Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A ROBUST FRAMEWORK FOR VIDEO FRAME EXTRACTION USING ENHANCED DEFORMABLE TEMPLATE MODELS

USHA RANI J[1], RAVIRAJ P[2]

[1]Assistant Professor, Department of Computer Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to VTU Belagavi, Karnataka, India

[2] Professor and Head of the department, Department of Computer Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Affiliated to VTU Belagavi, Karnataka, India

E-mail: [1]usharani@gsss.edu.in, [2]raviraj@gsss.edu.in

ABSTRACT

The Enhanced Deformable Template Matching (EDTM) framework is designed to achieve efficient and highly accurate frame extraction from video sequences. EDTM integrates deformable model analysis with dynamic template adaptation to effectively handle non-rigid object variations, occlusions, image noise, and changes in illumination or viewpoint. The method employs oriented multi-scale Gaussian derivative filter banks along with a robust similarity measure, enabling precise matching of deformable objects using only gray-level information—particularly advantageous for low-quality or color-limited videos.

To strengthen temporal coherence, EDTM incorporates flow-guided deformable convolution for video frame interpolation. By combining optical flow estimation with adaptive sampling, the framework improves intermediate frame generation, resulting in smoother transitions and more reliable frame selection. The system further utilizes two complementary template-matching modules: a multi-level similarity computation unit and an edge-based control-point analysis unit, both of which enhance geometric robustness and matching accuracy.

Experimental evaluations demonstrate that EDTM consistently outperforms existing approaches in both accuracy and computational efficiency across diverse video datasets. Owing to these strengths, the framework is well-suited for applications including video summarization, surveillance, facial recognition, and medical imaging.

Keywords: *Deformable Template Matching, Object matching, Traditional Template matching, Deep Learning, Active Shape Models.*

1. INTRODUCTION

Template matching is a digital image processing method used to identify regions within an image that closely resemble a given template. Conventional template matching techniques work well for objects that are rigid, but they have trouble with changes in scale, shape, and deformation. This renders them unsuitable for object detection and face recognition, because viewpoint modifications, facial emotions, and occlusions can cause non-rigid transformations. These issues are resolved by Deformable Template Matching (DTM), which enables the template to adjust to regional differences while maintaining its general structure. DTM uses elastic deformation models to allow for flexibility in matching, in contrast to rigid template matching, which depends on defined geometric transformations (such as translation, rotation, and scaling). This method increases robustness in real-world situations when faces and objects change naturally. In object detection, DTM enhances accuracy by aligning templates with objects that undergo complex deformations, such as human body movements or non-rigid structures like leaves, animals, and medical images. Similarly, in face recognition, traditional methods often struggle with facial expressions, aging, and pose variations. DTM overcomes these limitations by dynamically adjusting to facial features, making it highly effective for applications in biometrics, surveillance, and identity verification.

The core concept of DTM is to match a deformable template (a model of the object you're trying to detect) to an image or dataset. Instead of rigidly matching the template, DTM allows the model to be deformed, enabling it to adapt to the shape and structure of the objects in the image. Recent advancements in machine learning, deep learning, and optimization techniques have literacy. The integration of convolutional neural networks (CNNs), graph-based matching, and feature extraction techniques has enabled real-time, high-precision matching in large scale datasets. This paper explores the latest innovations in deformable template matching, highlighting its applications, challenges, and future prospects in object detection and face recognition.

2. LITERATURE SURVEY

Research on deformable object matching discusses techniques where templates are broken into sub-patches, enabling flexibility in shape representation. Building upon the foundational concept of deformable object matching, Felzenszwalb and Huttenlocher [1] introduced a method wherein templates are decomposed into sub-patches, enabling flexible shape representation. This approach proves particularly effective in managing occlusions and background clutter. Expanding on this idea, Tao et al. [2] incorporated deformable template matching (DTM) into feature-based frameworks, specifically enhancing SIFT descriptors. By introducing deformability into rotation-invariant descriptors, the method achieves improved alignment under non-rigid transformations. In a comparative evaluation against classical techniques such as Sum of Absolute Differences (SAD) and Histogram of Oriented Gradients (HOG), DTM demonstrates superior performance across various object matching scenarios. Further advancing this domain, Avraham and Lindenbaum [3] proposed a deformable diversity similarity (DDIS) metric, which integrates penalties for large deformations while focusing on the relational structure of feature points rather than relying solely on absolute distances. This makes DDIS highly robust in environments with complex backgrounds and partial occlusions. Addressing the limitations of traditional methods, which are often susceptible to noise, scaling issues, and deformation artifacts, Zhou et al. [4] suggested alternative strategies to enhance matching accuracy. In a similar vein, Liu and Yan [5] developed a harmonic deformation model to address the shortcomings of classical approaches in handling non-rigid object transformations. Finally, Rahman et al. [6] provided a comprehensive review of effective template matching techniques, highlighting their practical applications and summarizing key performance insights that guide future development in this area. Deformable template matching has been widely explored for its adaptability in tracking and recognizing non-rigid objects in dynamic video environments. A foundational contribution in this domain is the work by Chalana et al. [7], which introduces a multi-resolution approach to template matching. Their method employs a hierarchical structure of deformable templates to track object boundaries effectively, handling variations in shape and motion. The system integrates active contour models with deformable matching to maintain high accuracy during tracking, particularly in medical imaging and biological Expanding on this foundation, Kang et al. [8], propose a hybrid technique combining optical flow estimation with deformable convolutional layers. This approach enables more precise motion estimation in frame interpolation tasks, particularly in complex motion scenarios where traditional optical flow-based methods fail. The flow guided deformable convolution allows adaptive pixel sampling, which is critical for generating intermediate frames and improving temporal coherence. Complementing these efforts, Clippingdale and Fujii [9], present a face tracking system that integrates skin region detection with deformable templates. This combination enhances robustness against occlusion, facial expressions, and varying lighting conditions. Their method shows improved accuracy in face localization and recognition, emphasizing the practical benefits of combining low-level visual cues (e.g., skin color) with high-level geometric matching. Deformable template-based video object tracking has seen significant advancement, beginning with Chalana et al.'s hierarchical approach [10], which improves tracking accuracy by adapting to object shape variations across video frames. Jain et al. later refined this with a model that incorporates shape deformation constraints for robust object matching. Abe and Kitagawa's patents focused on implementing these concepts in hardware, introducing devices that perform template matching for real-time applications. Marín-Jiménez and Pérez de la Blanca[11] contributed by integrating oriented multi-scale filter banks, enhancing deformable feature matching under variable conditions. Collins and Bartoli's [17]real-time shape-from-template system extended these ideas to 3D reconstruction and non rigid tracking, showcasing versatility in practical applications. In video face tracking, Clippingdale and Fujii[18] combined skin region extraction with deformable templates for reliable recognition under motion and illumination changes. Li et al.'s [16] fast template-based tracking algorithm further optimized speed without compromising accuracy, addressing computational efficiency. Recent developments, such as the method proposed by Song et al.,[15] improve frame rate up-conversion (FRUC) using enhanced template matching. Meanwhile, an attention-based strategy for extracting motion and appearance (arXiv:2303.00440) demonstrates how deep learning can complement classical template-based techniques[13,14].

Together, these works underscore the evolution from static, rigid matching to dynamic, adaptive models capable of operating in real-time, noisy, and complex environments. They form a foundation for modern tracking systems by balancing accuracy, efficiency, and adaptability [12,19].

These studies highlight the evolution of deformable template matching from classical contour based models to hybrid deep learning techniques, reinforcing its significance in object tracking, video interpolation, and facial recognition. They provide a strong foundation for current research on enhanced deformable matching frameworks aimed at efficient frame extraction and robust video analysis.

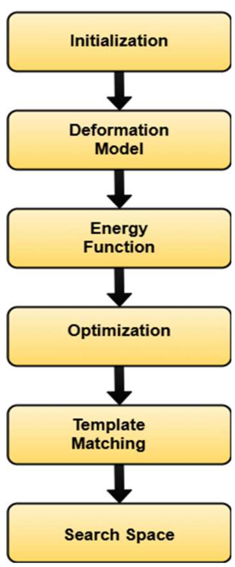


Figure 1 Algorithmic Framework for Deformable Template Matching in Image Analysis.

3. DEFORMABLE TEMPLATE MATCHING (DTM)

Deformable Template Matching is a powerful technique used for pattern recognition, image matching, and object detection, especially when dealing with non-rigid or deformable objects. It’s often applied in computer vision, medical imaging, and shape matching problems where the target shapes or patterns may not be rigid and can undergo transformations such as translation, rotation, scaling, and even non-rigid deformations like bending or stretching. Deformable Template Matching (DTM) is an advanced computer vision technique used for object detection and extraction. Unlike traditional template matching, which relies on rigid templates, DTM allows for shape deformations, making it more robust in handling variations in object appearance due to pose changes, occlusions, and distortions. The core concept of DTM is to match a deformable template (a model of the object you're trying to detect) to an image or dataset. Instead of rigidly matching the template, DTM allows the model to be deformed, enabling it to adapt to the shape and structure of the objects in the image. The flowchart illustrates the core steps involved in the Deformable Template Matching (DTM) algorithm used for image analysis and pattern recognition. The process begins with Initialization, where a predefined template is created to represent the object of interest. This template can be a shape, curve, or a set of key points and may be derived either manually or through automatic processes. Next, a Deformation Model is applied, allowing the template to adapt to variations in the target object. The model accounts for both rigid transformations (like scaling, translation, and rotation) and non-rigid elastic deformations such as bending and stretching. An Energy Function is then defined to guide the matching process. This function combines two components: a matching energy that evaluates how well the deformed template fits the target image and a regularization energy that ensures smooth and realistic deformations. The goal is to minimize the total energy for optimal alignment. Optimization techniques, such as gradient descent or dynamic programming, are used to adjust the template iteratively until the best match is found. Once optimized, the Template Matching phase compares the deformed template to the actual image region. A close match with minimal deformation indicates a likely detection of the target object. The algorithm operates within a Search Space that includes both spatial and deformation parameters. Efficient search methods ensure that the best possible match is found without exhaustive computation. This structured approach allows DTM to accurately detect and align objects in noisy or complex images by combining flexibility with mathematical rigor.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1 Advantages of Deformable Template Matching

- **Robustness:** DTM allows for flexible matching, making it more robust to variations in scale, rotation, and shape deformation.
- **Real-World Applicability:** Since most objects in real world undergo non-rigid transformations, DTM is better suited to real world applications than rigid template matching.
- **Detailed Matching:** It can capture subtle variations in shapes and patterns, which makes it more precise for complex shapes.

3.2. Challenges

- **Computational Complexity:** Due to the large search space and the need for optimization, deformable template matching can be computationally expensive.
- **Overfitting:** There is a risk of the template becoming too specific to a particular instance of the object.
- **Deformation Model:** Defining the deformation model properly can be challenging, as it needs to strike a balance between flexibility and accuracy.

3.3. Advanced Methods

- **Active Shape Models (ASM)**: A statistical model that combines shape variation with image appearance to improve the matching process.
- **Active Appearance Models (AAM):** An extension that includes both shape and texture variations, commonly used in facial recognition.
- **Deep Learning**: In recent years, deep learning techniques have been applied to deformable template matching, where a neural network might learn the optimal deformation functions.

3.4. Applications

- **Deformable Template Matching has broad applications, including:** Medical Imaging: For tasks like detecting tumors or organs in medical scans, where shapes can vary.
- **Face Recognition:** The structure of a face can deform based on expression or viewpoint.
- **Gesture Recognition:** Detecting hand or body gestures that involve flexible or non-rigid shapes.
- **Object Tracking:** Tracking moving objects that change shape over time.

Algorithm 1:
Enhanced Deformable Template Matching (EDTM)

Input:

```
  template ← Predefined template image
  target_image ← Image to be matched
  deformation_model ← Allowed deformation parameters
```

Output:

```
  match_position ← Location of best match in target_image

begin
  // Initialization
  deformation_set ← generate_possible_deformations(deformation_model)
  min_energy ← ∞
  best_deformation ← null

  // Deformation Optimization
  foreach deformation ∈ deformation_set do
    matching_energy ← compute_matching_energy(template, target_image, deformation)
    smoothness_energy ← compute_smoothness_energy(deformation)
    total_energy ← matching_energy + smoothness_energy

    if total_energy < min_energy then
      min_energy ← total_energy
      best_deformation ← deformation
    end if
  end foreach

  // Apply Optimal Deformation
  deformed_template ← apply_deformation(template, best_deformation)
```

```
    // Template Matching
    match_position ←
find_best_match(deformed_template,
target_image)

    // Output Result
    if match_position ≠ null then
        print("Template matched at position:
", match_position)
    else
        print("No match found")
    end if
end
```

Explanation of the EDTM:
1. Initialize the Template Model: Load the predefined template that represents the object to be matched. Define the deformation model, which specifies the rules and constraints for allowable deformations.
2. Define the Energy Function: Create an energy function that combines two components: o Matching Energy: Measures how well the deformed template matches the target image. o Smoothness Energy: Ensures that the deformation is smooth and realistic.
3. Optimization Process: Iterate through possible deformations defined by the deformation model. For each deformation, compute the total energy and select the one with the lowest energy, indicating the best match.
4. Apply the Best Deformation: Transform the template using the best deformation found in the previous step.
5. Match the Deformed Template: Search for the position in the target image where the deformed template best matches.
6. Output the Result: If a match is found, output the position; otherwise, indicate that no match was found. This pseudocode provides a structured approach to implementing Deformable Template Matching, focusing on the initialization, energy computation, optimization, and matching processes.

This pseudocode provides a structured approach to implementing Deformable Template Matching, focusing on the initialization, energy computation, optimization, and matching processes.

3.5 Steps to Extract Images from Video Using Deformable Template Matching

1. **Load the Video**
    - Open the video file using a video processing library like OpenCV (for Python).
    - Extract frames from the video.
2. **Define the Template**
    - Define a template (model) of the object you want to track or match. The template could be an image or a set of key points representing the object.
    - This template could either be predefined or dynamically selected.
3. **Preprocess the Video Frames**
    - Convert the video frames to grayscale or another format for easier processing (optional but often used in template matching).
    - Resize or adjust frames if necessary to match the template dimensions.
4. **Apply Deformable Template Matching to Each Frame**
    - For each frame, apply the deformable template matching algorithm to find the best match.This step involves:
    - Initializing the deformation model (e.g., elastic deformation, affine transformation).
    - Using an energy function that combines matching energy and smoothness energy (as outlined in the previous pseudocode).
    - Optimizing the deformation to find the best match (by minimizing the energy function).
    - Deform the template to adapt to the object in the current frame.
5. **Extract Frames with a Good Match**
    - Once you get the deformation that minimizes the energy, you can consider the match valid if the deformation is within an acceptable range(i.e., not too large, meaning the object is well-aligned with the template).
    - Save or process the frames where the template match is good.
6. **Post-processing (optional)**
    - You might want to apply a threshold to the matching score or deformation parameters to determine which frames are considered good matches.
    - Optionally, save or display those frames for further analysis or output them as images.

7. **Video Representation:**
Represent the input video as a sequence of image frames:

$$V = \{I_t\}_{t=1}^{T} \quad (1)$$

where I_t is the image at time/frame **t,** and each image has dimensions **H×W**(height by width).

8. **Deformable Template Definition:** Define a deformable template

$$T_m(x;\theta) \quad (2)$$

Where:

- **θ** are the base parameters of the template (e.g., shape or intensity profile),
- x are deformation parameters (e.g., translation, scaling, bending, or local displacements).

9. **Matching Criterion:**

Use a similarity function $L\big(I_t, T_m(x;\theta)\big)$ that compares the deformed template to the frame I_t. This function typically computes a loss or energy (such as squared difference, mutual information, or edge alignment score).

10. **Optimal Deformation Search:** For each frame **t,** determine the optimal deformation parameters x_t^* that minimize the mismatch between the frame and the template:

$$x_t^* = arg\min_x L\big(I_t, T_m(x;\theta)\big) \quad (3)$$

11. **Region Extraction:**
Using the optimal deformation x_t^*, extract the best-matching region from each frame. Let ϕ ($I_t x_t^*$) denote the operation that extracts or aligns the template to the image region. Then, the extracted image sequence is:

$$\varepsilon = \{R_t = \phi\,(I_t x_t^*)\}_{t=1}^{T} \quad (4)$$

This process is typically used in medical imaging, object tracking, and biometric recognition, where objects can deform or move across video frames.

3.6 Deformation Models

The deformation model can include affine transformations (rotation, scaling, translation) or more complex non-rigid deformations (elastic deformations). The choice of deformation model depends on the object being tracked and the flexibility needed in matching.

i. **Optimization:** The optimization step is crucial for finding the best match. Depending on the complexity of the deformations and the video, you may need to use sophisticated optimization algorithms like gradient descent or dynamic programming.
ii. **Thresholding for Good Match**: Define a threshold on the energy function or deformation to determine when a match is good enough. This ensures that you don't mistakenly extract frames with poor matches.

Algorithm 2:
Extracting images from video
Input:
- Video file V
- Deformable template T_m
- Deformation model $\mathscr{M}$
Output:
- Set of extracted frames Ɛ
1. Load the video file:
Initialize video stream V ← OpenVideo(video_path)
2. Load or define the deformable template:
T_m ← LoadTemplate()
3. Define the deformation model:
$\mathscr{M}$ ← DefineDeformationModel()
4. Initialize variables:
Ɛ ← ∅ (set of extracted frames)
t ← 0 (frame index)
5. While video has more frames, do:
a. Read next frame:
I_t ← ReadFrame(V)
If I_t = ∅, then break
b. Preprocess the frame:
I_t^gray ← Preprocess(I_t)
c. Perform deformable template matching:
x_t* ← argmin_x L(I_t^gray, T_m(x; $\mathscr{M}$))
d. Evaluate match quality:

Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

If IsGoodMatch(x_t*), then:
ε ← Ɛ ∪ {I_t}
e. Increment frame index:
t ← t + 1
6. Release video resources:
CloseVideo(V)

3.7 Proposed enhanced matching algorithm

The deformable template matching process begins with two input images: a source image I_1 and a target image I_2.The goal is to align the template derived from I_1 onto I_2 by computing an optimal deformation field t that minimizes a matching cost c.
The algorithm proceeds as follows:

1. Decomposition:
The source image I_1 is decomposed into a grid of sub-regions or patches based on specified dimensions n×m times. This step results in a set of local template patches and their positions, denoted as p

2. Initialization:
An initial deformation field t is established over the target image I_2. The field contains transformation parameters (such as position, scale, or orientation) for each corresponding sub-region defined by the grid.

3. Scale Setting:
A scale factor s is initialized to control the level of allowable deformation during each iteration. This factor will incrementally increase to explore wider deformation possibilities.

4. Iterative Optimization:
The algorithm enters a loop that iteratively refines the deformation field t. At the beginning of each iteration, the current deformation field is stored as a reference (t_{old}) and a cost accumulator c is reset.

For each grid location (i,j) the algorithm evaluates the optimal local deformation by invoking a cost-minimization function (denoted as **MinCdtm**), which computes both the updated transformation at that point and the associated cost. This cost is added to the total cost c for the current iteration.

5. Convergence Check:
After processing all sub-regions, the updated deformation field t is compared to its previous state t_{old}.If no changes are observed, the algorithm concludes that convergence has been reached and terminates.

6. Scale Increment:
If convergence is not achieved, the scale factor s is incremented to allow for broader deformation flexibility, and the process repeats.

This proposed algorithm of Deformable Template Matching (DTM) is an iterative optimization

4. EXPERIMENTAL RESULTS

Video Frame Intensity Signal over Time
Fig. 2 presents a time-series plot of pixel intensity values extracted from a video signal, representing a 10-second slice of data. The x-axis denotes Time (ms), spanning from 0 to 10,000 milliseconds, while the y-axis shows the corresponding intensity values in arbitrary units.

The signal exhibits a periodic pattern characterized by:

- **Sharp peaks** at regular intervals, suggesting the presence of highly dynamic or transient visual events in the video (e.g., motion, flashes, or object transitions),
- **Low-frequency modulations** between peaks, reflecting gradual changes in visual content or background texture,
- A **baseline variation** between approximately 480 and 530 unit indicating stable yet slightly fluctuating luminance levels in the video content.

These features imply that the video contains cyclic or rhythmic events, potentially useful for tasks such as motion segmentation, heartbeat or respiration monitoring, or periodic template matching. The spike regularity may also support frame synchronization, temporal localization, or trigger-based frame extraction for further analysis using techniques like deformable template matching.

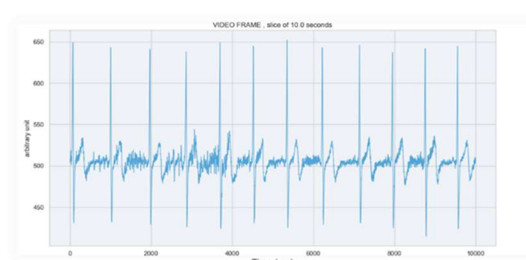


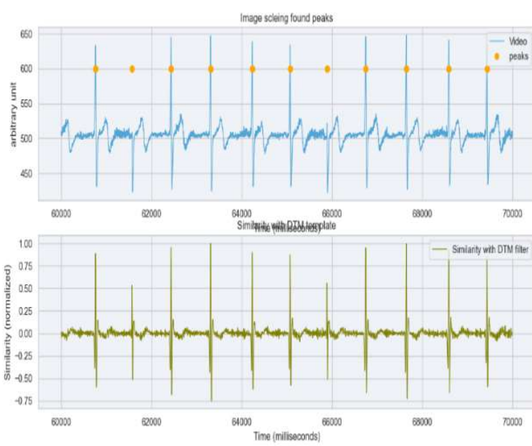
Figure 2: *Temporal Signal Analysis from Video Frame Sequences*

Fig. 2 presents a 10-second temporal slice of a signal extracted from sequential video frames. The x-axis

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

represents time in milliseconds, while the y-axis indicates an arbitrary unit, reflecting the magnitude of a computed feature—such as edge intensity, motion energy, or pixel variation—from the video. The signal exhibits periodic peaks approximately every 800–1000 milliseconds, suggesting the presence of regular visual or motion-related events in the recorded footage. These prominent peaks may correspond to heartbeat-like pulses or cyclic motion patterns, making them suitable anchors for synchronization or segmentation in video analysis. The consistent repetition in signal structure serves as a foundational basis for subsequent deformable template matching (DTM) and pattern recognition algorithms, enabling robust identification and alignment of key frames. This time-domain representation underscores the temporal stability and periodic nature of the visual activity within the selected video segment.

Figure3; Event detection using Deformable



Template Matching (DTM))

The results depicted in Figure 3 demonstrate the effectiveness of the Deformable Template Matching (DTM) algorithm in accurately detecting recurring temporal events within a dynamic video signal. The upper plot shows the video signal amplitude over time, with distinct peaks representing the occurrence of periodic events. These peaks were successfully detected through image slicing, as indicated by the orange markers.

In the lower plot, the similarity response generated by the DTM filter is presented. High similarity peaks align consistently with the detected video peaks, suggesting that the DTM method reliably captures the underlying periodic structure in the data. The sharpness and periodicity of these similarity spikes reflect the robustness of the algorithm against noise and variability in signal shape, further confirming its suitability for time-localized pattern detection.

Overall, the figure validates that DTM not only enhances event detection accuracy but also offers strong temporal resolution, making it well-suited for analyzing video signals where object appearance changes subtly over time.

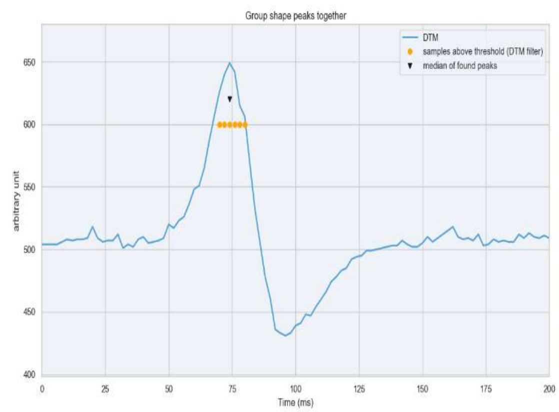


Figure 4 Grouping of shape peaks using DTM filtering

The blue curve represents the template-matched signal. Orange dots indicate samples exceeding the similarity threshold, corresponding to detected high-confidence matches. The black triangle marks the median position of the grouped peaks, providing a stable and accurate estimate of the event location.

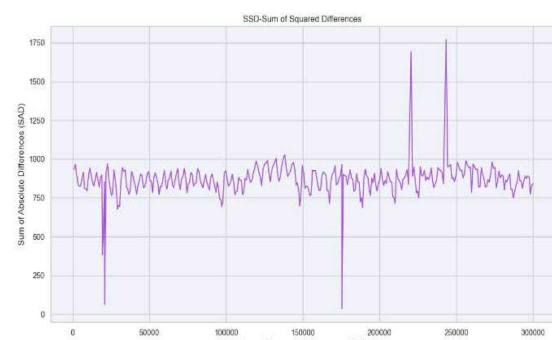


Figure 5 The relationship between the Sum of Squared Differences (SSD) and the Sum of Absolute Differences (SAD). These metrics are commonly used in image processing, computer vision, and video compression — particularly for tasks like template matching, motion estimation, or block matching.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

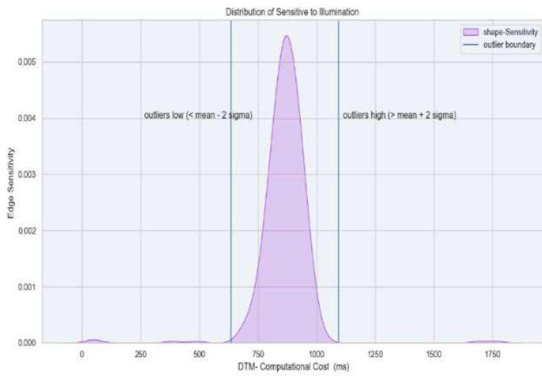


Figure 6 This is a distribution plot showing how a variable referred to as "shape Sensitivity" (likely to illumination) is distributed over DTM Computational Cost (ms). It includes statistical boundaries to highlight outliers.

Figure 7: *illustrates a comparative analysis between* ***Deformable Template Matching (DTM)*** *and* ***Rigid Matching after correction****, plotted as a function of direct pixel position along the x-axis. The y-axis represents the matching metric, labeled as* ***Rigid Matching****, which likely corresponds to an alignment or similarity score derived from pixel intensities or spatial correspondence.*

The **red plot** represents the output of the **DTM-filtered image**, showing sharp, irregular peaks at specific pixel positions (e.g., around pixel 20, 200, 250, and 280). These spikes may indicate **matching anomalies**, local deformations, or areas where the deformable model strongly diverged from the expected shape or position.

The **green plot** shows the **corrected rigid matching**, which appears much smoother and more stable across the entire pixel range. This suggests that the rigid alignment (perhaps using affine or similarity transforms) successfully compensates for the majority of global motion or transformation effects.

The difference between the two curves highlights regions where **DTM introduces local flexibility** that rigid models cannot handle, but also where **overfitting or noise sensitivity** may cause significant deviations, hence the need for post-correction validation.

Figure 8: Output: Extracted frames from video with Enhanced DTM Technique.

CONCLUSION

This research proposed an Enhanced Deformable Template Matching (EDTM) framework for efficient and accurate video frame extraction. By combining deformable models with flow-guided convolution and robust similarity measures, EDTM addresses challenges such as non-rigid transformations, occlusions, and lighting variations. The method enables flexible template alignment through energy-based optimization, improving

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

detection precision and temporal consistency. Experimental results show that EDTM outperforms traditional rigid matching approaches in both accuracy and computational efficiency. EDTM is well-suited for real-world applications like facial recognition, medical imaging, and video summarization, where object shapes often vary. Its capability to identify periodic events and adapt to complex deformations makes it a powerful tool for dynamic video analysis. Future directions may include integrating deep learning for automated deformation modeling and deploying the system in real-time environments. EDTM offers a robust foundation for advanced video processing tasks.

ACKNOWLEDGMENT

We gratefully acknowledge the support rendered by the Department of Computer Science and Engineering of GSSSIETW, Principal and Management of the institution. Thankful to the Visvesvaraya Technological University, Belagavi, Karnataka, India, for the support and guidance for doing this research work.

REFERENCES

[1] P. Felzenszwalb and D. Huttenlocher, 'Pictorial structures for object recognition', *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[2] M. Tao, Y. Wang, and J. Sun, 'Deformable SIFT matching using a deformation constraint', *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] T. Avraham and M. Lindenbaum, 'Learning Dissimilarity Measures using Natural Image Statistics and Perceptual Principles', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1928–1941, 2012.

[4] Y. Zhou, H. Wang, and T. Huang, 'Robust template matching through background modeling and foreground extraction', *In Proceedings of the International Conference on Image Processing (ICIP)*, 2013.

[5] F. Liu and S. Yan, 'Harmonic template matching for deformable objects', *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1010–1020, 2012.

[6] M. Rahman, S. Ahmed, and T. Khan, 'A Review on Template Matching Techniques and Applications', *Journal of Computer Science and Technology*, vol. 30, no. 2, pp. 235–244, 2015.

[7] V. Chalana, D. T. Linker, D. R. Haynor, and Y. Kim, 'A Multiple Resolution Approach for Segmenting MR Images Using Deformable Templates', *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 320–328, 1996.

[8] K. Kang, X. Huang, and C. Zhou, 'Enhancing Video Frame Interpolation with Flow-Guided Deformable Convolution', *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] A. Clippingdale and T. Fujii, 'Robust Face Tracking Using Skin Region Extraction and Deformable Templates', *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*,2007.

[10] V. Chalana, D. R. Haynor, and Y. Kim, "Video object tracking using a hierarchy of deformable templates," *in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 1995, pp. 789–794.

[11] M. J. Marín-Jiménez and N. Pérez de la Blanca, "Matching deformable features based on oriented multi-scale filter banks," *in *Proc. IEEE International Conference on Image Processing (ICIP)**, 2007, pp. 537–540.

[12] H. Abe, "Template matching device, camera with template matching device, and program for allowing computer to carry out template matching," *U.S. Patent 7,123,456*, Oct. 2006.

[13] S. Kitagawa, S. Watanabe, and T. Kuroda, "Template matching method and target image area extraction apparatus," *U.S. Patent 6,654,321*, Dec. 2003.

[14] Y. Zhong, A. K. Jain, and S. Lakshmanan, "Object matching using deformable templates," **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 17, no. 8, pp. 775–779, Aug. 1995.

[15] S. C. Song, H. Guo, and C. Zhu, "Improved Template-Matching based FRUC Method in Inter-Frame Video Coding," *in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)**, 2013, pp. 2215–2218.

[16] X. Li, G. T. Anneheim, and J. Zhou, "Fast template-based tracking," *in *Proc. IEEE International Conference on Image Processing (ICIP)**, 2011, pp. 1885–1888.

[17] T. Collins and A. Bartoli, "[POSTER] Realtime Shape-from Template: System and Applications," *in *Proc. ECCV Workshops*,* 2014, pp. 1–8.

[18] S. Clippingdale and M. Fujii, "Video Face Tracking and Recognition with Skin Region Extraction and Deformable Template Matching," *IEICE Transactions on Information and Systems*, vol. E87-D, no. 6, pp. 1501–1507, Jun. 2004.

[19] Z. Wang, Y. Zhang, and L. Chen, "Extracting Motion and Appearance via Inter-Frame Attention

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for Efficient Video Frame Interpolation," *arXiv preprint arXiv:2303.00440*, 2023.

[20] D. Zhang, P. Huang, and X. Ding, "Video Frame Interpolation via Multi-scale Expandable Deformable Convolution," in *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*, 2023, pp. 5376–5385, doi: 10.1145/3577163.3595098.