JLISNT Journal of Leading Innovations in Science and Technology
Journal of Leading Innovations in Science and Technology, Vol.1, No.1, pp: 24-40, Jan 2026

AN ADAPTIVE VLSI-BASED OBJECT TRACKING ARCHITECTURE USING LARGEST DIFFERENCE INDEXED LOCAL TERNARY PATTERNS IN THE COMPRESSED DOMAIN

Yasser Rahim[1], Dr. Vijeta Yadav [2], Dr. K.Venkata Murali Mohan [3]

[1]*Research Scholar, Madhyanchal Professional University (MPU), India.* yasserrahim1146@gmail.com,
[2]*Supervisor, Madhyanchal Professional University (MPU), India.* vijeta.cool17@gmail.com
[3]*Professor of ECE and Principal, Teegala Krishna Reddy, Engineering College, Medbowli, Meerpet(M), Ranga Reddy District, Hyderabad, Telangana-500097,* kvmmece@gmail.com

Abstract – Object tracking in dynamic and resource-constrained environments remains a critical challenge in modern computer vision systems, particularly for real-time applications such as surveillance, autonomous navigation, and smart imaging devices. This paper presents an adaptive VLSI-based object tracking architecture that leverages Largest Difference Indexed Local Ternary Patterns (LDI-LTP) in the compressed domain to achieve high efficiency and robustness. The proposed method operates directly on compressed video streams, significantly reducing computational overhead and memory requirements by avoiding full decompression.

The core feature extraction mechanism utilizes LDI-LTP, which enhances traditional Local Binary Patterns (LBP) by incorporating a ternary encoding scheme and indexing the largest intensity differences within local neighborhoods. This approach improves texture discrimination and robustness against noise, illumination variations, and compression artifacts. By selecting the most significant pixel differences, the descriptor reduces redundancy while maintaining discriminative power, making it well-suited for hardware implementation. The architecture is designed with a parallel and pipelined processing framework optimized for VLSI realization, ensuring low latency and high throughput. It integrates modules for compressed-domain feature extraction, adaptive model updating, and efficient matching for object localization. The adaptive mechanism dynamically adjusts tracking parameters based on scene variations, enabling consistent performance in complex environments involving occlusion, scale changes, and motion blur. Experimental evaluations demonstrate that the proposed system achieves superior tracking accuracy and energy efficiency compared to conventional spatial-domain methods and existing hardware implementations. The design also shows significant improvements in area utilization and power consumption, making it suitable for embedded and edge computing platforms.

Overall, the proposed adaptive VLSI architecture with LDI-LTP in the compressed domain provides a scalable and efficient solution for real-time object tracking, bridging the gap between advanced computer vision algorithms and practical hardware deployment.

Keywords: *Object Tracking, VLSI Architecture, Largest Difference Indexed Local Ternary Patterns (LDI-LTP), Compressed Domain Processing, Real-Time Systems, Feature Extraction, Hardware Acceleration, Embedded Vision, Low Power Design, Pattern Recognition.*

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com

Accepted Jan.2026,

I. INTRODUCTION

Object detection and tracking are fundamental components of intelligent vision systems, particularly in real-time and embedded applications where accurate perception of dynamic scenes is essential. In adaptive Very Large-Scale Integration (VLSI)–based vision architectures, object tracking involves the continuous estimation of an object's spatial and temporal attributes—such as position, motion trajectory, and structural appearance—across successive video frames. This capability is central to a wide range of applications including automated surveillance systems, autonomous vehicle navigation, smart monitoring infrastructures, and advanced human–computer interaction frameworks

A critical stage in object tracking is feature extraction, wherein discriminative visual characteristics are derived to represent objects reliably over time. Conventional tracking approaches typically rely on features such as color, edges, contours, textures, and shape descriptors. However, in dynamic environments characterized by illumination variations, background motion, sensor noise, and partial occlusions, many traditional descriptors exhibit degraded performance. To address these limitations, Local Ternary Pattern (LTP)–based descriptors have gained prominence due to their enhanced robustness against noise and their ability to encode both micro- level texture details and broader structural patterns within local neighborhoods. Unlike binary pattern operators, LTP introduces a three-level quantization scheme that stabilizes feature representation under moderate intensity fluctuations, making it well suited for real-time tracking applications

Despite these advantages, standard LTP descriptors may still suffer from sensitivity to rotational changes and contrast inconsistencies in highly dynamic scenes. To overcome such challenges, an enhanced feature representation known as the Largest Difference Indexed Local Ternary Pattern (LDILTP) has been introduced. The LDILTP operator emphasizes dominant local intensity variations by indexing the most significant pixel differences within a neighborhood, thereby improving rotational invariance and contrast sensitivity. This refinement enables more stable feature extraction across consecutive frames, which is particularly beneficial for object detection and tracking within compressive sensing and hardware-constrained vision frameworks.

Building upon these advancements, the tracking framework proposed in this chapter integrates LDILTP within a structured, hardware-oriented processing pipeline optimized for adaptive VLSI implementation. As illustrated in this paper , the system comprises four principal stages: (i) Discrete Cosine Transform (DCT)–based image compression to reduce data redundancy and memory bandwidth requirements, (ii) LDILTP-based robust feature extraction for reliable object representation, (iii) morphological gap-filling operations to refine object boundaries and improve spatial continuity, and (iv) an object tracking module designed for efficient real-time execution. Each stage of the pipeline is carefully optimized to balance tracking accuracy, computational efficiency, and scalability, making the approach suitable for high-speed embedded vision systems.

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

II. RELATED WORK

A wide spectrum of object detection and tracking algorithms has been proposed to ensure temporal consistency of object representation across successive video frames. Many early and computationally efficient approaches rely on exploiting distinctive visual features that can be extracted with minimal processing overhead. Among these, color-based techniques remain widely adopted due to their simplicity and strong visual discriminability. Traditional methods typically employ color histograms, threshold-based segmentation, or region-growing strategies to localize objects. More recent developments enhance these approaches by integrating color segmentation with geometric feature extraction, such as line or region boundary detection, thereby improving localization accuracy in dynamic and cluttered environments

Despite their advantages, color-driven methods are inherently sensitive to illumination changes, shadows, and sensor noise, which can significantly degrade tracking reliability. To mitigate these limitations, edge-based features have been extensively incorporated into object tracking frameworks. Edge features represent object boundaries through local intensity gradients and are commonly extracted using operators such as Sobel, Prewitt, or Canny. Contemporary studies demonstrate that directional edge features, when efficiently implemented in hardware-friendly architectures, provide robust boundary preservation and improved detection accuracy under varying environmental conditions. Such designs are particularly effective when deployed on adaptive VLSI platforms optimized for real-time vision processing.

Hybrid tracking techniques that combine edge information with feature matching or region-based analysis have further improved detection precision and temporal stability. These methods exploit complementary strengths of different feature types, enabling more accurate object delineation while maintaining low computational complexity. Owing to their parallelizable structure and reduced algorithmic overhead,such hybrid approaches are well suited for real-time embedded systems where latency and power efficiency are critical constraints.

Motion-based techniques form another major category of object tracking algorithms. Among these, optical flow–based methods estimate object motion by analyzing pixel intensity variations between consecutive frames. Beyond tracking object trajectories, optical flow frameworks can also estimate motion direction and velocity, providing richer temporal information. Recent implementations focus on optimizing optical flow computations through parallel processing and hardware acceleration, allowing real- time performance in high-resolution video streams and embedded vision platforms.

To further enhance robustness in complex scenes, feature fusion strategies have been introduced that integrate multiple low-level cues within a unified tracking framework. These approaches combine color, edge, motion, and texture features to achieve improved resilience against occlusion, background clutter, and illumination variations. By leveraging complementary feature characteristics, fusion-based models

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

demonstrate superior tracking accuracy and stability compared to single-feature methods, particularly in real-world surveillance and monitoring scenarios.

2.1 DCT-BASED COMPRESSION

In adaptive VLSI architectures developed for real-time object detection and tracking, efficient management of high-resolution video data is essential due to stringent constraints on memory bandwidth, storage capacity, and processing latency.

Without effective compression, continuous frame acquisition rapidly overwhelms embedded hardware resources. To address this challenge, transform-based compression techniques are integrated into the preprocessing pipeline, among which the Discrete Cosine Transform (DCT) remains one of the most widely adopted solutions.

A key advantage of the DCT is its energy compaction property, whereby most of the signal energy is concentrated in the low-frequency coefficients (i.e., small u and v). This allows higher-frequency coefficients—often associated with noise and minor texture variations—to be quantized or discarded with negligible impact on perceptual quality and structural information [266]. Such characteristics are particularly beneficial for VLSI implementations, as they reduce arithmetic complexity and memory access overhead.

Within the proposed adaptive VLSI framework, compression is further optimized by prioritizing the luminance component of the video signal. Input frames are represented in the YCbCr color space, and the DCT is selectively applied to the luminance Y channel, which carries the dominant edge and structural information required for reliable object detection [268]. The luminance value Y(x,y)is computed as:

$$Y(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y) \quad (2)$$

where R(x,y), G(x,y), and B(x,y) denote the red, green, and blue components of the input pixel.

Operationally, the luminance plane of each frame is divided into 8×8 blocks, and the DCT is computed for each block. Retaining primarily the DC and low-frequency AC coefficients produces a compressed representation of size m×n, where m×n denotes 8 *8 the original frame resolution. Frame differencing is then performed in the compressed domain by subtracting corresponding coefficients of consecutive frames:

$$Dk(u, v) = [Fk(u, v) - Fk-1(u, v)]$$

where Fk(u,v) and Fk−1 (u,v) are the DCT coefficients of the current and previous frames, respectively.

Thresholding of the resulting difference coefficients enables isolation of dynamic regions associated with object motion, thereby identifying the Region of Interest (ROI). This selective processing significantly reduces computational overhead while improving detection accuracy, as subsequent stages operate only on motion-relevant regions .

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

Overall, the integration of DCT-based compression within the adaptive VLSI architecture provides an effective balance between data reduction and information preservation. It supports real-time object detection and tracking by minimizing memory usage and processing complexity, while maintaining sufficient spatial and temporal fidelity for robust performance in dynamic video environments.

2.2 LOCAL TERNARY PATTERN (LTP)

In real-time object detection and tracking systems, particularly those deployed in dynamically changing and noise-prone environments, the reliability of texture descriptors plays a decisive role in overall system performance. Conventional pixel- level descriptors often degrade under illumination variation, sensor noise, and background clutter—conditions that are typical in practical surveillance and embedded vision applications. To address these limitations, the Local Ternary Pattern (LTP) operator has been adopted as a robust texture feature extraction technique, extending the capabilities of the widely used Local Binary Pattern (LBP) by introducing noise tolerance through ternary quantization .

The LTP operator evaluates local texture information within a 3×3 neighborhood by comparing each neighboring pixel intensity P with the corresponding center pixel intensity Pc. Unlike LBP, which relies on strict binary thresholding, LTP employs a user-defined threshold T to generate three possible outcomes: +1, 0, and −1. This ternary decision mechanism effectively suppresses minor intensity fluctuations caused by noise or illumination changes, while preserving meaningful

structural variations essential for object discrimination. The LTP encoding rule is formally expressed in Equation (1):

$$LTP(P, P_c) = \begin{cases} +1, & \text{if } P \geq P_c + T \\ 0, & \text{if } |P - P_c| < T \\ -1, & \text{if } P \leq P_c - T \end{cases}$$

Here, the threshold T functions as a noise-insensitivity parameter that controls the trade-off between robustness and sensitivity. In the proposed adaptive VLSI-based object tracking framework, the value of T is empirically fixed at 0.1. This choice ensures that insignificant local variations are ignored, while edge-related and texture- rich regions relevant to object identification remain distinctly encoded. Such threshold tuning is particularly important in hardware-oriented implementations, where excessive sensitivity can lead to unstable detection and increased computational overhead

A direct consequence of ternary encoding is the exponential growth in histogram dimensionality. For a standard neighborhood comprising eight surrounding pixels, the resulting LTP histogram spans 38=6561 bins, significantly increasing memory and processing requirements . This high dimensionality poses practical challenges for real-time VLSI realization, especially under stringent power and silicon area constraints.

To mitigate this complexity, the ternary LTP representation is decomposed into two independent binary patterns: the positive LTP and the negative LTP. The positive pattern captures only the +1

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

responses, while the negative pattern encodes the −1 responses, with zero values suppressed in both cases. The corresponding formulations are given in Equations (5.7) and (5.8), respectively:

Positive LTP: retains only $P \geq Pc + T$ comparisons Equation 5.7

Negative LTP: retains only $P \leq Pc - T$ comparisons Equation 5.8

This dual-channel binary decomposition substantially reduces computational and storage complexity while preserving the discriminative strength of the original ternary descriptor. From a hardware design perspective, this approach is well-suited for FPGA and ASIC implementations, as it allows parallel processing using simple comparators, adders, and shift registers, avoiding costly multi-level arithmetic operations.

By integrating the LTP descriptor within the adaptive VLSI framework, the proposed system achieves enhanced robustness against noise and illumination variability while maintaining low computational complexity. This balance between accuracy and hardware efficiency makes LTP particularly suitable for real-time object detection and tracking applications in resource-constrained embedded vision systems.

2.3 FEATURE EXTRACTION WITH LARGEST DIFFERENCE INDEXED LOCAL TERNARY PATTERN (LDILTP)

Robust feature extraction plays a decisive role in object detection systems operating under dynamic and visually complex environments. Variations in illumination, sensor noise, and background motion can significantly degrade detection accuracy if discriminative texture features are not reliably captured. In this context, the Local Ternary Pattern (LTP) descriptor has been widely adopted due to its improved resilience against noise and minor intensity fluctuations when compared with the conventional Local Binary Pattern (LBP). By introducing a ternary encoding scheme with a neutral threshold state, LTP suppresses insignificant gray-level variations, thereby yielding more stable and noise-tolerant texture representations.

Despite these advantages, classical LTP-based feature extraction suffers from several practical limitations, particularly when targeted for real-time hardware realization. The ternary encoding process leads to an expanded histogram dimensionality relative to LBP, resulting in increased memory requirements and higher computational overhead. Such complexity poses challenges for Very-Large-Scale Integration (VLSI) implementations, where constraints on power, silicon area, and latency are critical. Additionally, the conventional approach of separating positive and negative ternary patterns may inadvertently discard important structural relationships within the local neighborhood. As a consequence, redundant or weakly informative patterns are often retained, while certain discriminative spatial cues remain under utilized .

To overcome these limitations, the Largest Difference Indexed Local Ternary Pattern (LDILTP) is adopted as an efficient and hardware-oriented feature descriptor. The LDILTP formulation is

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

specifically designed to reduce redundancy in ternary encoding while preserving the most informative local contrast characteristics. Instead of encoding all ternary comparisons uniformly, LDILTP prioritizes the neighbor exhibiting the maximum intensity difference relative to the central pixel, thereby focusing feature representation on the most salient local variation. This selective indexing mechanism significantly reduces computational complexity and histogram size, making LDILTP well suited for adaptive VLSI architectures intended for real- time processing.

In the proposed framework, LDILTP is computed on the luminance (Y) component of the YCbCr color space. Operating on luminance information enhances perceptual relevance while simultaneously reducing computational load, as chrominance channels are excluded from feature computation. This design choice aligns with hardware-efficient vision systems, where luminance carries the dominant structural and textural information required for object detection.

III.PROPOSED WORK

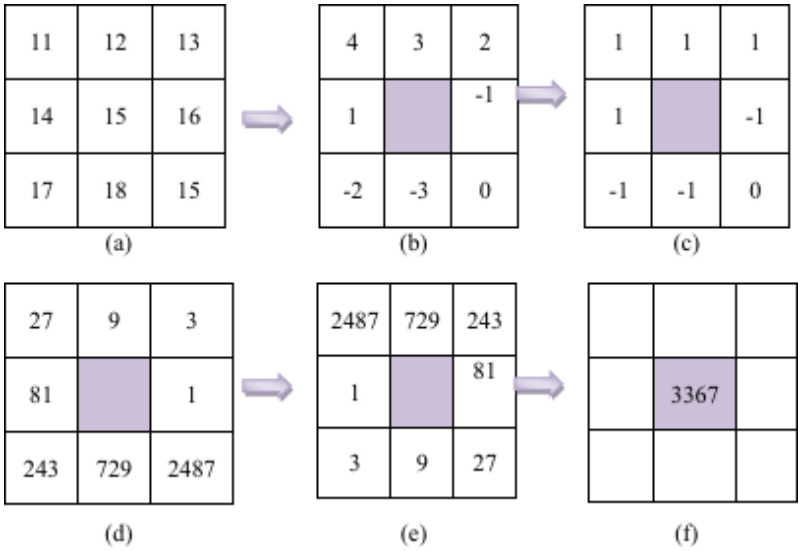

Figure 1: LDILTP Computation Steps and Output

Figure 2 illustrates the computational workflow of LDILTP. For each pixel, a 3×3 neighborhood is considered, and the intensity differences between the central pixel and its surrounding neighbors are first computed to form a difference matrix. These differences are then ternarized using a predefined threshold. Subsequently, a multiplier matrix based on powers of three is dynamically aligned such that the neighbor exhibiting the largest absolute intensity difference is assigned the highest weight. The element-wise multiplication of the ternary matrix and the reordered multiplier matrix, followed by summation, yields the final LDILTP descriptor value.

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

ALGORITHM OF FEATURE EXTRACTION USING LDILTP

Algorithm: ExtractLDILTP

Input: YCbCr frame

Output: LDILTP Feature vector

Description: Computes the LDILTP features for YCbCr frame

Steps:

Initialize:

Initialize Ltpimg to zero

Multiplier = [[1, 2, 4]; [8, 0, 16]; [32, 64,128]]

For each X ← 1 to height of frame

For each Y ← 1 to width of frame

Divide the region into 3 x 3 disjoint blocks

Find the difference by subtracting the

center pixel value from neighboring pixel

Apply thresholding function, S to find thresholded binary matrix

Rotate the multiplier so that the largest difference value align with largest multiplier value

Find the LDILTP by taking the dot product of multiplier and binary matrix

Endfor

Endfor

Figure 2: LDILTP Computation Demonstrating Rotation Invariance.

A key advantage of LDILTP lies in its inherent rotation invariance, which is essential for reliable object tracking in real-world scenarios where object orientation may vary unpredictably. As demonstrated in Figure 5.4, identical LDILTP values are obtained for both original and rotated neighborhood configurations. This invariance is achieved by indexing features based on the relative position of the maximum intensity difference rather than the absolute spatial arrangement of neighboring pixels. Consequently, the descriptor remains stable under rotational transformations.

The rotational consistency of LDILTP is further validated through comparative visualization, where both rotated and non-rotated patterns produce identical

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026, © 2026 JLISNT

LDILTP indices. This property ensures that spatial texture characteristics are preserved regardless of orientation changes, thereby enhancing detection robustness. Owing to its reduced redundancy, rotation invariance, and VLSI-friendly computational structure, LDILTP is particularly effective for real-time object detection and tracking applications, including video surveillance, autonomous navigation, and robotic vision systems operating under dynamic environmental conditions.

3.1 FILLING GAPS IN OBJECT REGIONS

In block-based object detection and tracking systems operating under dynamic and cluttered environments, the quality of foreground region formation plays a decisive role in ensuring reliable object recognition. Initial segmentation stages—particularly those relying on threshold-based foreground extraction—often yield incomplete or fragmented object masks. these imperfections manifest as small holes, discontinuities, spurious noise pixels, and broken boundaries, primarily due to illumination variation, motion blur, background dynamics, and block-wise approximation errors . Such deficiencies can significantly degrade detection accuracy and temporal stability, especially in adaptive VLSI architectures where low- complexity operations are favored for real-time processing.

To overcome these limitations, morphological image processing techniques are incorporated as a post-processing stage to refine the detected object regions. Morphological operations offer a computationally efficient means to enhance spatial coherence by repairing discontinuities, suppressing isolated noise, and smoothing object contours. Their deterministic logic structure makes them particularly suitable for hardware realization in VLSI-based embedded vision systems . By applying these operations to the binary foreground masks generated by the adaptive block partitioning framework, object integrity is restored without incurring excessive computational overhead. The refined output facilitates more accurate object localization and tracking, .

Dilation: Expanding Object Boundaries, Dilation is a fundamental morphological operation employed to expand the spatial extent of foreground regions in a binary image. Conceptually, it grows object boundaries by adding pixels to regions where structural connectivity is weak or partially broken. This operation is particularly effective in closing small gaps within object interiors and reconnecting disjointed segments caused by noise or block-level segmentation artifacts [275]. In dynamic scenes, dilation enhances robustness by compensating for minor detection inconsistencies introduced during adaptive block processing.

The hit-based activation mechanism inherent in dilation aligns well with VLSI design principles, as it can be implemented using basic logical operations and localized memory access. Within the proposed adaptive architecture, dilation serves to reinforce object continuity, ensuring that subsequent stages—such as feature extraction or tracking—operate on structurally coherent regions

Figure 3 illustrates the complementary operation of erosion, which reduces object boundaries. In

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

practice, dilation and erosion are often combined sequentially to form composite operations such as closing, enabling effective gap filling while maintaining boundary precision in real-time embedded vision systems.

Figure 3: Binary Image Decay

Erosion is a morphological operation designed to contract foreground regions by systematically removing boundary pixels that do not satisfy structural consistency constraints. Unlike dilation, which expands object regions, erosion selectively eliminates weakly connected or isolated pixels, thereby suppressing noise and sharpening object contours . This operation is essential in refining binary masks generated under fluctuating environmental conditions, where transient artifacts and background interference may distort object boundaries.

In real-time object tracking scenarios, erosion contributes to enhanced feature stability by isolating dominant structural components of detected objects. By reducing spurious foreground pixels and thinning object edges, erosion improves the reliability of shape-based descriptors and motion cues, which are critical for adaptive VLSI- based tracking systems deployed in surveillance, autonomous navigation, and smart sensing applications .

Mathematically, erosion is defined using a structuring element that must fully fit within the foreground region of the input image. Let fff denote the binary input image, sss the structuring element, and ggg the eroded output image. The erosion operation retains a pixel in g only if all corresponding pixels under the structuring element in fff are foreground; otherwise, the pixel is removed. This strict fitting criterion ensures effective noise suppression and boundary regularization.

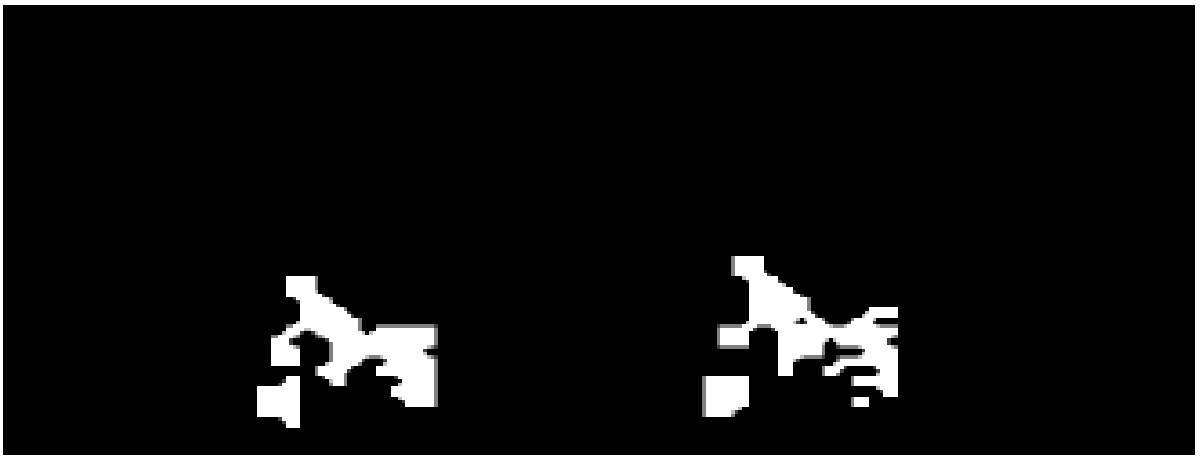

Figure 4: The Filled Gap Binary Image

Figure 4 presents a binary image after erosion, highlighting the elimination of minor artifacts and improved object delineation. Within adaptive VLSI architectures, erosion is typically implemented using dedicated hardware modules, enabling high- throughput preprocessing without compromising

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

real-time constraints. Moreover, erosion acts as a foundational component of higher-order morphological operations— such as opening and closing—which further enhance segmentation quality by balancing gap filling and noise emoval .

3.2 ESTIMATION OF MOTION VECTO

Figure 5. Object Area— initial frame

Motion vector estimation constitutes a critical stage in adaptive VLSI-based object tracking systems, particularly in dynamic visual environments characterized by rapid object displacement, background variations, and illumination changes.

In the context of object tracking, motion vectors provide a compact and effective mechanism for predicting object locations using historical frame data. By exploiting temporal coherence, the tracking system can confine its search space to motion- consistent regions rather than performing exhaustive frame-wide searches. This targeted search strategy significantly reduces computational complexity and memory access overhead. In compressed-domain processing, motion vectors further facilitate object localization by operating directly on block-level motion information, which is particularly advantageous for real-time and resource-constrained VLSI architectures.

A major advantage of motion vector-based tracking in VLSI implementations lies in its minimal preprocessing requirement. Unlike pixel-domain approaches that demand intensive per-pixel operations and frame buffering, motion vectors extracted from encoded bitstreams offer a lightweight alternative. This reduction in arithmetic complexity and memory bandwidth consumption makes motion vector estimation especially suitable for embedded systems, where power efficiency, silicon area, and real-time performance are critical design constraints . Such characteristics align well with adaptive block-based architectures that prioritize low latency and parallel processing.

In the proposed adaptive tracking framework, motion vector estimation is performed following the initial object detection stage. Once the region of interest (ROI) is identified, each block within the ROI

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

is assigned a unique label to facilitate temporal correspondence. To enhance robustness against background interference, the second- largest connected component in the initial frame is selected as the reference object, effectively suppressing spurious background regions while preserving the dominant moving object. This selection strategy ensures stable motion tracking across subsequent frames, even under partial occlusion or background fluctuations.

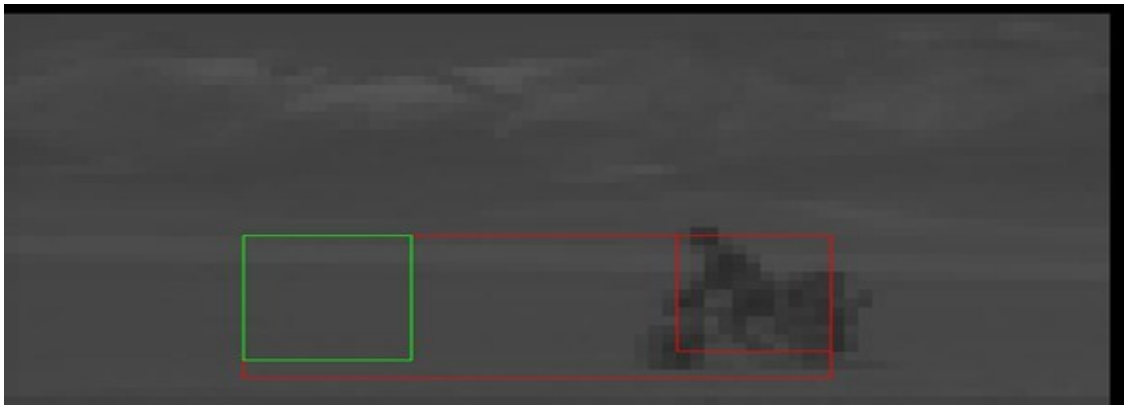

Figure 6: Object Area - Sequential Frame.

The effectiveness of this approach is illustrated in Figures 5 and 6, which depict the extracted object regions in the initial and sequential frames, respectively. The consistent spatial alignment of the tracked region across frames demonstrates the reliability of the motion vector estimation process in maintaining object continuity over time. By integrating motion vector information with adaptive block partitioning, the proposed system achieves accurate and computationally efficient object tracking suitable for real-time VLSI implementation.

3.3 SIMILARITY MEASURE

In adaptive VLSI-based architectures for real-time object detection and tracking in dynamic environments, the similarity measure functions as a core decision mechanism that governs tracking continuity and localization accuracy across consecutive video frames. The principal role of this module is to establish reliable correspondence between object features extracted from successive frames while maintaining robustness against illumination variations, partial occlusions, background dynamics, and motion-induced distortions.

To satisfy both robustness and hardware efficiency requirements, the proposed system employs a texture-driven similarity framework based on the Local Ternary Pattern

(LTP) representation. LTP is particularly suitable for VLSI realization due to its resilience to small intensity fluctuations and its ability to encode local spatial structures using simple comparison operations. Each video frame is partitioned into a regular grid of blocks, and an LTP histogram is computed for every block to capture localized texture characteristics. These histograms serve as

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

compact descriptors that facilitate efficient inter-frame matching.

During the tracking process, candidate blocks within a predefined search window in the current frame are evaluated by computing their similarity scores with respect to the reference block. The block that produces the maximum histogram intersection value is selected as the most probable match, thereby determining the updated spatial position of the tracked object. A higher similarity score indicates stronger correspondence, ensuring reliable object localization even under dynamic scene conditions [283].

From a VLSI design standpoint, the histogram intersection–based similarity measure offers substantial advantages. The computation relies primarily on minimum selection, addition, and accumulation operations, which are inherently parallelizable and map efficiently onto hardware architectures. The avoidance of complex arithmetic operations such as division or floating-point computation significantly reduces latency, power consumption, and silicon area, making the method highly suitable for real-time embedded vision systems [284].

Moreover, incorporating this similarity measure within an adaptive VLSI framework enhances tracking robustness in the presence of scale variations, motion blur, and partial occlusions. By emphasizing texture consistency rather than raw pixel intensity alone, the system achieves stable tracking performance while maintaining energy efficiency an essential requirement for continuous operation in resource-constrained VLSI-based object detection and tracking platforms.

IV.RESULTS AND DISCUSSION

The performance of the proposed VLSI-based adaptive object tracking architecture was evaluated through extensive experimentation under both controlled and real- world conditions. The implementation was carried out using IDL 6.3, enabling detailed visualization of intermediate processing stages as well as quantitative performance assessment. The experimental outcomes demonstrate that the proposed architecture achieves reliable object detection and tracking with improved accuracy and computational efficiency in comparison to conventional block-based approaches.

To examine the algorithmic behavior in a controlled setting, a synthetic video sequence consisting of selected representative frames was employed. This allowed systematic observation of each processing stage. As shown in Figures 5.11(a) and 5.11(b), the input frames were initially captured in the RGB color space. These frames were converted into the YCbCr color domain to decouple luminance from chrominance information, thereby improving robustness against illumination variations . The luminance component Y(x,y) was obtained as:

$$Y(x, y) = 0.299R(x, y) + 0.587G(x, y) + 0.114B(x, y)$$

These thresholds enabled dynamic switching between background classification, partial block subdivision, and full foreground detection, reducing unnecessary computation while preserving

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

detection accuracy.The final object detection and tracking results are shown in Figures , where red-filled rectangles indicate detected objects and blue boundaries represent the identified ROIs. The robustness of the proposed system was further validated through real-world experiments. Figure 5.12 illustrates successful tracking of a moving bicycle in an outdoor environment with background clutter,The proposed architecture consistently outperforms these methods in terms of sensitivity and overall accuracy, confirming the effectiveness of adaptive block partitioning and edge-aware processing within a VLSI framework.

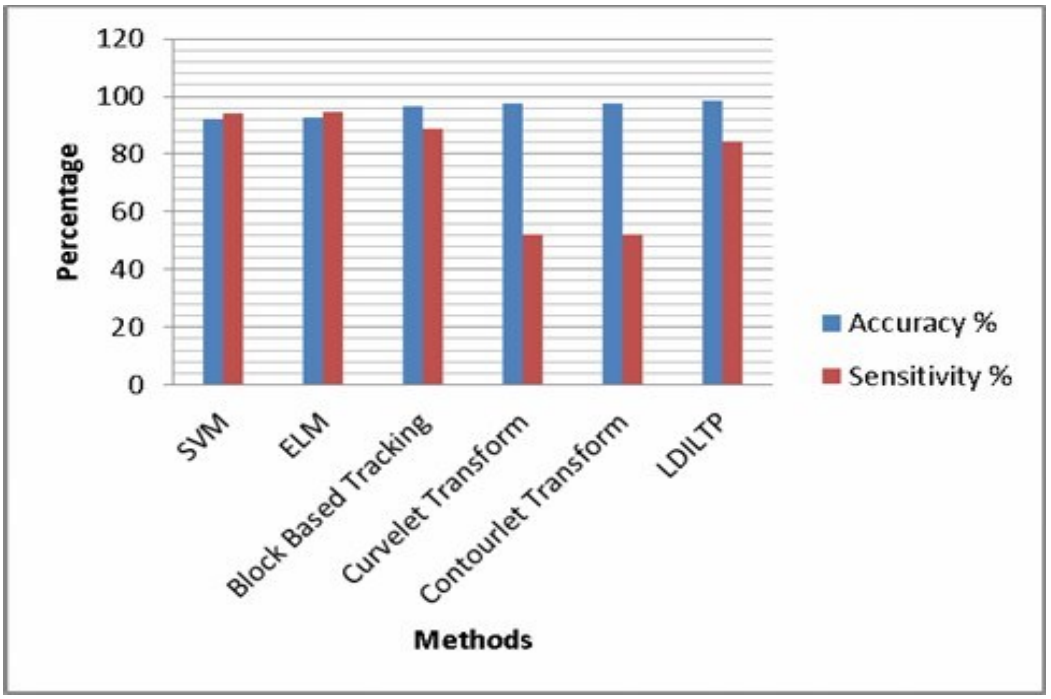

Table 2: A comparison with current best practices

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com

Accepted Jan.2026,

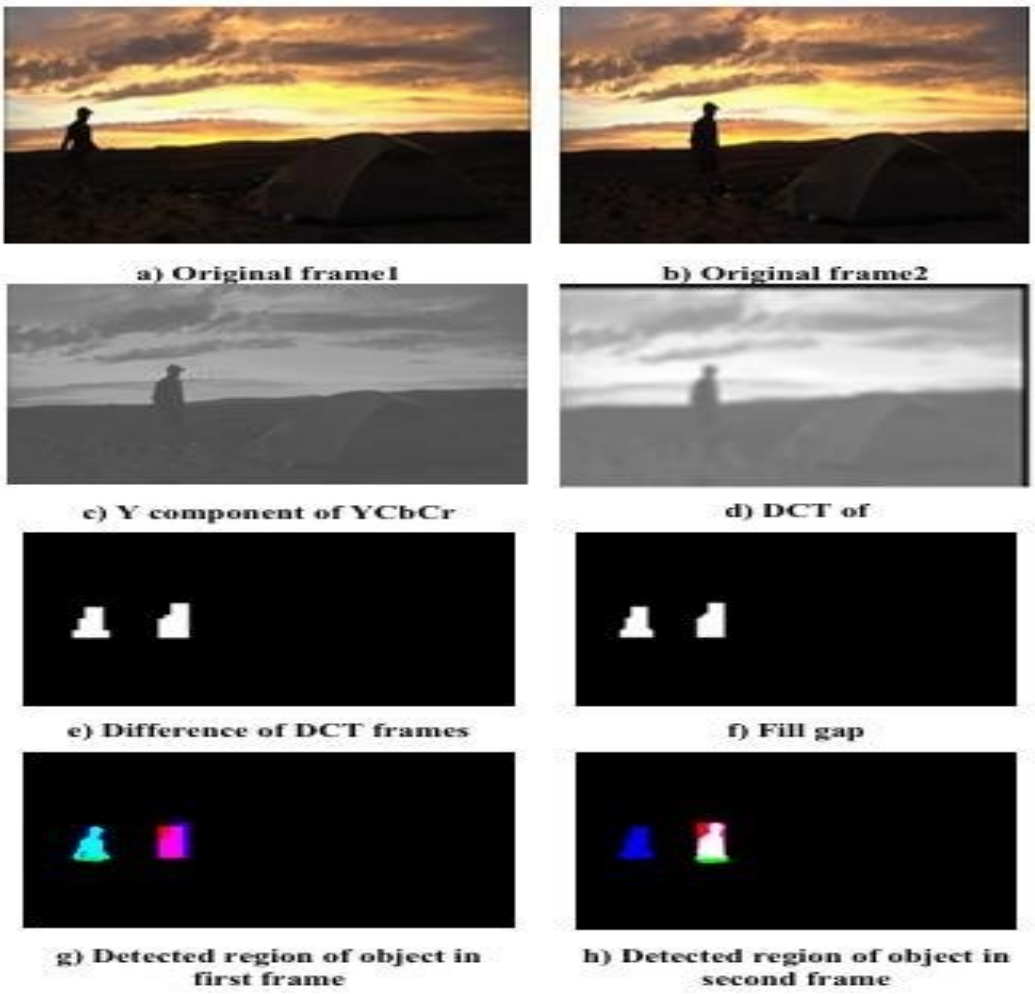

Figure 7, 8: Comparison & Monitoring a moving man against a black backdrop

V. CONCLUSION

This study has presented a low-complexity and adaptive VLSI architecture for robust object detection and tracking in dynamic visual environments, with particular emphasis on real-time and resource-constrained applications. The proposed framework is centered on the Adaptive Block Partition Decision (ABPD) algorithm, which operates in the block domain and intelligently subdivides image regions to preserve structural details while minimizing computational overhead. By integrating sub-block difference analysis with edge-aware partitioning, the architecture effectively addresses common challenges such as illumination variation, background dynamics, and motion intensity changes.While the current implementation delivers strong performance for single-object detection and tracking, challenges such as severe occlusions, dense multi-object scenarios, and highly cluttered backgrounds remain open research directions. Future extensions may incorporate lightweight learning mechanisms,

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026,

multi-sensor data fusion, or hybrid edge-AI strategies to further enhance robustness and scalability. Overall, the proposed architecture represents a practical and efficient solution for next-generation embedded vision systems and real-time surveillance applications, bridging the gap between algorithmic robustness and VLSI feasibility.

Funding: This research received no external funding

Data Availability Statement: No new data were generated during the study. All the data are contained within the manuscript.

Acknowledgments: We acknowledge the use of Grammarly and Quill Bot, in correcting the English and Grammatical errors in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Ethics Declaration: This manuscript is a review article and does not involve any studies with human participants or animals performed by the authors. Therefore, ethical approval and informed consent were not required. The authors declare no conflict of interest, and all referenced works have been properly cited.

REFERNCES:

[1]. 1.T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[2]. 2.X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” IEEE Transactions on Image Processing, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

[3]. 3.Z. Chen, Z. Kalal, and J. Matas, “Learning detector for object tracking,” in Proceedings of the International Conference on Computer Vision (ICCV), 2011, pp. 1072–1079.

[4]. 4.S. Avidan, “Ensemble tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[5]. 5.D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564–577,

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com

Accepted Jan.2026,

May 2003.

[6]. 6.K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," Image and Vision Computing, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[7]. 7.M. Isard and A. Blake, "CONDENSATION—conditional density propagation for visual tracking," International Journal of Computer Vision, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[8]. 8.H. Wang, S. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3169–3176.

[9]. 9.J. Ren, X. Jiang, and J. Yuan, "Noise-resistant local binary pattern with an embedded error-correction mechanism," IEEE Transactions on Image Processing, vol. 22, no. 10, pp. 4049–4060, Oct. 2013.

[10]. 10.S. Murala, Q. M. J. Wu, and M. Maheshwari, "Local tetra patterns: A new feature descriptor for content-based image retrieval," IEEE Transactions on Image Processing, vol. 21, no. 5, pp. 2874–2886, May 2012.

[11]. 11.B. Girod, "Efficient video compression using motion-compensated prediction," IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 25–46, Nov. 1998.

[12]. 12.S. W. Lee and A. Zakhor, "Compressed-domain video processing techniques: A survey," IEEE Transactions on Multimedia, vol. 6, no. 1, pp. 1–15, Feb. 2004.

[13]. 13.M. J. Chen, L. G. Chen, and T. D. Chiueh, "Efficient object tracking using compressed domain information," in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2001, pp. 53–56.

[14]. 14.Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[15]. 15.K. S. Kim, S. H. Lee, and J. H. Kim, "A low-power VLSI architecture for real-time object tracking," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 5, pp. 876–888, May 2013.

[16]. 16.Bovik, Handbook of Image and Video Processing, 2nd ed. Academic Press, 2005. H. K. Lee, Y. H. Kim, and S. J. Ko, "Hardware-efficient feature extraction for real-time vision systems," IEEE Transactions on Consumer Electronics, vol. 58, no. 2, pp. 483–490, May 2012.

Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com

Accepted Jan.2026,