ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

See Without Decoding: Motion-Vector-Based Tracking in Compressed Video

Duché Axel[1,2], Chatelain Clément [2] and Gasso Gilles[2] *

1- Actemium Paris Transport, Nanterre - France
2- INSA Rouen, LITIS UR 4108, Saint-Étienne-du-Rouvray - France

Abstract. We propose a lightweight **compressed-domain tracking model** that operates directly on video streams, without requiring full RGB video decoding. Using motion vectors and transform coefficients from compressed data, our deep model propagates object bounding boxes across frames, achieving a computational speed-up of order up to 3.7× with only a slight 4% mAP@0.5 drop vs RGB baseline on MOTS15/17/20 datasets. These results highlight codec-domain motion modeling efficiency for real-time analytics in large monitoring systems. Our implementation is publicly available at : `https://github.com/Fr4cti0n/See_without_decoding`

1 Introduction

Modern cities operate extensive camera networks across public areas and sensitive zones. These systems must deliver reliable, continuous analytics—including motion detection, intrusion monitoring, and behavior analysis—under strict constraints on computation, storage, and energy consumption across thousands of concurrent video streams. Conventional image-domain preprocessing [1] (e.g., background subtraction or frame differencing) offers low-cost activity filtering but remains fragile under illumination or perturbations. Deep learning–based vision models have substantially improved robustness and precision while maintaining a good balance between speed and accuracy, as seen in recent architectures such as RT-DETR [2] and the latest YOLOs [3]. However, most vision pipelines and methods still rely on fully decoded RGB frames, which are computationally heavy to process. Performing real-time inference on high-resolution RGB data requires powerful GPUs or specialized hardware, making large-scale deployment difficult. The reliance on extensive RGB processing poses a significant scalability challenge for camera networks. This paper adress this limitation by starting from a simple hypothesis: Compressed video streams already carry most of the spatial–temporal information required for tracking. Based on this, this work proposes a lightweight hybrid architecture that performs a single detection on an initial decoded RGB frame, followed by tracking and refinement of bounding boxes directly from codec-domain features. Hence, the method avoids redundant pixel-level computation while maintaining competitive accuracy and supporting scalable, energy-efficient analytics across large camera infrastructures. Hereafter, we first explain the compressed video representation that enables our approach and review prior work on compressed-domain video understanding and how these methods balance efficiency and accuracy under similar constraints.

*This research is supported by ANRT and ACTEMIUM Paris Transport. We thank ACTEMIUM Paris Transport for the funding and CRIANN for the computation facilities

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

2 Compressed Video and Related Work

2.1 Compressed Video Representation

Standard video codecs (e.g., MPEG-4, H.264) organize streams into *Groups of Pictures* (GOPs) $\mathcal{G}^g = \{f_0^g, f_1^g, \ldots, f_N^g\}$, where f_0^g is a fully encoded RGB image, termed I-frame, and temporal frames f_n^g $(n \geq 1)$, also known as P-frames, are encoded via block motion estimation. Each P-frame contains two components: $f_n^g = \{MV_n^g, \Delta(Y, Cb, Cr)_n^g\}$, Motion vectors MV_n^g that capture block-wise displacement by seeking matching blocks within a search zone, while residuals $\Delta(Y, Cb, Cr)_n^g$ encode appearance differences via pixel subtraction. Since MPEG-4 applies Discrete Cosine Transform (DCT) to compress residuals, one has access to directly work with DCT coefficients that compactly represent texture, edges, and appearance changes in the frequency domain.

These "codec features" are already computed during encoding, making them extractable from the bitstream at minimal cost—a key advantage for large-scale video analytics. Motion vectors are dramatically more compact than RGB images because they encode the displacement of a 16×16 pixel block using only two floating-point values. DCT residuals, on the other hand, remain considerably larger, yet still far lighter than full RGB data, providing efficient motion and appearance cues at negligible extraction cost.

2.2 Related Work and Positioning

In this work, we exploit codec features to design a fast tracking models with minimal decoding. Previous studies have explored different levels of reconstruction before prediction.

Extensive decoding: Conventional approaches fully reconstruct RGB frames and increase the inputs with compressed features. Methods such as MV-YOLO [4], originally designed for city surveillance, and later RGB–motion fusion variants like MV-Soccer [12] and ReST [13], for sports tracking, all exploit codec motion vectors alongside RGB cues to enhance temporal consistency. Although this design effectively leverages motion information, it still depends on full-resolution RGB decoding and deep feature extractors, resulting in architectures that are computationally demanding.

Partial decoding: To reduce overhead, a few methods leverage motion and residual information computed by the codec while still decoding parts of the frame. Frame-group aggregation methods [5, 6] process an entire GOP in a single inference using the initial decoded RGB I-frame and decompressed residuals and motion vectors to predict the objects in all the frames of the GOP.Despite the use of smaller data, this architecture is still computationally intensive and offers only limited improvements in terms of runtime efficiency. Other hybrid designs [7] decode small regions or key patches while combining them with motion cues from the bitstream. However, because video data are sequentially encoded, even partial decoding typically requires decoding earlier related patches, which limits the achievable efficiency.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

No-decoding methods: Recent works such as [8] and [16] operate directly on compressed-domain, bypassing the need for RGB reconstruction. Video-domain approaches are reliant on codec features alone. One example is the naïve Mean-MV baseline in [7], where boxes obtained from an RGB image are tracked using the average of the motion vectors in each box. There's also the MV models [15] and DCT-based methods [14]. Although highly efficient, they struggle with static or slow-moving objects due to a lack of motion information. Nevertheless, no-decoding strategies remain attractive for scalable and efficient video analytics.

Building on the latter idea, we propose a hybrid lightweight tracking scheme that bridges partial decoding and no-decoding, where only the I-frame is decoded, while therest of the GOP is being handled entirely in the compressed domain. This eliminates the need for repeated reconstruction while ensuring that the cues needed for accurate, scalable video analytics are maintained.

3 See Without Decoding using BAFE

Our approach obtains initial I-frame detections from an RT-DETRv2m [2] model operating in the RGB domain, and then applies a fully compressed-domain tracking system for the P frames to propagate these detections together with the motion vectors. This results in a lightweight hybrid architecture. As shown in Figure 1, once the initial boxes are obtained, each P-frame f_n^g provides its motion vectors MV_n^g, DCT residuals ΔY_n^g, and the previous boxes $bb(n-1)$ at the *Inputs* stage (left). To replace naïve motion-only propagation such as Mean-MV, we introduce a *Box-Aligned Feature Extraction (BAFE)* module, which gathers motion and appearance cues from both the box and its surrounding neighborhood. In the *BAFE Extraction* block (center), motion vectors and DCT residuals are transformed into compact MV and DCT A-Features, while previous boxes are encoded through a lightweight box embedding. These codec features are then fed into the *Fusion & Temporal* module (right), where a BiLSTM predicts refinement terms that the *ABoxes* component transforms into boxes $bb(n)$ by updating previous boxes $bb(n-1)$.

Addressing SOTA Limitations. Our design directly tackles limitations previously observed. (1) *vs. partial- and full-decoding methods* [5, 6, 4, 12]: our approach avoids any form of repeated or selective RGB reconstruction, eliminating both the heavy computational cost associated with full decoding and the synchronization overhead inherent to partial decoding; (2) *vs. no-decoding methods* [14, 8]: by performing a single RGB-based detection on the I-frame f_0^g, we obtain high-quality initial boxes that allow us to track static and slow-moving objects, which motion-only approaches typically fail to capture. The strong I-frame priors convert a difficult detection problem into a lightweight propagation task for which codec features are sufficiently informative. To reliably exploit these compressed-domain cues, we further introduce the BAFE module, which provides a learnable alternative to naïve MV-based propagation by extracting richer motion and appearance information from both the interior of each bounding box and its immediate neighborhood. We use Y-channel residuals only, cutting data by 3× while maintaining tracking information.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

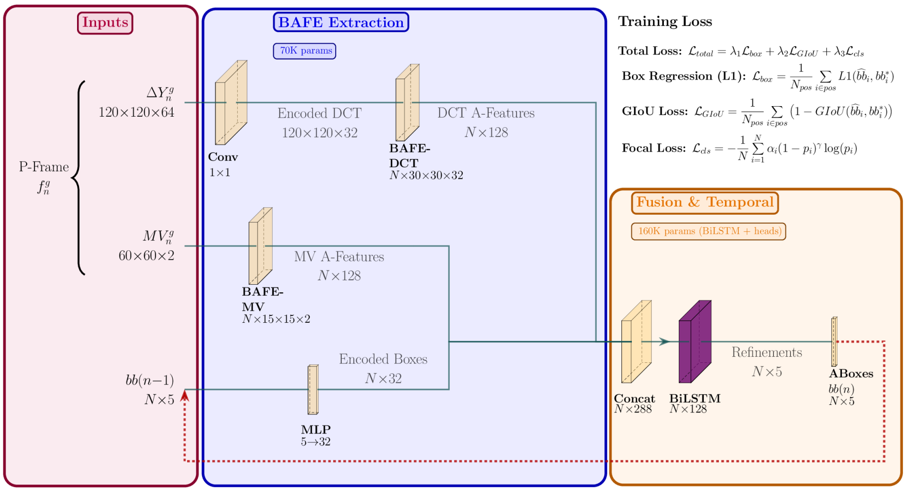

Figure 1: BAFE architecture: box-aligned MV/DCT features are fused with box encodings and then fed to BiLSTM for temporal modeling.

BAFE (Box-Aligned Feature Extraction). The BAFE module extracts features *spatially aligned* with each bounding box, ensuring that only object-relevant regions are processed. For every box b_n^i in frame f_n^g, motion vectors MV_n^g and DCT residuals ΔY_n^g are sampled on a fixed-size grid. Because block motion estimation in *MPEG-4 Part 2* operate on fixed block size. BAFE selects a compact grid covering both the box interior and its immediate neighborhood, capturing the most informative motion and appearance cues. The chosen grid sizes (Figure 1) balance detail and efficiency, producing lightweight MV and DCT A-Features while retaining essential spatiotemporal structure. By focusing computation on box-aligned regions instead of full-frame tensors, BAFE provides an expressive yet efficient representation well suited for real-time tracking.

BiLSTM Temporal Propagation. To model temporal dependencies across P-frames, we employ a bidirectional LSTM that takes as input the concatenation of: (1) box-aligned motion vector features from MV_n^g, (2) box-aligned DCT coefficient features from ΔY_n^g, and (3) encoded bounding box geometry (position/size of b_{n-1}^i). The BiLSTM hidden state captures motion patterns over time, enabling the model to predict refined bounding box updates $\Delta b_n^i = (\Delta x, \Delta y, \Delta w, \Delta h)$ relative to the previous frame. This temporal modeling is crucial for handling occlusions, camera motion, and non-linear trajectories that simple motion vector averaging cannot capture.

ABoxes and Training Loss. The ABoxes module applies the predicted refinements to the previous frame's bounding boxes by summing the predictions with the previous coordinates. Training relies on a weighted combination of three standard detection objectives: an L1 loss ($\mathcal{L}_{\text{box}}$) for bounding box regression, a GIoU loss ($\mathcal{L}_{\text{GIoU}}$) to improve geometric alignment, and a focal loss ($\mathcal{L}_{\text{cls}}$)

to manage potential disappearances. The total loss is computed as $\mathcal{L}_{\text{total}} = \lambda_1\mathcal{L}_{\text{box}}+\lambda_2\mathcal{L}_{\text{GIoU}}+\lambda_3\mathcal{L}_{\text{cls}}$, with weights $\lambda_1 = 5$, $\lambda_2 = 2$, and $\lambda_3 = 2$, following the original RT-DETRv2 implementation for fair comparison. The box regression and GIoU losses on boxes by comparing predicted boxes $\widehat{bb}_i$ with ground truth bb_i^*. The focal loss is computed over all $N = 200$ proposals using class balance weights α_i, predicted probabilities p_i, and focusing parameter $\gamma = 2$ to down-weight easy examples and focus training on hard negatives.

4 Experiments and results

We train and evaluate on MOT15 [9]/17 [10]/20 [11] with RGB-based baselines RT-DETRv2 and the lighter YOLOv8s, as well as the **Mean-MV** baseline used in [7]. The MOT datasets focus on pedestrian tracking from surveillance and on-board cameras, but we restrict our study to static-camera sequences, as our target application is surveillance. The videos used for training and evaluation follow the official train/test splits of the MOT datasets. All trainable models are trained on 200 GOPs of size 6 for 100 epochs. Training follows the inference pipeline: detections are obtained on the I-frame to initialize tracking, after which the LSTM-based BAFE model is trained sequentially to propagate predictions across the P-frame sequence.

We report mAP@0.5 since the tracker runs without IDs and aims to minimize detector usage. As shown in Table 1, our best BAFE model (MV+DCT) reaches **0.8962 mAP** on MOT17, improving over Mean MV by **+2.86%** while remaining competitive with both RT-DETRv2 and YOLOv8s. Notably, MOT17 contains fewer occlusion-heavy scenes than MOT15 and MOT20, which partly explains why the simple Mean-MV baseline achieves relatively strong performance on this dataset. On MOT15 and MOT20, our approach yields gains of **over 20%** by better capturing fast pedestrian motion and complex dynamics.

Table 1: Evaluations of BAFE models on static cameras (GOP-6, mAP@0.5).

Model	MOT17	MOT15	MOT20	FPS	FLOPs	Streams	Params
RT-DETRv2m	**0.9187**	**0.8234**	**0.8153**	296	92G	13	31.45M
YOLOv8s	0.8856	0.8023	0.7946	450	28.6G	21	11.2M
Mean MV	0.8676	0.5879	0.6590	**2600**	–	**62**	**0**
BAFE (MV)	0.8756	0.7851	0.7990	2350	**2.1G**	48	160K
BAFE (DCT)	0.8543	0.7564	0.7843	2250	2.4G	45	202K
BAFE (MV+DCT)	**0.8962**	**0.7958**	**0.8020**	2200	2.6G	42	230K

Results and Discussion Table 1 summarizes both efficiency and deployment capacity on a 16GB GPU for GOP-6 (1 I-frame + 5 P-frames). Despite having only 160K parameters and requiring just 2.1G FLOPs, our MV-only variant sustains **2350 FPS** and supports **48 concurrent streams**, compared to **13 streams** for RT-DETRv2m (31M params, 92G FLOPs). Even with a smaller RGB detector such as YOLOv8s (11M params, 28.6G FLOPs), throughput remains limited to **21 streams**, highlighting the scalability advantage of compressed-domain propagation. The naïve Mean-MV baseline reaches 62 streams but suffers a $\sim$20% mAP drop relative to our learned model. Our BAFE-based models consistently outperform the naïve Mean-MV baseline, validating learned box-aligned compressed features with temporal modeling. The MV-only variant

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

achieves strong accuracy at minimal cost, demonstrating the efficiency of motion-vector tracking. We fix the GOP size to 6 to balance accuracy and propagation length, as larger GOPs would require heavier temporal models and further limit handling of new object appearances. Table 1 shows that end-to-end scalability is best captured by concurrent streams, while the large FPS gap between propagation (2200+) and RT-DETRv2 (296) highlights the detector bottleneck despite being applied only once per GOP, confirming propagation as a solution for real-time surveillance.

5 Conclusion and Future Work

We presented a hybrid compressed-domain tracking framework that performs a single RGB initialization before relying on codec-domain features. With the BAFE module and a lightweight temporal head, our method achieves strong accuracy and high scalability, highlighting the benefit of exploiting compressed information directly. Future work includes extending BAFE to multi-scale box-aligned grids for improved robustness to scale variations and strengthening the BiLSTM to handle larger GOPs without performance loss.

References

[1] D. Lohani, C. Crispim-Junior, Q. Barthélemy, S. Bertrand, L. Robinault, and L. Tougne, "Perimeter intrusion detection by video surveillance: A survey," *Sensors*, vol. 22, 2022.

[2] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs beat YOLOs on real-time object detection," in *Proc. CVPR*, 2024.

[3] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," arXiv preprint, 2024.

[4] S. R. Alvar and I. V. Bajić, "MV-YOLO: Motion vector-aided tracking by semantic object detection," in *Proc. MMSP*, 2018.

[5] S. Wang, H. Lu, and Z. Deng, "Fast object detection in compressed video, *ICCV*, 2019.

[6] X. Wang, Z. Huang, B. Liao, L. Huang, Y. Gong, and C. Huang, "Real-time and accurate object detection in compressed video by long short-term feature aggregation," *CVIU*,2021

[7] R. Tran, A. Kanaujia, and V. Parameswaran, "Fast object detection in high-resolution videos," in . *ICCV Workshops*, 2023.

[8] B. Deguerre, C. Chatelain, and G. Gasso, "Object detection in the DCT domain: Is luminance the solution?," in *Proc. ICPR*, 2020.

[9] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," in *Proc. CVPR*, 2015.

[10] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," arXiv preprint, 2016.

[11] P. Dendorfer *et al.*, "MOT20: A benchmark for multi-object tracking in crowded scenes," arXiv preprint, 2020.

[12] F. Majeed *et al.*, "MV-Soccer: Motion-vector augmented instance segmentation for soccer player tracking," in *Proc. CVPR Workshops*, 2024.

[13] F. Majeed *et al.*, "ReST: High-precision soccer player tracking via motion vector segmentation," in *Proc. VISAPP*, 2025.

[14] K. El Khoury, J. Samelson, and B. Macq, "Deep learning-based object tracking via compressed-domain residual frames," *Frontiers in Signal Processing*, vol. 1, 2021.

[15] B. Deguerre, C. Chatelain, and G. Gasso, "End-to-end traffic flow rate estimation from MPEG-4 part-2 compressed video streams," *IEEE Trans. Intell. Transp. Syst.*, 2024.

[16] L. Gueguen *et al.*, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.