h$ which pixel values represent the lastest SB updated. The purpose of this image to have an easy and visual way to verify that the next block selected has not been processed yet as we will randomize the id of the next block until it corresponds to an *old* block. When a block is selected to be updated, it also updates the corresponding pixel of the preview map with a max value of 255. The preview map pixel values are lowered at the end of the process by 10 so blocks which are not updated for quite some time will get darker and darker as opposed to newly processed blocks which will appear in bright. Finally, the background model regions corresponding to the selected blocks are updated pixel μ_t^i per pixel through the following equation:



(a) Linear Order

(b) Random Order

(c) Mixed Order

Figure 3: Representation of the preview map at frame 0. The white pixels represent the selected SB blocks which will be updated; Top: Linear and random order; Bottom: Mixed linear-random order.

$$\mu_t^i = \rho I_t^i + (1 - (0.01 \times mb\_size\_w \times mb\_size\_h))\mu_{t-1}^i, \tag{3}$$

3. Experimental Results

Our proposed MBBM has been tested with our custom 4K dataset [10] composed of 12 4K videos of various length, from 10 seconds to 40 minutes with a framerate of 60 fps. The PC used is a Quadcore i7@2.83GHz with 16 GB of RAM and the parameters of our methods are $n\_mb\_w = 16$, $n\_mb\_h = 9$, $mb\_size\_w = 15$, $mb\_size\_h = 15$, $\rho = 0.1$.
First of all, the modelling speed of our method is compared to a classic full frame equivalent (Table 1). The proposed solution is not barely faster but much faster. Indeed, we can observe that with an average speed of 76.9 fps our method is about 80 times faster and faster than the capture framerate of 60 fps.
We also compared the processing time of state-of-the-art background subtraction methods [5][4][3], our method combined to the Adaptive Block-Propagative Background Subtraction (ABPBGS) [6] and the full frame modelling combined with the ABPBGS. Unlike the previous comparison, those methods distinguish themselves in that they not only process all the pixels at every frames but also they mix the foreground detection results in the background modelling process. It results in an huge increase of the complexity and of the computational time. This is observable in Table 2 which

Table 1: Average speed comparison between our method and a full frame equivalent on the 4K sequences.

Method	Time/frame	Frames/s
Our method	0.013 s	76.9 fps
Full frame method	1.04 s	0.95 fps

Table 2: Average speed and estimated time to process 10s long 4K scene *parking_short*.

Method	Time/frame (s)	Frames/s	Time for video
Our Method+ABPBGS	0.117	8.55	70.9 s
Full frame+ABPBGS	1.08	0.92	11 min
PBAS	130.560	0.008	21 hours
MultiLayerBGS	3738.43	0.00027	26.1 days
ZivkovicAGMM	1363.01	0.0007	9.53 days

presents the estimated time required by each method to process a single 4K video of 10 seconds long (606 frames). We can observe that with our method, the total process is just 10 times slower than the length of the video while the other state-of-the-art methods require a minimum of 21 hours up to 26 days to process such a short video. This result shows how far the current state-of-the-art methods are from being able to process UHD videos. Those results confirm that the state-of-the-art methods are not suitable to UHD sources as it would indeed require years to process the 12 4K videos of the dataset for a total video length of about 1 hour.

Since it is impossible to process all the videos of the dataset to compute quality results, Table 3 presents a comparison of our MBBM on 4K video to 270p version of the state-of-the-art methods with a pixel-based quality comparison with the F-Measure (also called F1 score in some literatures) as well as the average speed for the whole dataset composed of 12 videos. The first and unexpected fact is that even though our MBBM uses the original 4K resolution, with an average detection speed of 3.78 fps it performs faster than two of the three state-of-the-art methods which were processing images 16 times smaller. Last but not least, the F-Measure of our MBBM is $+20\%$ to $+42\%$ higher than the state-of-the-art methods. It shows that not only the MBBM is much faster it also shows that it improves the detection quality results. Compared to the RBBM which is a totally random approach, the MBBM performs both 0.21 fps faster and 6.6% better. The slight increase in speed is explained by the improved quality as a better background model would reduce fragmentations and false positives. The speed of the labelling process to extract objects is scaled on the number of objects whether they are objects of interest or false positives from background noise. Therefore, a higher quality of detection also improves the speed of the total process. Finally, Figure 4 present the visual comparison on a frame from the *parking sequence*. We can observe that the MBBM is able to not only reduce the amount of false positives but also able to detect correctly most of the object of interest.

4. Conclusion

A fast mixed block background modelling method has been presented. Instead of updating the whole background model at once, we spread the update of the background through multiple frames by processing a limited number of small blocks each time. This new proposal using both linear and random block order insures that all the image gets updated. The main condition is that the input source has a high framerate so the full background model can be updated in a short amount of time. While other methods are very time consuming, the present method can perform up to 42% better than state-of-the-art methods and 80 times faster than an equivalent full frame background modelling.

Table 3: F-Measure and speed comparisons of the proposed method to some of the state-of-the-art methods.

Method	Size	F-Meas.	Speed (fps)
MBBM (proposed)	**4K**	**0.469**	3.78
RBBM [9]	**4K**	0.440	3.57
PBAS [3]	270p	0.390	2.05
MultiLayer [4]	270p	0.337	2.99
ZivkovicAGMM [5]	270p	0.330	7.79

References

[1] K. Toyama and J. Krumm and B. Brumitt and B. Meyers, “Wallflower: principles and practice of background maintenance,” *ICCV*, vol. 1, pp.255-261, 1999.

[2] J. Le Feuvre and J-M. Thiesse and M. Parmentier and M. Raulet and C. Daguet, “Ultra High Definition HEVC DASH Data Set,” *ACM Multimedia Systems Conference (MMSys)*, March, 2014.

[3] M. Hofmann and P. Tiefenbacher and G. Rigoll, “Background Segmentation with feedback: The pixel-based adaptive segmenter,” *CVPRW*, pp.38-43, 2012.

[4] J. Yao and M. Odobez, “Multi-layer background subtraction based on color and texture,” *CVPR*, 2007.

[5] Z.Zivkovic and F. van der Heijden, “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] A. Beaugendre and S. Goto, “Adaptive Block-Propagative Background Subtraction Method for UHDTV Foreground Detection,” *IEICE Regular*, 2015.

[7] D. Berjon and C. Cuevas and F. Moran and N. Garcia, “Region-based moving object detection using spatially conditioned nonparametric models in a GPU,” *ICCE*, pp.359-360, 2014.

[8] D. Berjon and C. Cuevas and F. Moran and N. Garcia, “GPU-based implementation of an optimized nonparametric background modeling for real-time moving object detection,” *IEEE Transactions on Consumer Electronics*, pp.361-369, 2013.

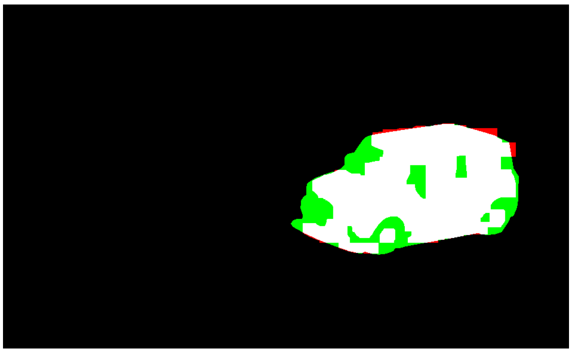
[9] A. Beaugendre and S. Goto and T. Yoshimura, “Random Block Background Modelling for Foreground Detection in UHD videos,” *IPSJ SIG-CVIM*, Jan. 2016.

[10] Waseda GOTO Lab’s HD and 4K UHDTV Dataset, *http://www.f.waseda.jp/goto/html/uhdtv_dataset.html*

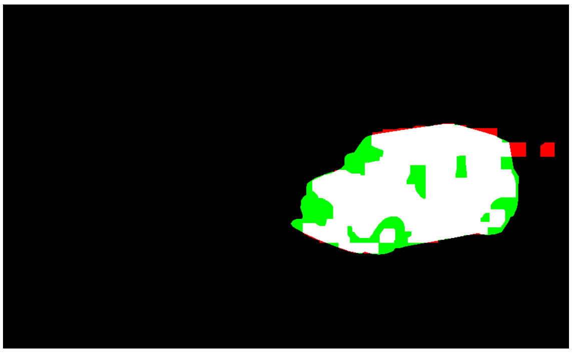
(a) Original 4K Image

(b) Region of Interest

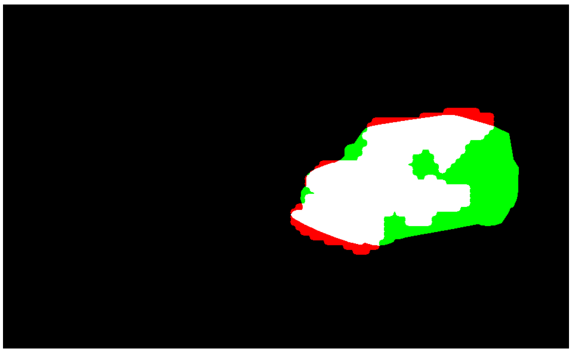
(c) Ground Truth



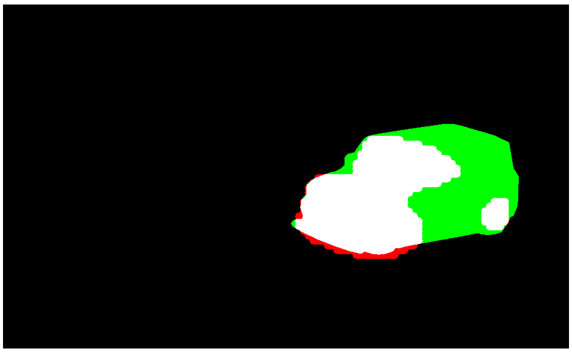
(d) MBBM (proposed)



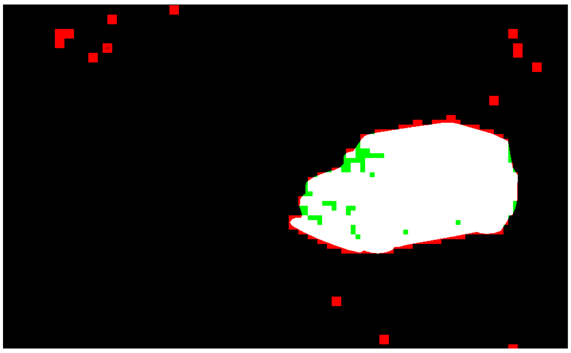
(e) RBBM



(f) PBAS



(g) MultiLayer



(h) ZivkovicAGMM

Figure 4: Foreground detection result on the 4K *parking* sequence. Color legend: Red-False Positive; Green-False Negative; White-True Positive; Black-True Negative.