Moving Object Detection in the HEVC Compressed Domain for Ultra-High-Resolution Interactive Video

Johan De Praeter, Jan Van de Vyver, Niels Van Kets, Glenn Van Wallendael, and Steven Verstockt

In: IEEE International Conference on Consumer Electronics (ICCE), 135-136, 2017.

De Praeter, J., Van de Vyver, J., Van Kets, N., Van Wallendael, G., and Verstockt, S. (2017). Moving Object Detection in the HEVC Compressed Domain for Ultra-High-Resolution Interactive Video. *IEEE International Conference on Consumer Electronics (ICCE)* 135-136.

Moving Object Detection in the HEVC Compressed Domain for Ultra-High-Resolution Interactive Video

Johan De Praeter, Jan Van de Vyver, Niels Van Kets, Glenn Van Wallendael, and Steven Verstockt
Ghent University – iMinds, ELIS – Data Science Lab, Ghent, Belgium
Email: {johan.depraeter, jrvdvyve.vandevyver, niels.vankets, glenn.vanwallendael, steven.verstockt}@ugent.be

Abstract—**Pixel-domain techniques are too computationally complex for automatic object tracking in ultra-high resolution interactive panoramic video. Therefore, this paper proposes a fast object detection method in the compressed domain for High Efficiency Video Coding. Evaluation shows promising results for optimal object sizes.**

I. Introduction

Advances in digital video capturing allow cameras to capture videos with increasingly high resolutions. Using stitching technology, the output of these cameras is stitched together as a panoramic video with a resolution far beyond HD. Since such an amount of data cannot easily be transported to viewers at home, a cropped version of the video is sent to the user. The user can then interactively pan and tilt a virtual camera to choose his desired viewpoint to have a greater sense of immersion. As an example, the user can decide to follow specific players in a sports match. However, manually tracking the players is cumbersome for the consumer and will negatively impact the interest to use such interactive video. As a solution, detection of moving objects can be used on the entire panoramic video in order to let the consumer automatically track players with his cropped view.

Although pixel-domain object detection techniques already exist [1], [2], these algorithms are evaluated on videos with a resolution smaller than 1920×1080 pixels, which is much smaller than the resolution of panoramic video. When applied to ultra-high resolution panoramic video, the computational complexity of these techniques will thus increase dramatically. As an alternative, we propose a compressed-domain object detection method based on the High Efficiency Video Coding (HEVC) standard [3]. This method uses motion vectors, which are already present in the panoramic video when it is encoded for transport over the network.

II. High Efficiency Video Coding

The HEVC encoder divides a frame into Coding Tree Units (CTUs), which are blocks of 64×64 pixels. These blocks can then be recursively split into CUs according to a quadtree structure down to a minimum size of 8×8 pixels. These CUs are further subdivided into Prediction Units (PUs) with the smallest possible size being 4×4 pixels. Each of these PUs is assigned a motion vector. These vectors are created by finding blocks that are similar to the considered block in one or more reference frames and can thus be an indication of movement of objects in a video.

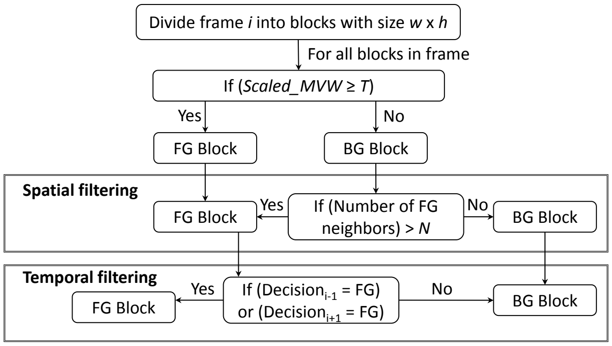

Fig. 1. Flowchart of the proposed method

III. Methodology

A motion vector can be associated with each position in a video frame based on the motion vector of the PU that the pixel at that position belongs to. If the x- and y-components of this motion vector have a value close to 0, it can be assumed that no motion is present at that position in the video. Therefore, in order to determine the amount of movement in an area σ with a size of $w \times h$ pixels, we propose to use a *motion vector weight* feature as defined in (1). This feature is calculated by summing the norms of the motion vectors for all pixels $i \in \sigma$ and normalizing this value through division by the area of σ. An extra division by 2 is added since a motion vector is 2-dimensional.

$$Scaled\_MVW = \frac{\sum_{i \in \sigma} \|MV_i\|^2}{2 * w * h} \tag{1}$$

with

$$||MV|| = \sqrt{x^2 + y^2} \tag{2}$$

The full algorithm as illustrated in Fig. 1 works as follows. Each frame of the video is uniformly divided into blocks with a configurable size of $w \times h$ pixels. For each of these blocks, the $Scaled\_MVW$ is calculated. If the resulting value is greater than or equal to an experimentally determined threshold T, the area in the block is classified as a foreground (FG) block. Otherwise, the area is considered a background (BG) block.

After the thresholding process, the detection is further augmented by applying a spatiotemporal filter as was done by Poppe et al. for object detection in the H.264/AVC compressed domain [4]. Spatiotemporal filtering is a combination of both spatial and temporal filtering. The spatial filter reduces the amount of BG blocks surrounded by FG blocks, which would cause holes in the detection. This is done by counting the number of FG neighbors and comparing this number to a parameter N. The optimal value of this parameter was experimentally determined to be $N = 4$ for all sequences, which is consistent with a spatial filter in H.264/AVC [4]. The temporal filter then further reduces the amount of misclassified blocks by filtering out blocks that are labelled as FG for only one frame.

IV. Results

A. Evaluation scheme

The algorithm was evaluated on three sequences of ultra-high resolution, each of them containing footage from sports games. Their resolutions were 10000×2248, 10000×1880 and 10000×2016 pixels for respectively basketball, hockey and soccer games. Only the area in the video that contains the playing field was used in the evaluation, since this is the area that viewers are interested in. The video sequences were encoded using version 16.5 of the HEVC Test Model with a configuration of an intra-frame followed by predicted-frames and a quantization parameter of 27.

The used sequences have been manually annotated by drawing bounding boxes around the moving players on the field. Each sequence consists of 10 seconds at a frame rate of 60 frames per second (fps) for the basketball and hockey content, and 50 fps for the soccer content. A representative set of 3 fps was annotated for each sequence, resulting in respectively 180 and 150 annotated frames.

Although bounding boxes are subjective, this effect was reduced by evaluating 75% of their central areas instead of the full 100%, as the aim is to identify the centers of the moving objects rather than their specific contours. The calculation of the amount of true positives and false negatives was thus restricted to 75% of the central area of the bounding boxes. The amount of false positives was determined using the regular 100% bounding boxes areas. Consequently, the remaining 25% of the bounding box area can be seen as a buffer zone.

B. Results

First, the optimal block size was determined by testing block sizes of $2^i \times 2^i$ pixels with i varying from 2 to 7. The optimal block sizes are 64×64 for the basketball and hockey content, and 32×32 for the soccer content.

Although the optimal block size depends on the content, two trends occur across all sequences. First, for lower block sizes such as 4×4 and 8×8, the algorithm has to make a decision based on a small amount of (possibly noisy) motion vectors. Therefore, higher block sizes prove to be more robust. Second, when the block size increases beyond 64×64, it becomes harder to describe the specific contours of the moving objects, resulting in a lower precision. This situation is especially the case for the soccer sequence, where the moving objects are relatively small compared to the other sequences. This results in smaller bounding boxes for players, which requires smaller blocks for the detection algorithm in order to avoid detecting many false positives.

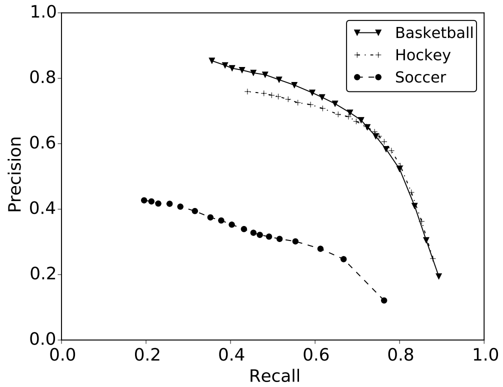

Fig. 2. Performance of the proposed method

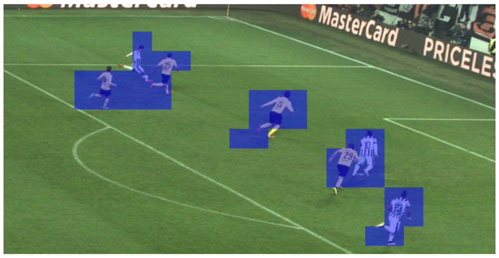

Fig. 3. Subjective example of soccer sequence with block size 32×32

Fig. 2 shows the performance of the proposed method on the three sequences with optimal parameters. By varying threshold T, a trade-off is made between precision and recall. A lower T results in more blocks being detected, resulting in a higher recall. However, this lower threshold also makes the algorithm more sensitive to noise, resulting in a lower precision. The proposed method performs well on the basketball and hockey sequences. On the other hand, the soccer sequence has a lower precision compared to the others due to many false positives. This is caused by the players being small compared to blocks of 32×32 pixels (see Fig. 3).

V. Conclusion

Performance of moving object detection for ultra-high resolution video in the HEVC compressed domain performs best for larger object sizes. As future work, the optimal block size and threshold of the algorithm should be determined automatically based on a pre-analysis of the size of the objects that should be detected by the system.

References

[1] D. Berjon, C. Cuevas, F. Moran, and N. Garcia, “GPU-based implementation of an optimized nonparametric background modeling for real-time moving object detection,” *IEEE Trans. Consum. Electron.*, vol. 59, no. 2, pp. 361–369, May 2013.

[2] C. Cuevas, R. Mohedano, and N. Garcia, “Statistical moving object detection for mobile devices with camera,” in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan 2015, pp. 15–16.

[3] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[4] C. Poppe, S. De Bruyne, T. Paridaens, P. Lambert, and R. Van de Walle, “Moving object detection in the H. 264/AVC compressed domain for video surveillance applications,” *J. Vis. Commun. Image R.*, vol. 20, no. 6, pp. 428–437, 2009.