INTEGRATED H.264 REGION-OF-INTEREST DETECTION, TRACKING AND COMPRESSION FOR SURVEILLANCE SCENES

Ivan Alen Fernandez[1], *Patrice Rondao Alface*[2], *Tong Gan*[3,4], *Rudy Lauwereins*[3,4] *and Christophe De Vleeschouwer*[1]

[1] ICTeam, Université catholique de Louvain, Louvain-la-Neuve B-1348, Belgium
[2] Alcatel-Lucent Bell Labs, Copernicuslaan 50, B-2018, Antwerp, Belgium
[3] Interuniversity Microelectronics Center, Kapeldreef 75, Leuven B-3001, Belgium
[4] Interdisciplinary Institute for BroadBand Technology, Gent-Ledeberg B-9050, Belgium

ABSTRACT

Motion detection and tracking is an important vision topic for many applications such as video surveillance. When this process takes place during video encoding and transmission, Regions-of-Interest (RoIs) turn to be a very useful tool in order to favor the encoding of such regions compared to the fixed background. In this paper, we show that, in conjunction with effective spatio-temporal filters, H264 Motion Estimation can be efficiently used for a robust and coarse-grain detection and tracking of moving objects. The integration of the compression and detection modules enables the prediction of ROIs positions from previous frames, offering therefore tracking at a very low computational cost. In the case of high resolution sequences affected by severe quality degradation (such as improper interlacing, light reflections and camera shaking), the global video compression ratio can be dramatically improved without damaging the ROI. This is especially the case when appropriate encoding options, i.e. appropriate Flexible Macroblock Ordering (FMO) types, are exploited. Different proposals are offered to maximize the quality of the RoI facing a dynamic constraint of the network bandwidth.

***Index Terms*—** H.264/AVC, motion detection&tracking, region of interest, constrained bandwidth, traffic monitoring

1. INTRODUCTION AND OVERVIEW

Recent years have witnessed a massive diversification and growth of applications in both wired and wireless networking technologies. Hence, despite the progress achieved both in terms of networking and video compression technologies such as H.264/AVC [1], there is still a gap between the bandwidth required by large camera networks and the transmission capabilities offered by wireless networks.

As a consequence, video analysis and processing have been largely investigated to identify spatial regions or temporal periods that deserve accurate representation and high compression quality compared to the rest of the captured scenes. Those efforts have been especially popular in the context of videosurveillance systems. A typical case is traffic monitoring with static cameras installed along tunnels and highways. The videos captured by different cameras have to be encoded in real-time, transmitted to the central server, and then processed for vehicle detection, identification, and multi-camera tracking. Because the analysis and recognition of vehicles is generally a computationally expensive process, it has to be implemented on the server side (assuming for example low-power capturing devices). However, bandwidth limitations do not allow transmission of high resolution and high quality content to the server. Hence, there is a strong incentive to identify the regions of the captured video that are really relevant to the subsequent analysis tasks. As an example, in [2] a scale-invariant detection algorithm reaching pixel accuracy is proposed to detect moving objects in the scene. The resulting bounding boxes of moving objects were coded as ROIs with a state-of-the-art H264 encoder while the fixed background was not transmitted. This has the advantage of lowering the required bandwidth for transmitting the coded bitstream. Nevertheless, for some application scenarios pixel-domain processing of the captured video is still too computationally expensive for the system embedded within the camera.

In particular, when the detection and tracking algorithms have to be executed together with real-time video encoding on an embedded device, there is a strong incentive to embed the detection and tracking algorithm within the video encoder, so as to take advantage of the analysis task already inherently performed by the encoder.

In this paper we propose integrated coarse-grain detection, tracking and compression technique to preserve a high video quality in the RoI of the H264 encoded video while lowering the required bitrate. Without the ROI information, the video content should be transmitted with a uniform high quality in order to make sure that noise does not prevent from missing moving objects or detecting motion noise. H.264 motion estimation (ME) information can be efficiently used to detect and predict the positions and sizes of the moving objects (RoIs) in advance for every frame with a lower complexity than pixel-domain approaches. In order to minimize the impact of false detection, the background is not skipped but still encoded with a lower quality. Our method then ensures a usage of the bandwidth and processing power (minimized

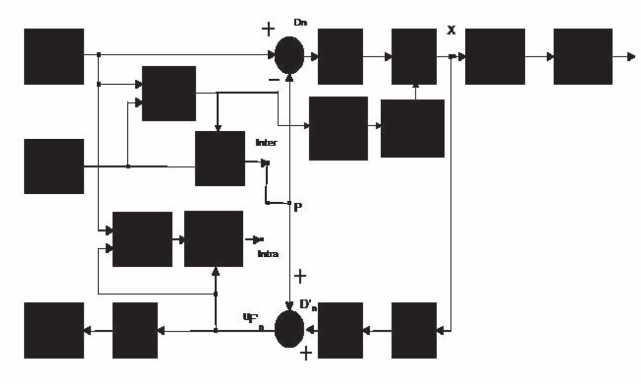


Fig. 1: Diagram of the standard H.264/AVC encoder emphasizing the proposed features.

Fig. 2: Quality preserved in a rectangular RoI encoded with FMO type 2, while the bitrate is drastically decreased.

complexity in the decoder) that is more relevant to the application needs than uniform quality transmission. In Figure 1, a modified scheme of the H.264/AVC encoder is depicted including our new features.

In our application scenario, very few macroblocks (MBs) are encoded from frame to frame in SKIP mode. This is because the sequences contain quality degradation due to previous video compression and/or improper interlacing. Furthermore, the lighting conditions can change dramatically on a frame-by-frame basis due to flashing lights or reflection of lights, thus reducing the temporal correlation between successive frames, particularly for tunnels sequences and therefore decreasing the SKIP mode in the quantization process. As a result, applying different quantization parameters in the RoI and the background becomes a must tool to drastically decrease the bitrate of the sequence without decreasing the visual quality of the RoI.

In our RoI encoding framework, MBs are mapped in different slice groups by implementing the FMO tool of H.264/AVC. FMO allows the ordering of MBs in slices according to a predefined map rather than using the raster scan order. While 1 of the slice groups consists on the region of des-interest(background), the rest are linked to the different RoIs. This mapping can also be used as an error-resilience tool, as originally defined, improving the robustness of the RoI to transmission errors while preserving a good trade-off with the global bitrate when applying an Unequal Error Protection (UEP) technique [3]. Furthermore, if different slice groups are encoded independently [4], the computation and complexity in the decoder can be dramatically decreased while processing the RoI. In figure 2 we present the result of applying our framework over a tunnel sequence.

Research on RoI over H.264 is basically focused on rectangular shapes by implementing FMO type 2. We demonstrate how this technique presents several constraints and disadvantages for video processing and network management. As an original contribution of our paper, we propose an explicit mapping of the MBs of the RoIs which alleviates these limitations by using FMO type 6. Largely discarded in the literature [5] due to the cost of the headers for its signalization in the decoder, we also demonstrate that under some network conditions and levels of quantization, this method can achieve a better overall compression than applying rectangular RoIs.

In the rest of the paper, Section 2 considers the detection of RoIs according to the ME, and investigates how to track the RoI along the whole video sequence. The different FMO compression techniques based on different quantization paradigms and their limitations are explained in Section 3. In Section 5, our framework is compared with previous works. Experimental results illustrating the performance gains and limitations of the proposed schemes are presented in Section 4 while Section 6 concludes.

2. ROI DETECTION AND TRACKING BASED ON MOTION ESTIMATION

Our algorithm is integrated in a state-of-the-art Baseline-Profile H264 encoder, with Flexible Macroblock Ordering (FMO) capability. The aim is to detect and track moving objects over which ROI-based compression is applied. As the encoding process is based on the FMO tool and its granularity is at MB level, the procedure for detection and tracking doesn't require a finer accuracy than this level. By re-using Motion Estimation Information, the algorithm benefits from a reduced complexity compared to an independently computed pixel-based models of the scene. The algorithm, similar to the one proposed by [6] (further comparison in Section 5), is composed of four steps as follows:

- *Background Segmentation*: In this initial step, the background is estimated at MB level by comparing the reconstructed previous frame and the original current frame with a quantization-aware threshold. Hence, MBs are classified in background and foreground classes.

- *Temporal and Spatial Filtering*: Morphological operators are applied to the foreground MBs, to filter out the isolated ones. The output of this step is defined as the *active MBs*. Since meaningful moving objects have a minimal lifetime in a video surveillance context, a temporal filter is used to filter out false positive detection due to sporadic noise.

- *MBs clustering and group labeling*: Active MBs are clustered in groups of connected MBs. Each group is then treated independently.
- *Object collision and speed computation*: Information about previously detected moving objects (position and motion) is compared to the current foreground detection to propagate RoIs positions to the current frame.

Each one of those four steps is described in details in the following. Actual specificities and contributions of our work are also highlighted.

2.1. Background Segmentation

Background segmentation builds on motion estimation. A simple subtraction between two consecutive frames is considered to detect foreground motion. This is measured at MB level by the *Stationarity* S_t, which is given by the difference between the original input frame at time t and the reconstructed frame at time $t-1$ in all the pixels:

$$S_t(i,j) = \sum_{k=0}^{15}\sum_{l=0}^{15} |x_t(i+k,j+l) - x_{t-1}^r(i+k,j+l)| \quad (1)$$

Where i and j denote the indices of the MB. Stationarity depends on the compression parameters such as the quantization parameter (QP) in the previous frame. A threshold can be set in order to easily detect non-moving regions taking into account noisy artifacts and quantization errors. The threshold should obviously account for the quantization parameter. Therefore, we rely on our earlier investigation [7], analyzing how the quantization distortion is affected by the quantization parameter, and define the threshold to be:

$$t_{QP} = 256 qStep(QP)/2 \quad (2)$$

Where *qStep* is the quantization step defined in the H.264 standard [1]. This method is certainly too simple and lacks of robustness against noise or camera vibrations as shown in Figure 3. Some refinements are proposed in the subsequent sections. The motivation for comparing every new frame with the previous reconstructed instead of the original one is twofold: it saves memory requirements and computational resources(computation already fully-integrated for every *P frame*), and it assigns less conservative thresholds to the RoI MB, which implicitly increases the probability for a RoI MB to stay in a RoI.

2.2. Temporal and spatial filtering

Spatial filter aims at discarding MBs declared as part of the foreground that are associated to small objects, such as the

(a) (b)

Fig. 3: Result of the stationarity measured on noisy input.

ones caused by glare lights or other kinds of noise. An operator, similar to the morphological erosion, is applied in order to define which MBs from the previous step are going to be considered as *active*, to define the RoI in the current frame at last. Smaller objects to detect, are consider to occupy at least a region of 2×2 MBs. In practice, the filter consists on that every MB in the foreground should form at least one box of 2×2 MBs with any of the neighboring MBs where at least 3 out of the 4 MBs must are in the foreground. If so, the MB is considered as *active*. Formally, the filtering proceeds as follows: The 4-neighbors are checked first by using the 3×3 morphological cross as structured element as depicted in Fig. 4. MB is declared active if there are foreground MBs in both of the spatial axis.

Fig. 4: Spatial filter. 4-neighbors are checked first.

$$MB^{Act}_{f_{i,j}} \rightarrow (MB^f_{(i-1,j)} \cup MB^f_{(i+1,j)}) \cap (MB^f_{(i,j-1)} \cup MB^f_{(i,j+1)}) \quad (3)$$

Otherwise, if is still possible to satisfy the conditions of the filter, the rest of the neighboring MBs are checked. This filter is less aggressive in the borders of the picture.

Although some blocks corresponding to a real motion object might be filtered out in this step, this rarely happens. In contrast, many of the noisy MBs that were previously considered as foreground are filtered and removed. To cope with camera shaking and remove the noise that has passed the spatial filtering, additional temporal filtering is applied.

As in [6], an observation period is defined to consider the new active MBs as a real object with motion. Defining this period as *Tobs*. If during the *Tobs* interval, the occurrence of the groups of *active MBs* overlapping and updating the *active MBs* that initially define an object overcomes a threshold, then the object is declared as *Real*. After declaring an object as *Real*, the RoI starts to be applied over it while is tracked with the same technique (*active MBs* updated with new overlapping active ones).

As depicted in Figure 5, the *active MBs* related to an object don't define the RoI until the observation period is passed.

(a) (b) (c)

Fig. 5: The temporal filter prevents noise but also delays for every object the RoI-based compression. In the left, the active MBs of a new object entering the scene. In the centre and right, RoI compression is applied after the temporal filter.

While this process creates some false negatives when an object enters the scene, filters all the false positives regarding noise that passed the spatial filter.

2.3. MBs clustering and tracking.

In this section we denote a group of *active MBs* to be a set of connected foreground MBs. A recursive morphological operator is used to label the MBs into active groups. Different active groups depicted in Figure 6. The active groups are then tracked along the time, simply by matching the groups that (partly) overlap in consecutive frames. Our experiments reveal that in practical surveillance scenario (15-25 fps, rather far view), the overlap between objects in consecutive rames is sufficient to support our tracking method.

Fig. 6: Different labels are exploited to perform tracking and temporal filtering.

In case that an object tracked with RoI stops, there won't be any active group of MB overlapping it for some frames. After several frames in the same condition, the RoI is disallowed. This procedure covers the case an object disappears from the camera view.

Sometimes objects might merge in the image (overlapping of their *active groups*) even if they are not very close, due to shadows, sporadic noise, etc. At the same time, these objects might split after some frames. In case of objects merging, this information is stored. In case of splitting of an active group of an object in two or more, if not merging is recorded, the smaller active groups of MB will pass the temporal filter as they might be due to noisy artifacts. This might create a certain number of false negatives but is needed to avoid noisy false positives and to deal with the fact that when two objects split due to different speeds, the active MBs that belong to them are merging and splitting during some frames. To avoid this issue, the splitting could be disallowed but this could decrease the compression efficiency as depicted in Figure 7.

(a) (b) (c)

Fig. 7: Two objects enter together in the camera view but progressively move away due to the difference of speeds. In the centre, lost compression as the splitting is not defined.

As the RoI detection relies on the ME, which in turns is computed during raster encoding of MBs, the RoI mask computed on a frame has to be used on the next frame. Hence, it is necessary to compensate for the potential RoI movement occurring between the current frame and the next one. In practice, this is done by simple extrapolation of the displacement measured between the current and previous RoI.

3. ROI-BASED COMPRESSION

The tracked RoI objects are encoded independently from the rest of the picture by using different *slice groups*. A finer quantization (lower QP) is applied over the tracked RoIs, based on standardized FMO techniques. Two different approaches are considered: FMO type 2 and 6. While the first one is commonly used in the literature for RoI definition, we observe that the second one presents several advantages under certain scenarios and networks conditions.

3.1. MB quantization based on FMO type 2

The slice groups in FMO type 2 are based on rectangular boxes. As depicted in Figure 8 every slice group belonging to a RoI is defined by a top-left MB and a bottom-right MB. This information that has to be sent to the decoder through the update of the Picture Parameter Set (*PPS*) in case of any change every frame. The bitrate overhead is lower than 1% [8]. The last slice group is formed by the rest of MBs that are not specified inside the RoI boxes and therefore represents the background.

(a) (b)

Fig. 8: FMO type 2 implementation and result.

An external RoI (*xRoI*) has been applied in [3] to automatically smooth out the disturbing edge between the ROI and the non-ROI areas with an intermediate QP. In our case, some small parts of the object might stay outside from the RoI for three reasons: the aggressive thresholding and filtering and the rounding of the speed of the object at MB level. In order to overcome this issue, an optional xROI can be defined for some of the cardinal directions receiving the same QP than the rest of the RoI.

$$MB_{tLeft}^{'} = MB_{tLeft} + 1 * xEast - xNorth * nMbH \quad (4)$$

$$MB_{bRight}^{'} = MB_{bRight} + 1 * xWest + xSouth * nMbH \quad (5)$$

Where *nMbH* is the number of MBs in the width of the picture. *xEast*,*xNorth*,*xWest* and *xSouth* are 1 if the MBs of the border of the predefined RoI in each direction overcome a threshold of *active MBs* which is set to a 60%.

As shown in Figure 9, some borders of the object are not included in the *active MBs* of the object but by using the statistical xRoI the default RoI for this object is extended in some of the directions. As a result, the entire object is included in the final RoI, but although this prevents leaving parts of the moving object outside the RoI, it also enlarges the RoI detection, damaging the compression, as some axis are extended while not being necessary, as for this case: north and east direction.

(a) (b)

Fig. 9: Active MBs and xRoI applied in the next frame.

3.2. Limitations when using FMO type 2

According to H.264 standard specifications only 8 slice groups can be defined through FMO type 2. This means that not more than 7 rectangular boxes can be defined in the picture. The problem of using FMO type 2, although a lot of bitrate is saved in the signalization, is the few precision of the definition of the RoI and the resulting cost in bitrate. This becomes a real issue, in particular:

- The RoIs overlap, and form a single active group, which in turns results in a large rectangular box (see Figure 10).
- If the colliding regions are mapped independently to save bitrate, the computation is increased. Furthermore, the number of RoIs that can be defined becomes an issue. For two overlapping RoIs, 1 extra RoI is used [9].

For those reasons, FMO type 6 is considered next.

Fig. 10: The active MB groups of the objects are overlapped. One big box has to be defined with a huge compression cost.

3.3. Implementation and analysis of FMO type 6

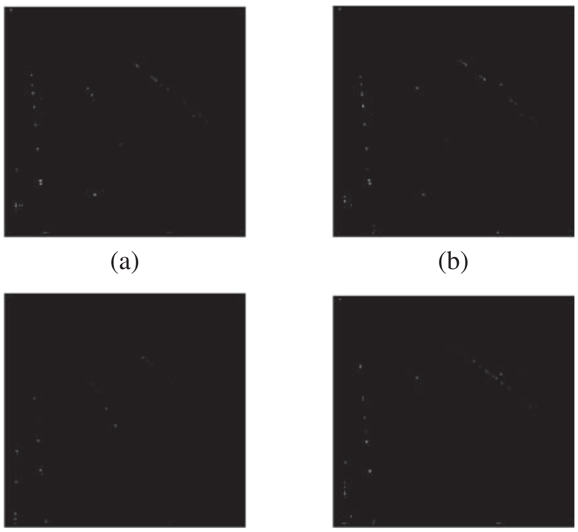
FMO type 6 is known as the *explicit mapping type*. Consequently, for every single MB its slice group has to be explicitly defined. The typical bitrate overhead lies around 10% of the total bitrate [8]. In this paper, we demonstrate that this cost can be overcome in some circumstances while the constraints of FMO type 2 disappear and some other compression advantages are present:

- All the RoIs can be defined in the same slice group or in different ones (without limitations). This flexibility is profitable in error-prone networks, where all RoIs can be processed in the same NAL unit.
- All the problems of overlapping RoI or splitting objects from type 2 disappear. Not extra MBs from the background have to give a particular shape to the RoI.
- The process of decoding is faster as the decoder doesn't have to create a mapping of all MBs, as the MB type information is already included in the PPS.
- The RoI is defined more accurate in FMO type 6. Hence, he decoder has to process less data, saving many resources and time.
- The inter/intra prediction works better because the slice groups division provides more accurated object segmentation.
- The MBs could be classified in two different QP levels without using FMO type 6. But with FMO 6 the bitrate cost is lower preserving all FMO features.

In our framework, we define a unique slice group for all RoIs. This reduces the FMO type signaling information to 1 bit per MB [1]). The MBs that are assigned to the RoI directly correspond to the motion compensated active group MBs.

To cover the potential holes or missing parts of the object, closing and dilation morphological operations such as closing and dilation are applied with a 3×3 cross (4-neighbors) and a 3×3 rectangule (8-neighbors) as structural elements. With the dilation, the borders of the objects are extended as with the xRoI in type 2.

The dilation with the cross is as precise and more accurate than the same operation with the rectangle. The closing operations are very accurate but also have missings in the borders of the objects. A combination of the 8-neighbors dilation with the 4-neighbors erosion surpasses this issue achieving the best accuracy of all the methods with a good precision. This precision is similar to FMO type 2 without the xRoI. In Figure 11 some cases are depicted.



(a) (b)

(c) (d)

Fig. 11: RoI MBs with FMO 6. On top, dilations of 8 and 4 neighbors. Below, closing-8 and the hybrid closing approach. The best methods, on the right, are compared in the Results.

4. RESULTS AND DISCUSSION

Our testbench, Figure 12, consists of multiple tunnel and outdoors surveillance videos taken by traffic monitoring cameras at a spatial resolution of 720x576 and a frame rate of 25 frames per second (fps).

Fig. 12: Distortion of test videos: improper chroma component, blocking effects and improper interlacing.

The next results analyze the implementation of FMO type 2, including the additional xRoI feature. We applied a static QP value of 32 for the background. Through several simulations, the mean increment in the global processing time due to RoI computation was less than 0.1% compared to the initial compression time, and the increment of bitrate associated to slice groups was around 0.37%.

The accuracy of the motion detection&tracking was analyzed for the indoors and outdoors sequences separately,

(a) (b) (c)

Fig. 13: Dark areas and chroma degradation (c) in tunnels and light reflections (b) causing some errors in the ME.

based on a manually generated ground truth. The objects that are very far away respect to the camera are not considered as they are not significantly moving with respect to the camera. The precision *P* and the recall *Re* are computed as it follows:

$$Re = Hits/(Hits + FP + FN + FpN) \tag{6}$$
$$P = Hits/(Hits + FP) \tag{7}$$

Where *FP* are the false positives or RoIs that are encoded without any object inside. The *Hits* are successes of objects tracked with a RoI. *FN* are the false negatives when an object is in the image but not RoI is applied and the *FpN* are the cases where less than 90% of the object is tracked by the RoI. The results are presented in Table 1.

Type of Sequences	Hits	False Positive	False Negative	False Partial Negative	**Precision** (%)	**Recall** (%)
Outdoors	1684	114	16	76	95.68	92.73
Outdoors not splitting	1712	86	16	76	95.75	94.34
Indoors	1496	186	274	152	90.78	76.48

Table 1: Performance of the motion detection&tracking

We observe that the results are much better outdoors than inside the tunnel were the image degradation and the noise (light reflections) are much larger. The splitting of objects is not considered in the tunnels, due to the *aperture problem* in the darkest parts of the image[1].

In Figure 13 some weak points are illustrated. For example, in Figure5 we observe that an object entering the image creates false negatives until the temporal filtering ends and in Figure 13 *a* an object leaving the image creates some false positives for a few frames.

Regarding compression effectiveness, we observe in Figure 14 that only encoding the RoI with a fine QP significantly reduces the bitrate compared to a scenario for which the entire image is encoded with a fine QP. A QP equal to 12 results in 85% bitrate savings. This result makes sense due to the image resolution (1620 MBs in the image), the low percentage of MBs in the RoI (an average of less than a 10%) and the fact that most of the background is never encoded in SKIP mode due to the quality degradation of the input.

[1]The splitting has some false negatives due to the temporal filter but a better compression after. Aperture problem occurs in large parts of the image with uniform color where motion can't be computed

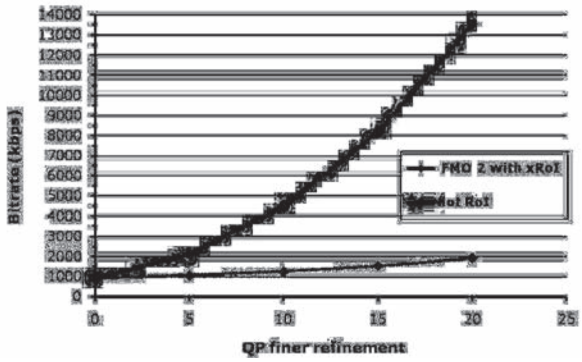


Fig. 14: Reduction on bitrate when using RoI compression.

4.1. Comparison of FMO compression techniques

In our simulations, the cost of the headers of FMO type 6 was around 7% of the global bitrate when not improvement of quality was applied over the RoI. FMO type 2 is compared now with the best approaches of type 6. Type 2 with xRoI has the same robustness as type 6 with 4-neighbors dilation (both of them expand the borders of the detected active MB). In the case of type 2 this expansion is more selective. Without applying the xRoI, the recall through the simulations was similar to the dual operation in type 6 dilation 8-neighbors and erosion 4-neighbors that is used to close the holes of the objects. The rest of the results that are presented, are based on frames where with all the RoI encoding techniques the precision and recall were 100%. In Figure 15, the total number of MBs encoded in the RoI for all these techniques through several frames is depicted. The decrement of MBs in the RoI with FMO type 6 is 20%-30% for the two comparable cases, as shown in Figure 15. This difference is lower between the dilation and type 2 with xRoI, as this not necessarily expands in all directions.

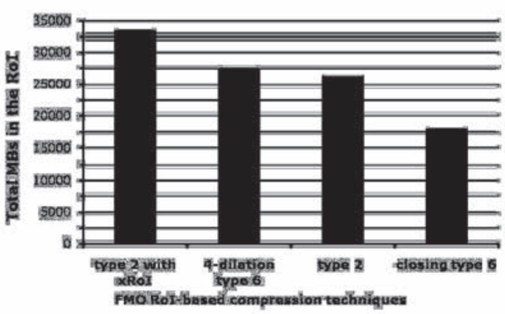


Fig. 15: Comparison of FMO techniques for RoI accuracy.

With these results, it is expected that although the compression is worse applying the same quality in the RoI and the background, there will be a crossing point increasing the quality in the RoI where the compression with FMO type 6 will be better. This is depicted in Figure 17. FMO type 6 achieves a better compression in both of the cases when the QP is between 10-15 units finer than the background. As a result, with a fixed low-quality in the background, a combination of both FMO types gives the best compression performance depending of the limit of bandwidth of the network. Therefore, if we also want to approach a certain bitrate, depending on the bandwidth available one of the two techniques is able to achieve a finer QP in the RoI and consequently better quality. If this limit of bitrate is dynamic, and varies between the two regions where FMO type 2 or 6 is better, then the selective combination of the two methods gives the best performance results rather than just using one of them.

5. RELATED WORKS

A lot of work has been presented about motion detection in H.264. While a lot of low complexity methods rely on compressed features: motion vectors or DCT coefficients [10], they are not robust enough to achive a good precision and recall results. Our approach for motion detection relies on a previous publication[6], however their approach, in the decoder side, is not enough to handle our noisy input as depicted in in 16. Their background extraction is based on SKIP MBs and a simple spatial filter. Regarding RoI compression, a similar approach in [11] doesn't perform object tracking just a simple FMO type 2 for one object. Furthermore, as an original contribution, we also apply FMO type 6, while discarded in [5], we demonstrate its profits for a better video compression and a non-constrained video encoding scheme.

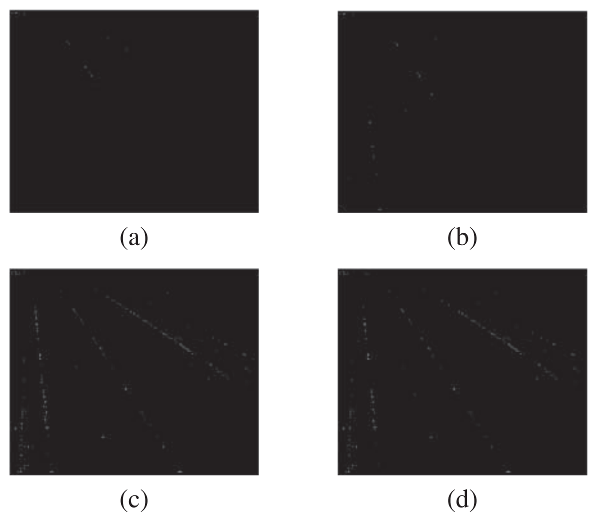


Fig. 16: Comparison of our technique with the SKIP mode decision of the reference. In the left, our background segmentation and active MBs after the spatial filter.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented an integrated RoI-based compression in the motion estimation of an state-of-the-art H.264 encoder. Our framework is tested with static cameras of traffic surveillance. Our approach with aggressive noise filtering achieves a very good performance superior to 90% of precision and recall in outdoors while the complexity is as low as

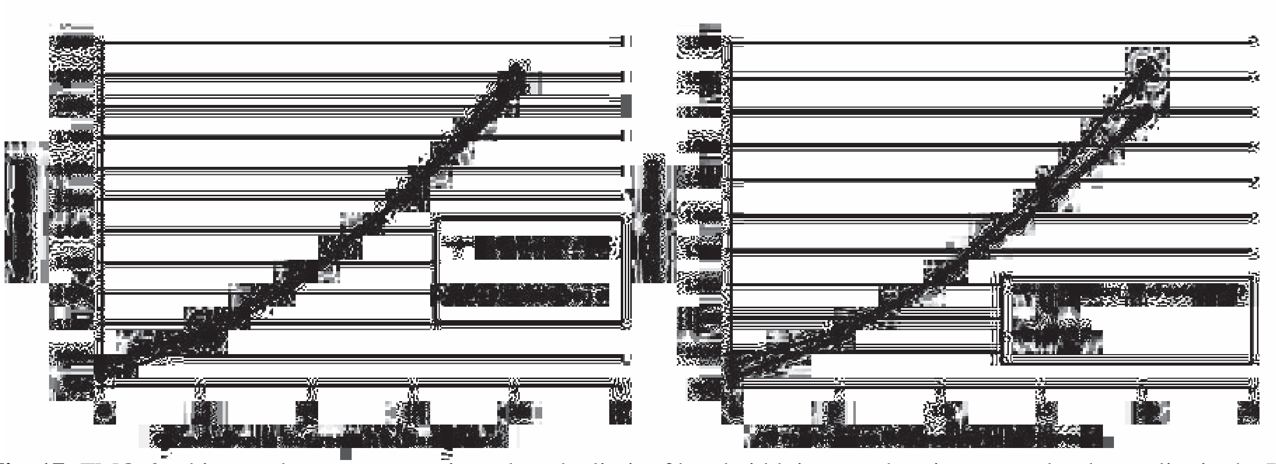


Fig. 17: FMO 6 achieves a better compression when the limit of bandwidth is enough to improve a lot the quality in the RoI.

the standard compressed methods. This low complexity features motion detection&tracking in real time.

Furthermore, FMO type 2 is used to apply the RoI compression. The bitrate cost of the headers is very low, while the compression is drastically improved in our distorted input as many MBs aren't be encoded with SKIP mode. This bitrate compression achieves 85% in the finer quantization steps. An optional xRoI is proposed improving the robustness of the motion detection while slightly worsening the global accuracy and compression. According to the latest RoIs allocations, more probability of occurrence of motion is applied in regions where the quality compression was better. Using the xRoI and the dynamic background thresholds, the two steps feedback their information improving the global performance.

FMO type 6 is also applied. With a higher signalization cost, it still provides many advantages over the compression constraints of FMO type 2. Furthermore, with a reduction of around 25% of MBs in the RoI it achieves a better compression when the bandwidth of the network overcomes a limit. A combination of both FMO types provides the best use of the network resources, achieving at every moment the best possible quality in the RoI for a certain dynamic limit of bandwidth and a standard low-quality compression in the background.

As a future work, this model can be improved in a single frame basis based on the shape of the objects. For example, if this shape is exactly rectangular FMO type 2 works better even with unlimited bandwidth. At the same time, FMO type 6 could achieve better results under particular object shapes in the low-bandwidth network region where based on the global video tests, FMO type 2 presents typically a better compression performance.

7. ACKNOWLEDGMENT

The authors would like to thank Walloon Region project Walcomo and Belgian NSF for funding part of this work. The first and second author performed part of this work while being at the Interuniversity Microelectronics Center (IMEC).

8. REFERENCES

[1] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audio-visual services," *ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, July 2005.

[2] J.Lu, G.Lafruit, and F.Catthoor, "Fast reliable multi-scale motion region detection in video processing," in *ICASSP*, April 2007, vol. 1, pp. 689–692.

[3] S.Van Leuven, K.Van Schevensteen, T.Dams, and P.Schelkens, "An implementation of multiple region-of-interest models in h.264/avc," *Multimedia Systems and Applications, Springer US*, vol. 31, pp. 215–225, 2006.

[4] F.Boulos, W.Chen, B Parrein, and P. Le Callet, "A new. h.264/avc error resilience model based on regions of interest," in *17th Int. Packet Video Workshop*, May 2009.

[5] Y.Dhondt, P.Lambert, S.Notebaert, and R.Van De Walle, "Flexible macroblock ordering as a content adaptation tool in h.264/avc," in *SPIE*, May 2005, pp. 44–52.

[6] W. You, M. S. H. Sabirina, and M. Kima, "Real-time detection and tracking of multiple objects with partial decoding in h.264/avc bitstream domain," in *SPIE*, Feb. 2009, vol. 7244.

[7] C. DeVleeschouwer, T. Nilsson, K. Denolf, and J. Bormans, "Algorithmic and architectural co-design of a motion-estimation engine for low power video devices," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, Dec. 2002.

[8] P. Lambert, W. De Neve, Y. Dhondt, and R. Van deWalle, "Flexible macroblock ordering in h.264/avc," *Elsevier Journal of Visual Communication and Image Representation*, vol. 17(2), pp. 358–375, April 2006.

[9] T.M. Bae, T. C.Thang, D.Y.Kim, Y.M.Ro, J.W.Kang, and J.G.Kim, "Multiple region-of-interest support in scalable video coding," in *ETRI Journal*, April 2006, vol. 28, pp. 239–242.

[10] W.Wang, J.Yang, and W.Gao, "Modeling background and segmenting moving objects from compressed video," *IEEE transactions on circuits and systems for video technology*, vol. 18, pp. 670–681, May 2008.

[11] P.Baccichet, X.Zhu, and B.Girod, "Network-aware h.264/avc region-of-interest coding for a multi-camera wireless surveillance network," in *Picture Coding Symp.*, April 2006.