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

Duché Axel[1,2], Gasso Gilles[2] and Chatelain Clément [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 7× with only a slight 4% mAP@0.5 drop vs RGB baseline on MOTS15/17/20. These results highlight codec-domain motion modeling efficiency for real-time analytics in large monitoring systems.

1 Introduction

Modern cities operate extensive camera networks across transportation hubs, 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 changes, vibrations and clutter. 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 YOLO versions [3]. However, most methods still rely on fully decoded RGB frames, which are computationally heavy and memory-intensive to process. Performing real-time inference on high-resolution RGB data requires powerful GPUs or specialized hardware, making large-scale deployment difficult and energy-intensive. This dependency on heavy RGB processing represents a fundamental scalability barrier for camera networks.

The paper addresses this limitation by starting from the following assumption: compressed video streams already carry most of the spatial–temporal information required for tracking. In particular, motion vectors and transform coefficients estimated by the codec can act as meaningful motion and appearance cues without explicit RGB reconstruction. Building on this, the paper 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. By leveraging information already available in the stream, the method avoids redundant pixel-level computation

*This work was supported by the ANRT under a CIFRE contract with Actemium Paris Transport.

while maintaining competitive accuracy, supporting scalable and energy-efficient analytics across large camera infrastructures.

Hereafter, we first explain the compressed video representation that enables our approach, then review prior work on compressed-domain video understanding and how these methods balance efficiency and accuracy under similar constraints.

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 and temporal frames f_n^g $(n > 0)$ are encoded via block motion estimation (Fig. 1).

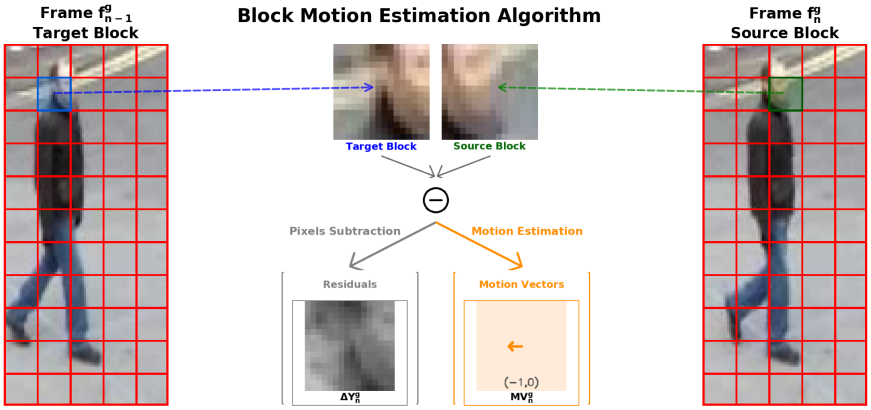

Figure 1: Block motion estimation in MPEG-4: the encoder searches for matching blocks between consecutive frames, computing motion vectors MV_n^g and residuals ΔY_n^g (pixel subtraction). The search zone shows candidate blocks explored during estimation.

Each P-frame contains two key components:

$$f_n^g = \{MV_n^g, \Delta Y_n^g\}, \qquad f_0^g = \{\mathcal{DCT}(Y_0^g)\}.$$

Motion vectors MV_n^g capture block-wise displacement by searching for matching blocks within a search zone, while residuals ΔY_n^g encode appearance differences via pixel subtraction. Since MPEG-4 applies Discrete Cosine Transform (DCT) to compress residuals, we directly work with DCT coefficients that compactly represent texture, edges, and appearance changes in the frequency domain. To reduce computational cost, we operate exclusively on the luminance (Y) channel—equivalent to a grayscale image—cutting data volume by $\sim$3× while preserving spatial-temporal structure for tracking.

Efficiency. 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** (384× smaller), providing efficient motion cues at negligible memory cost. DCT residuals, while having the same spatial resolution as grayscale images (80×80 for 8×8 blocks), can be further compressed by the energy-compaction properties of the DCT transform—similar to JPEG compression—concentrating information in low-frequency coefficients and enabling potential data reduction as exploited in standard compression pipelines.

2.2 Related Work and Positioning

Codec-domain features already computed during video encoding provide compact motion and appearance cues at minimal cost. However, existing methods differ in how much of the original RGB data they reconstruct before inference. They can be broadly categorized into three decoding regimes: *full*, *partial*, and *none*. Full-decoding approaches restore complete RGB frames for analysis, achieving strong accuracy but at high computational expense. Partial-decoding methods limit reconstruction to selected regions or frames, reducing cost while retaining spatial detail. Finally, no-decoding models work directly on compressed-domain data such as motion vectors or transform coefficients, maximizing efficiency for large-scale video analytics.

Full decoding. Conventional approaches fully reconstruct RGB frames before analysis. They rely on convolutional or transformer-based deep models to detect and track objects in the pixel domain, achieving high accuracy but at a substantial computational cost. Methods such as MV-YOLO [4], originally designed for city surveillance, and later RGB–motion fusion variants like MV-Soccer [13] and ReST [14], 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 accurate but computationally demanding and difficult to scale for large, real-time video analytics.

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 one GOP at a time, using the initial decoded RGB I-frame together with decompressed residuals and motion vectors to predict objects across subsequent frames. While this design effectively exploits temporal redundancy and achieves strong accuracy, many of its internal operations—such as feature resizing and residual decompression—are not computationally optimal, limiting the overall speed and 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.

No-decoding methods. Recent studies [15, 16, 8, 9] explore learning directly from compressed-domain data without reconstructing RGB pixels. Among them, [15] and [8] focus on object detection in still images compressed with HEVC-Intra and JPEG respectively, showing that partitioning depths, prediction modes, and transform coefficients preserve sufficient spatial structure for localization. However, these image-oriented designs cannot be directly extended to video, where temporal dependencies and motion cues play a critical role beyond static spatial transformations. In contrast, video-oriented methods [16, 9] rely exclusively on motion vectors and residuals extracted from compressed streams. While efficient, these architectures struggle to handle static or slow-moving objects, since such regions generate little or no motion information in the bitstream, leading to incomplete scene representation. Nevertheless, this last class of models remains the most suitable for minimizing computational cost and energy consumption, making it a promising direction for scalable, power-efficient video analytics.

Our contribution Building on the latter idea, we propose a lightweight *tracking model* that performs a single detection on an initial decoded frame and then updates bounding boxes using only motion vectors and frequency domain features from the compressed stream. Unlike full or partial decoding-based approaches, our formulation avoids repeated RGB reconstruction and heavy temporal aggregation, operating entirely on data already present in the bitstream. This design retains essential motion and appearance cues while drastically reducing computational burden, making it suitable for real-time, large-scale video analytics.

3 See Without Decoding using BAFE

Our approach initializes bounding boxes via RGB-based detection on I-frame f_0^g, then propagates them across P-frames using only codec features (motion vectors, DCT coefficients) extracted directly from the bitstream without full RGB decoding.

BAFE (Box-Aligned Feature Extraction). Unlike global pooling approaches that process entire frames, our architecture extracts features *spatially aligned* with each bounding box. For each box b_n^i at frame f_n^g, we extract fixed-size grid features from the bounding box region in motion vectors MV_n^g and DCT coefficients ΔY_n^g. For videos of size 960×960 pixels with mean bounding box resolution of 240×240, we use a 15×15 grid for motion vectors (matching the spatial resolution of 16-pixel macroblocks). For DCT coefficients, where each 8×8 pixel block yields 64 frequency channels, we employ a 30×30 grid to capture finer-grained appearance details. This box-aligned extraction focuses computation on object regions while ignoring background, enabling lightweight processing ($\sim$230K params) compared to full-frame architectures. Each extracted region is processed through shallow convolutional layers (2 Conv blocks, 64 channels) to produce compact 128-dimensional feature embeddings.

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.

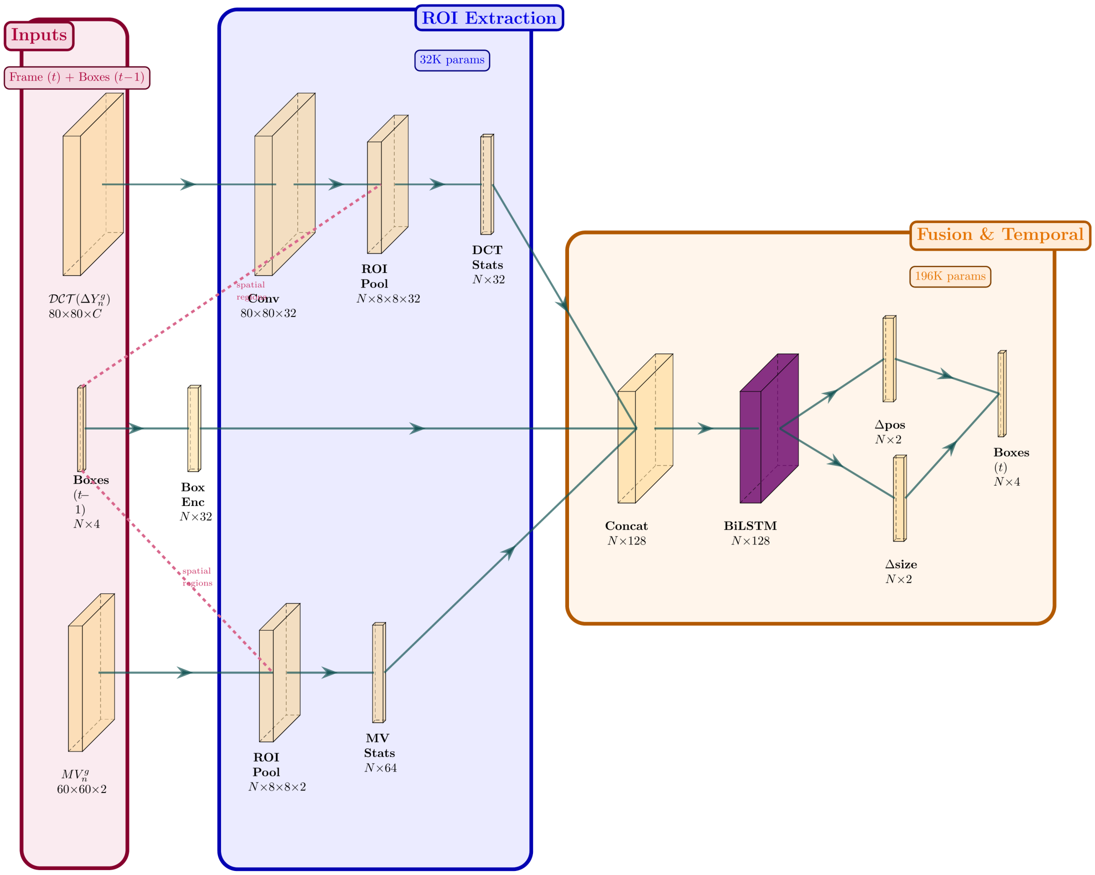

Figure 2: BAFE architecture: box-aligned MV/DCT features fused with box encodings, BiLSTM for temporal modeling.

Addressing SOTA Limitations. Our hybrid approach resolves three key problems in compressed-domain tracking: (1) *vs. no-decoding methods* [16, 9]: by using RGB-based detection on f_0^g, we obtain high-quality initial boxes that enable tracking of static or slow-moving objects (which purely motion-based methods miss); (2) *vs. full-decoding methods* [4, 13]: after the initial I-frame, we never reconstruct RGB images, avoiding computational overhead of repeated decoding; (3) *vs. partial-decoding methods* [5, 6]: we operate directly on bitstream features without selective decoding, eliminating synchronization overhead and codec-specific optimizations. The initial bounding boxes serve as strong priors that transform the challenging detection problem into a simpler tracking problem, where codec features suffice for propagation.

4 Experimental Results

We evaluate on MOT15/17/20 static cameras with baselines: **RT-DETR** (RGB-based detection per frame) and **Mean MV** (average motion vector per box). The MOT datasets focus on pedestrian tracking from surveillance camera recordings. All model variants are trained using DETR-style bounding box regression losses (focal loss for center prediction, L1 and GIoU for box geometry) on 200 GOPs of size 6. Since the tracking task involves a single pedestrian class, no classification head or cross-entropy loss is used—the model directly predicts box refinements Δb_n^i from codec features. Table 1 presents GOP-6 performance (1I+5P), where our best BAFE model (MV+DCT) achieves **0.8962 mAP** on MOT17, outperforming Mean MV by **+2.86%** while staying within **2.25%** of the RGB baseline. On **MOT15**, the model gains **+20.79%** over Mean MV by learning temporal patterns for fast-moving pedestrians. The MV-only variant maintains strong performance (**0.8756 mAP** on MOT17), demonstrating that motion cues alone suffice for accurate tracking when combined with learned temporal modeling.

Table 1: BAFE model performance on static cameras (GOP-6, mAP@0.5).

Metric	**MOT17**	**MOT15**	**MOT20**
Baseline RGB (RT-DETR)	**0.9187**	**0.8234**	**0.8153**
Mean MV	0.8676	0.5879	0.6590
BAFE Model (MV)	**0.8756**	**0.7851**	**0.7990**
BAFE Model (DCT)	0.8543	0.7564	0.7843
BAFE Model (MV+DCT)	**0.8962**	**0.7958**	**0.8020**
Our Model (Best) vs Mean MV	**+0.0286**	**+0.2079**	**+0.1430**
Our Model (Best) vs RT-DETR	-0.0225	-0.0276	-0.0133
Our Model (MV) vs RT-DETR	-0.0431	-0.0383	-0.0163

Deployment Scalability. Table 2 shows deployment capacity on 16GB GPU for GOP-6 (1I+5P). Despite its compact 160K parameters, our MV-only variant processes **71 concurrent streams** vs. **13 streams** for RT-DETR (32M params), achieving **5.5× throughput** with 200× fewer parameters. The lightweight Mean MV baseline (no learned parameters) handles 85 streams but sacrifices 20% mAP compared to our learned model. MV+DCT and DCT variants (230K and

202K params) support 21-23 streams while maintaining competitive accuracy, demonstrating flexible accuracy-throughput trade-offs for different deployment scenarios.

Table 2: Deployment capacity on 16GB GPU (30 FPS streams) for GOP-6 (1I+5P).

Variant	**Concurrent Streams**	**Parameters**
RT-DETR (RGB)	13	32M
Mean MV	85	0
BAFE Model (MV)	**71**	**160K**
BAFE Model (MV+DCT)	21	230K
BAFE Model (DCT)	23	202K

Discussion. Our learned BAFE models significantly outperform the naive Mean MV baseline (+2.9% to +20.8% across datasets), demonstrating that temporal modeling via BiLSTM effectively captures motion patterns beyond simple vector averaging. Notably, the MV-only variant achieves competitive accuracy (0.8756 mAP on MOT17) with only a 2.1% gap relative to MV+DCT (0.8962 mAP), suggesting motion vectors alone carry sufficient information for tracking when properly exploited through learned features. This validates the potential of codec-domain motion as a primary cue for video analytics.

Speed analysis reveals that throughput differences stem primarily from input data size rather than model complexity: MV-only (71 streams, 160K params) outperforms MV+DCT (21 streams, 230K params) by 3.4× despite similar architectures, highlighting the compactness advantage of motion vectors (384× smaller than images). Comparing our variants (21-71 streams, 160-230K params) against RT-DETR (13 streams, 32M params) confirms that efficient feature extraction at codec level—not just model size—drives scalability.

Future work could explore multi-scale bounding box grids inspired by YOLO-style feature pyramids to better handle scale variations, and selective DCT frequency exploitation (leveraging energy-compaction properties) to improve appearance modeling without sacrificing the current throughput advantages. Additionally, incorporating explicit object identity tracking (e.g., learnable ID embeddings) could further improve temporal consistency, as the current model operates purely on geometric and codec features without identity-aware associations across frames.

References

[1] D. Lohani, C. Crispim-Junior, Q. Barthélemy, S. Bertrand, L. Robinault and L. Tougne Rodet, Perimeter Intrusion Detection by Video Surveillance: A Survey, MDPI, 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, CVPR, 2024

[3] R. Khanam and M. Hussain, YOLOv11: An Overview of the Key Architectural Enhancements, ARXiV, 2024

[4] S. R. Alvar and I. V. Bajić, MV-YOLO: Motion Vector-aided Tracking by Semantic Object Detection, IEEE International Workshop on Multimedia Signal Processing, 2018.

[5] S. Wang, H. Lu, and Z. Deng, "Fast Object Detection in Compressed Video, ICCV, 2018

[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, ICCVW, October 2023.

[8] B. Deguerre, C. Chatelain and G. Gasso, Fast object detection in compressed JPEG images, ITSC , 2019.

[9] B. Deguerre, C. Chatelain and G. Gasso, Object Detection in the DCT Domain: is Luminance the Solution, ICPR 2020

[10] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking

[11] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking

[12] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A Benchmark for Multi-Object Tracking in Crowded Scenes

[13] F. Majeed, N. U. Gilal, K. Al-Thelaya, Y. Yang, M. Agus, and J. Schneider, "MV-Soccer: Motion-Vector Augmented Instance Segmentation for Soccer Player Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2024, pp. 3245–3255, doi:10.1109/CVPRW63382.2024.00330.

[14] F. Majeed, K. A. L. Al-Thelaya, N. U. Gilal, K. Swart-Arries, M. Agus, and J. Schneider, "ReST: High-Precision Soccer Player Tracking via Motion Vector Segmentation," in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 2, 2025, pp. 138–149, doi:10.5220/0013168000003912.

[15] L. Chen, Y. Li, C. Hou, and Z. Gao, "Fast Object Detection in HEVC Intra Compressed Domain," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland, Aug. 2021, pp. 756–760.

[16] K. El Khoury, J. Samelson, and B. Macq, "Deep Learning-Based Object Tracking via Compressed Domain Residual Frames," *Frontiers in Signal Processing*, vol. 1, Nov. 2021, article 765006. doi:10.3389/frsip.2021.765006.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2961–2969.