Enhancing Video Stability with Object-Centric Stabilization

1st Aparna Tiwari
Department of Mechatronics Engineering
Center for Advance Studies, A.K.T.U.
Lucknow, India
apanati1206@gmail.com

2nd K.S. Venkatesh
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
venkats@iitk.ac.in

3rd Hitika Tiwari
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
hitika@iitk.ac.in

4th Anuj Kumar Sharma
Department of Mechatronics Engineering
Center for Advance Studies, A.K.T.U.
Lucknow, India
sharmaanuj79@gmail.com

***Abstract*—Video stabilization is essential for producing high-quality videos by eliminating unwanted camera motion, resulting in smooth, steady footage that enhances visual coherence. However, existing methods often pose several challenges such as the difficulties in accurately tracking specific objects, and high computational demands. These limitations hinder the ability to achieve optimal stabilization results and limit the practical applicability of video stabilization techniques in real-world scenarios. To address these challenges, we introduce a novel end-to-end method for object-centric video stabilization in the presence of multiple independent objects. Our approach introduces two algorithms: 1) *Object-Centric Video Stabilization* (*OCViS*), which facilitates effective detection and tracking of specific objects within each frame, followed by stabilization with respect to the object of interest, and 2) *Optical Flow-based Transformation Accumulation and Smoothing* (*OFTrAnS*) to remove inherent shakiness and maintain consistent stabilization, even with multiple independent objects. Most importantly, our algorithms achieve these results without relying on deep neural networks or ground-truth data, providing a more efficient and accessible solution for enhancing video quality on mobile devices and handheld cameras. Our experiments demonstrate that the proposed approach significantly outperforms a strong baseline by a large margin emphasizing the efficacy of our method.**

***Index Terms*—Object-Centric Video Stabilization, Shakiness Removal, Multiple Independent Objects, Object Tracking, Video Processing.**

I. Introduction

The task of effectively removing shakiness from video footage is a mathematically ill-posed problem in computer vision. This challenge becomes even more significant when stability needs to be maintained concerning specific objects within the frame. Additionally, the presence of multiple objects in video frames adds complexity to the stabilization process. While numerous techniques exist for addressing frame shakiness in videos, the exploration of object-centric video stabilization with multiple independent moving objects remains relatively unexplored. This area holds significant potential for applications in emerging fields such as drone surveillance [1], autonomous vehicles, augmented reality [2], and sports analytic [3]. Shakiness removal methods [4]–[8] span a range of techniques aimed at stabilizing video footage. These methods can be broadly grouped into four categories, as shown in Fig. 1. More specifically, 2D stabilization methods [9], [10] focus on adjusting the position and rotation of points or regions within the video frame. By analyzing the motion of these reference points, algorithms compensate for shakiness and produce smoother footage. 2D stabilization techniques are particularly effective for handheld footage and involve adjustments that occur within the plane of the image. Unlike 2D stabilization, 3D stabilization methods [11], [12] account for scale and perspective adjustments in addition to position and rotation. These techniques stabilize footage in three dimensions, providing more comprehensive stabilization, especially for shots with significant depth or perspective changes. Motion tracking [13]–[15] involves the identification and tracking of specific points or features across frames of the video. By analyzing the movement of these tracked points, algorithms mitigate shakiness and stabilize the overall footage. Motion tracking is particularly useful for stabilizing footage with moving subjects or dynamic backgrounds. Warped stabilization techniques [16], [17] apply warping and distortion to the video frame to smooth out motion and reduce shakiness. While these methods effectively stabilize handheld footage, they introduce some visual artifacts or distortion to the final output. Moreover, object-centric video stabilization is beyond their scope, particularly in the presence of multiple independent objects. Furthermore, deep learning methods [17]–[20] often utilize supervised videos to stabilize unstable frames to their corresponding stable frames, which imposes a dependency on ground-truth labels and increases computational complexity.

To address the aforementioned challenges, we propose a novel end-to-end algorithm for facilitating object-centric video stabilization in the presence of multiple independent objects. More specifically, we introduce ***O**bject-**C**entric **Vi**deo **S**tabilization*

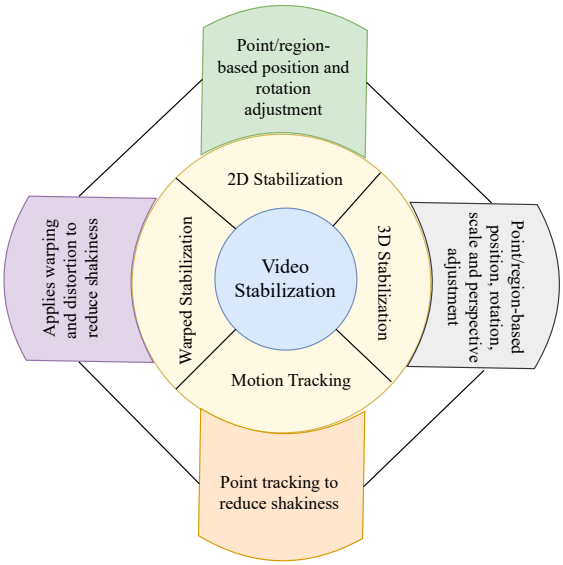


Fig. 1: Overview of the video stabilization techniques.

algorithm, dubbed ***OCViS***, to detect the object of interest within each frame and stabilize the footage accordingly. OCViS achieves this by tracking the coordinates enclosing the object and applying stabilization technique specific to the detected object. Furthermore, we propose ***O****ptical* ***F****low-based* ***Tr****ansformation* ***A****ccumulatio****n*** *and* ***S****moothing* algorithm, dubbed ***OFTrAnS***, to compensate for both the inherent shakiness of the video and the additional shakiness introduced due to stabilizing the object of interest. This is attained by computing the average of optical flow transforms, ensuring a smooth and stable output. Finally, by combining *OCViS* and *OFTrAnS*, our overall method presents a detailed solution to object-centric video stabilization in the presence of multiple independent objects. The main contributions of this work are as follows:

1) To the best of our knowledge, we are the first to present an end-to-end approach for an object-centric video stabilization approach in addition to shakiness removal.
2) Our approach tackles the intricate challenge of object-centric video stabilization by integrating object tracking, stabilization, and shakiness removal into a unified framework. Notably, our methodology is adept at addressing these issues even in scenarios involving multiple independent objects within the video footage.
3) Our approach provides efficiency without relying on computationally intensive deep neural networks, distinguishing it from existing methods. Additionally, it operates independently of ground-truth data, eliminating the need for laborious annotation processes typical in video stabilization techniques.

II. Related Work

Shakiness removal for video stabilization poses a significant challenge in computer vision. The problem becomes severe when the stability is needed with regard to object in addition to the shakiness removal. Presence of multiple objects in a video increases issue multi-fold.

Matsushita et al. [5] introduce a robust video stabilization approach that produces full-frame stabilized videos. This approach of video completion and deblurring maintains original image quality, addressing challenges in maintaining visual quality during stabilization. Feng et al. [21] consider sparse displacements by recovering 3D structure, ensuring content preservation during stabilization. Liu et al. [22] tackle challenges in video stabilization related to sudden depth changes and tracking failures by incorporating an additional depth map. Goldstein et al. [23] present a video stabilization method utilizing epipolar geometry for input modeling and output synthesis. Guilluy et al. [7] extend video stabilization to introducing a Video Stabilization Quality Assessment (VSQA), emphasizing not only on stabilizing video content but also assessing the quality of the stabilization process. Aguilar et al. [8] contribute to the field of microaerial vehicles (MAVs) by introducing a real-time model-based video stabilization technique. Their approach involves geometric transformations, outliers' rejection, and Kalman filtering based on an artificial neural network (ANN) model of MAV motion intention. Zhao et al. [24] propose a novel video stabilization approach based on local trajectories and robust mesh transformation. The method directly smooths local trajectory matrices, demonstrating effectiveness in modeling nonlinear video streams. Hu et al. [25] propose a digital video stabilization approach based on multilayer gray projection. The approach utilizes differential gray projection, ring-projection, and circular projection to estimate translation, scaling, and rotation separately. Sharif et al. [26] improve video stabilization for unmanned aerial vehicles using the SIFT-Log Polar technique. Hussain et al. [6] introduce a robust video stabilization algorithm based on global motion estimation using block matching. The algorithm demonstrates effectiveness in stabilizing YUV videos, particularly in scenarios involving premeditated camera motions.

In recent years, deep learning approaches have gained significant attention in video stabilization. Yu et al. [20] utilize deep learning-based optical flow for pixel-wise warp field inference. Zhao et al. [24] emphasize on learning pixel-wise warping maps for video stabilization. The approach is built upon a multi-stage cascade encoder-decoder architecture. The integration of convolutional neural networks (CNNs) in video stabilization is further explored by Kumawat et al. [19]. Their model, combining 3D CNNs and generative adversarial networks (GANs) with Gaussian Mixture, demonstrates effectiveness in reducing spatiotemporal data for unstable background subtraction. Xu et al. [27] incorporate divide-and-conquer concept from traditional stabilizers to facilitate deep unsupervised learning.

Although these methods demonstrate encouraging performance, addressing the object-centric stabilization for multi-object video is beyond their scope. Furthermore, the aforementioned approaches pose dependencies such as depth map and (or) utilize deep models, which result in inference speed

reduction. Beside, our method addresses the challenge of object-centric stabilization for multi-object video streams in real-time without imposing additional requirements or relying on deep networks.

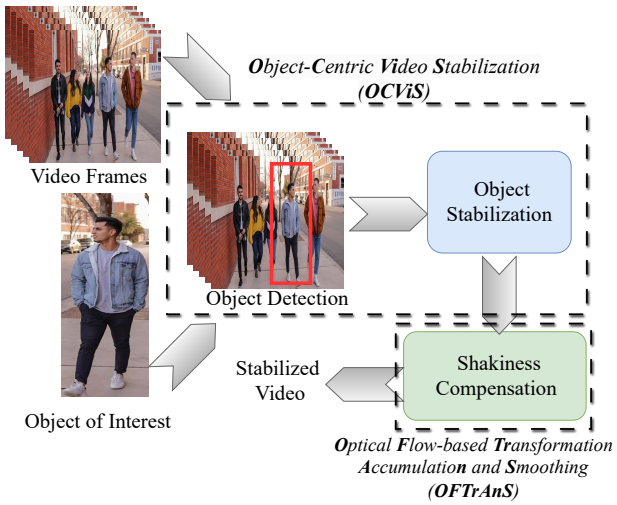


Fig. 2: Simplified overview of the proposed end-to-end object-centric video stabilization approach in presence of multiple independent objects.

III. Methodology

Object-centric video stabilization remains an under-addressed challenge despite advancements in stabilizing shaky videos, particularly with multiple objects, in addition to the computational complexity of deep-learning methods employing deep U-Net architecture. Therefore, our objective is to propose a novel approach that stabilizes videos with respect to objects while also removing video shakiness.(Fig. 2). To achieve this goal, we introduce a method consisting of two main components: 1) Object-centric video stabilization, which stabilizes the video stream relative to an object (Sec. III-A), and 2) Shakiness Removal, to eliminate overall shakiness of the video (Sec. III-B).

A. Object-Centric Video Stabilization

While several methods have been proposed to mitigate camera shakiness in video streams, object-centric video stabilization remains an insufficiently addressed concern. The problem becomes significantly more challenging in the presence of multiple independent objects within the video stream. To address this issue, we introduce the ***Object-Centric Video Stabilization (OCViS)*** method, designed to stabilize video frames containing multiple objects relative to specific objects within the scene. We leverage an input video sequence δ, comprising N frames with dimensions $X \times Y$, and stabilize these frames with respect to an object of interest. For this purpose, we derive a bounding box (x, y, w, h) around the object in the first frame δ_1, from which we extract an informative patch θ as our reference for stabilization. Note that x, y, w, and h denote x-location, y-location, width and height of the bounding box, respectively, For each frame i in the sequence, we compute the cross-correlation C between the reference patch θ and the

Algorithm 1 OCViS: **O**bject-**C**entric **Vi**deo **S**tabilization

Require: δ: Input video with N frames with each frame of size $X \times Y$, γ: Threshold

$(x, y, w, h) \leftarrow$ select_box_coordinates(δ_1) {Initial bounding box}
$\theta \leftarrow \delta_1(x : x + w, y : y + h)$ {Template patch}
$\delta_C \leftarrow \emptyset$ {Stabilized video}
for $i = 1$ **to** N **do**
  $\beta_x, \beta_y \leftarrow 0, 0$ {Motion vector initialization}
  for $j = 1$ **to** $X - w + 1$ **do**
    for $k = 1$ **to** $Y - h + 1$ **do**
      $\delta_{i_{wh}} \leftarrow \delta_i(j : j + w, k : k + h)$ {Current patch}
      $\overline{\delta}_{i_{wh}} \leftarrow \frac{1}{w \times h} \sum_{W=1}^{w} \sum_{H=1}^{h} \delta_i(W, H)$ {Mean of current patch}
      $\overline{\theta} \leftarrow \frac{1}{w \times h} \sum_{W=1}^{w} \sum_{H=1}^{h} \theta(W, H)$ {Mean of template patch}
      $I \leftarrow \delta_{i_{wh}} - \overline{\delta}_{i_{wh}}$ {Intensity difference}
      $T \leftarrow \theta - \overline{\theta}$ {Template difference}
      $C \leftarrow I \cdot T$ {Cross-correlation}
      $C_d, I_{sd}, T_{sd} \leftarrow 0, 0, 0$ {Initialize variables for similarity measure}
      for $W = 1$ **to** w **do**
        for $H = 1$ **to** h **do**
          $C_d \leftarrow C_d + C(W, H)$ {Sum of cross-correlation}
          $I_{sd} \leftarrow I_{sd} + I^2(W, H)$ {Sum of squared intensity differences}
          $T_{sd} \leftarrow T_{sd} + T^2(W, H)$ {Sum of squared patch differences}
        end for
      end for
      $\Gamma \leftarrow \frac{C_d}{\sqrt{I_{sd}}\sqrt{T_{sd}}}$ {Similarity measure}
      if $\Gamma > \gamma$ **then**
        $(\beta_x, \beta_y) \leftarrow (j, k)$
      end if
    end for
  end for
  $\beta_{x'}, \beta_{y'} \leftarrow \beta_x - x, \beta_y - y$ {Compute relative displacement}
  for $j = 1$ **to** X **do**
    for $k = 1$ **to** Y **do**
      $\delta_{i'}(j, k) \leftarrow \delta_i(j - \beta_{x'}, k - \beta_{y'})$ {Stabilize current frame}
    end for
  end for
  $\delta_C \leftarrow \delta_C \cup \{\delta_{i'}\}$ {Add stabilized frame to output}
end for

current frame patch $\delta_{i_{wh}}$, defined as follows:

$$C = \sum_{w=1}^{W} \sum_{h=1}^{H} (\delta_{i_{wh}} - \overline{\delta}_{i_{wh}}) \cdot (\theta - \overline{\theta}), \tag{1}$$

where $\overline{\delta}_{i_{wh}}$ and $\overline{\theta}$ represent the mean intensity values of the current patch and the reference patch, respectively. Next, a similarity measure Γ is computed as the normalized cross-correlation coefficient:

$$\Gamma = \frac{C}{\sqrt{I_{sd}}\sqrt{T_{sd}}}, \tag{2}$$

where I_{sd} and T_{sd} denote the sums of squared intensity differences of the current patch and its mean, and the reference patch and its mean, respectively. The computed value of Γ is then compared with a predefined threshold γ, and the motion vectors β_x and β_y are adjusted accordingly if Γ exceeds this threshold. These computed motion vectors are subsequently utilized to stabilize the current frame, aligning the object of interest with the reference template. A detailed algorithm outlining Object-Centric Video Stabilization, is provided in Algo. 1.

Algorithm 2 OFTrAnS: **O**ptical **F**low-based **Tr**ansformation **A**ccumulatio**n** and **S**moothing

Require: Object stabilized video δ with N frames, Window radius R

$T \leftarrow \emptyset$ {Initialize trajectory}
for $i = 2$ **to** N **do**
  $(dx, dy, da) \leftarrow$ **OFC**(δ_{i-1}, δ_i) {Compute optical flow}
  $T(i) \leftarrow (dx, dy, da)$ {Store transformation}
end for
$S_T \leftarrow$ zeros(N) {Initialize smoothed trajectory}
for $i = 1$ **to** N **do**
  $R_{start} \leftarrow \max(1, i-R)$ {Calculate starting frame index for window}
  $R_{end} \leftarrow \min(N, i+R)$ {Calculate ending frame index for window}
  $n \leftarrow R_{end} - R_{start} + 1$ {Number of frames in the window}
  sum $\leftarrow (0, 0, 0)$ {Initialize sum of transformations}
  for $j = R_{start}$ **to** R_{end} **do**
    sum $\leftarrow$ sum $+ T(j)$ {Accumulate transformations}
  end for
  $S_T(i) \leftarrow \frac{\text{sum}}{n}$ {Compute average transformation}
end for
for $i = 1$ **to** N **do**
  $N_T \leftarrow T(i) + (S_T(i) - T(i))$ {Update transformation}
  Apply N_T to frame δ_i
end for

B. Shakiness Removal for Video Stabilization

Although object-centric video stabilization has been achieved, addressing the shakiness in the background resulting from both object-centric stabilization and inherent camera shakiness remains necessary. Therefore, we utilize ***O**ptical **F**low-based **Tr**ansformation **A**ccumulatio**n** and **S**moothing **(OFTrAnS)*** for improving video stabilization. The algorithm employs optical flow techniques to compute the displacement vectors (dx, dy, da) between consecutive frames in a video sequence δ, capturing the motion of objects within the frames. *OFTrAnS* utilizes pairs of consecutive frames i and $i+1$ in the sequence, where optical flow technique computes the displacement vectors (dx, dy, da), representing translation and rotation between the frames. These vectors are stored in a trajectory matrix T, where each row corresponds to the transformation for a specific frame. To smooth out the transformations and reduce abrupt changes in stabilization, we employ a window-based approach. For each frame i, a window of radius R is defined, spanning frames $i - R$ to $i + R$. The transformations within this window are averaged to obtain a smoothed transformation for the current frame. Mathematically, let $T(i)$ represent the transformation vector for frame i, and $S_T(i)$ represent the smoothed transformation. The smoothed transformation $S_T(i)$ is computed as the average of the transformations within the window:

$$S_T(i) = \frac{1}{2R+1} \sum_{j=i-R}^{i+R} T(j) \tag{3}$$

Finally, the smoothed transformations $S_T(i)$ are applied to the respective frames in the video sequence, resulting in a stabilized output. A detailed algorithm is provided in Algo. 2. Note that the details of **OFC** and **CompA** are provided below.

1) Optical Flow Calculation (OFC): We propose an algorithm (Algo. 3) for computing optical flow vectors (dx, dy, da) between consecutive images I_1 and I_2, utilizing the Lucas-Kanade method with an affine model. In particular, the algorithm takes as input the images I_1 and I_2, a window size w, and a threshold T. For each pixel (x, y) in I_1, we compute the horizontal gradient I_x, vertical gradient I_y, and temporal gradient I_t, given by:

$$I_x = \frac{\partial I_1}{\partial x},$$
$$I_y = \frac{\partial I_1}{\partial y},$$
$$I_t = I_1(x, y) - I_2(x, y).$$

We initialize matrices A and b as zero matrices. Then, for each pixel (x', y') in a window of size w centered at (x, y), we compute spatial gradients ΔI_x, ΔI_y, and temporal gradient ΔI_t. Matrices A and b are updated as follows:

$$A \leftarrow A + \begin{bmatrix} \Delta I_x^2 & \Delta I_x \Delta I_y \\ \Delta I_x \Delta I_y & \Delta I_y^2 \end{bmatrix},$$
$$b \leftarrow b + \begin{bmatrix} \Delta I_x \Delta I_t \\ \Delta I_y \Delta I_t \end{bmatrix}.$$

After computing A and b for all pixels in the window, we solve the linear system $Ax = b$ to obtain the optical flow displacement vector $x = \begin{bmatrix} dx \\ dy \end{bmatrix}$. Additionally, we compute the change in angle da using the **CompA** algorithm (Algo. 4)

with parameters (I_1, I_2, w). If the magnitude of the optical flow vector $||(dx, dy)||$ exceeds the threshold T, we update the optical flow vector (dx, dy, da) at pixel (x, y). This iterative process is repeated for every pixel in I_1, resulting in the computation of optical flow vectors for the entire image.

Algorithm 3 OFC: **O**ptical **F**low **C**alculation using Lucas-Kanade Method

Input: Image I_1, Image I_2, Window size w, Threshold T
Output: Optical flow vectors (dx, dy, da)
for each pixel (x, y) in I_1 **do**
  $I_x \leftarrow$ Horizontal gradient of I_1 at (x, y)
  $I_y \leftarrow$ Vertical gradient of I_1 at (x, y)
  $I_t \leftarrow I_1(x, y) - I_2(x, y)$
  Initialize A, b as zero matrices
  for each pixel (x', y') in a window of size w centered at (x, y) **do**
    $\Delta I_x \leftarrow I_x(x', y')$
    $\Delta I_y \leftarrow I_y(x', y')$
    $\Delta I_t \leftarrow I_t(x', y')$
    $A \leftarrow A + \begin{bmatrix} \Delta I_x^2 & \Delta I_x \Delta I_y \\ \Delta I_x \Delta I_y & \Delta I_y^2 \end{bmatrix}$
    $b \leftarrow b + \begin{bmatrix} \Delta I_x \Delta I_t \\ \Delta I_y \Delta I_t \end{bmatrix}$
  end for
  Solve $Ax = b$ for $x = \begin{bmatrix} dx \\ dy \end{bmatrix}$
  $da \leftarrow$ **CompA**(I_1, I_2, w)
  if $||x|| > T$ **then**
    Update the optical flow vector (dx, dy, da) at pixel (x, y)
  end if
end for

2) Computing Angle Change (CompA): In this section, we present a method (Algo. 4) for computing the change in angle da between two consecutive images I_1 and I_2 using the Lucas-Kanade algorithm with an affine model. More specifically, given images I_1 and I_2 and a window size w, the algorithm aims to estimate the affine transformation parameters that best align a window W centered at each pixel (x, y) in I_1 with the corresponding region in I_2. To begin, for each pixel (x, y) in I_1, we compute the spatial gradients I_x and I_y using finite differences:

$$I_x(x, y) = \frac{\partial I_1}{\partial x},$$
$$I_y(x, y) = \frac{\partial I_1}{\partial y}.$$

We then extract a window W of size $w \times w$ centered at (x, y) from I_1 and warp it using the previous estimate of the affine transformation parameters to align it with the corresponding region in I_2. Let W' denote the warped window. We then compute the error image $e = W - W'$ and optionally smooth it using Gaussian blur to obtain $\hat{e}$. The smoothing operation is beneficial for robustness against noise and outliers. Using the gradients I_x and I_y, we construct the Jacobian matrix J for the affine transformation at pixel (x, y). The Jacobian matrix J is given by:

$$J = \begin{bmatrix} \frac{\partial W'}{\partial p_1} & \frac{\partial W'}{\partial p_2} & \cdots & \frac{\partial W'}{\partial p_n} \end{bmatrix},$$

where $p_1, p_2, \ldots, p_n$ are the parameters of the affine transformation. Next, we compute the weighted sum of squared errors $H = J^T J$ and the gradient vector $g = J^T \hat{e}$. The parameter update Δp is obtained by solving the linear system $H\Delta p = g$. Finally, we update the affine transformation parameters using the update Δp and compute the change in angle da from the updated affine transformation matrix. This iterative process is performed for each pixel (x, y) in I_1, resulting in the estimation of the change in angle for the entire image sequence.

Algorithm 4 CompA: **Comp**ute change in **A**ngle da using Lucas-Kanade with Affine Model

Require: Image I_1, Image I_2, Window size w
$A \leftarrow I$ {Initialize as identity matrix for affine transformation}
for each pixel (x, y) in I_1 **do**
  Compute gradient I_x and I_y at (x, y) in I_1
  Extract window W of size w centered at (x, y) from I_1
  Warp W using previous estimate of affine transformation to align with I_2
  Compute error image $e = W - Warp(I_2, A)$
  Compute weighted error image $\hat{e} =$ Gaussian_Blur(e) {Optional: Smooth the error image}
  Construct Jacobian matrix J for affine transformation at (x, y)
  Compute matrix $H = J^T J$
  Compute vector $g = J^T \hat{e}$
  Solve system of equations $H\Delta p = g$ for Δp
  Update affine transformation matrix A using Δp
  Compute change in angle da from the updated affine transformation matrix
end for

IV. Experimental Details

A. Implementation Details

Due to the absence of available datasets specifically designed for object-centric video stabilization [29]–[31], we have developed our own dataset comprising both real-world and synthetically generated shakiness. The synthetic shakiness levels are set at intensities of $5, 10, 15, 20, 25$, and 30. To evaluate the effectiveness of our approach, we measure the average difference and standard deviation in the location of object bounding boxes across frames for object stabilization, referred to as *average object error*. Additionally, to assess shakiness removal, we calculate *average motion error*, in which we derive the mean optical flow transformations across frames along with their standard deviation. We conduct testing on a total of 70 videos, with 10 videos representing each synthetic shake intensity, alongside 10 real-world shaky videos. Note that frame size of the videos is 224×224.

TABLE I: COMPARISON RESULTS WITH SHAKINESS REMOVAL METHOD (SRM) [28].

Syn. Shaky	Avg. Object Error ↓		Avg. Motion Error ↓	
	SRM [28]	Ours	SRM [28]	Ours
5	61.75 ± 14.11	5.76 ± 4.17	1.09 ± 0.43	0.92 ± 0.28
10	62.38 ± 14.20	5.93 ± 3.33	1.24 ± 0.43	1.04 ± 0.29
15	83.17 ± 14.27	6.75 ± 4.00	1.41 ± 10.46	1.26 ± 0.27
20	64.11 ± 14.22	7.61 ± 4.53	1.64 ± 0.50	1.44 ± 0.41
25	60.65 ± 22.75	7.15 ± 4.70	1.90 ± 0.62	1.66 ± 0.40
30	65.88 ± 14.37	7.79 ± 5.20	2.16 ± 0.71	1.85 ± 0.47
Real Shaky	17.73 ± 17.81	3.32 ± 2.73	0.91 ± 0.90	0.66 ± 0.69

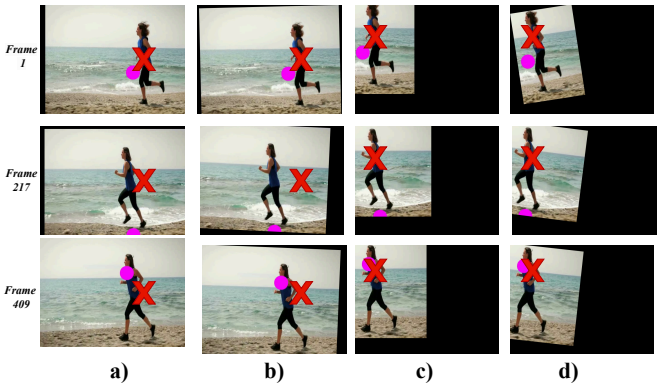
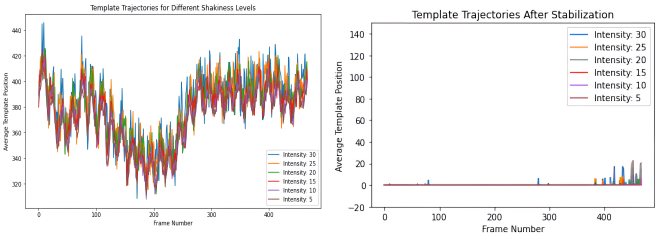


Fig. 3: Comparative results on real-world video: a) Original video frames. b) Results obtained from conventional shakiness removal method [28]. c) Results from our OCViS approach. d) Results obtained from our proposed algorithm (combination of OCViS and OFTrAnS). Note that × should be aligned across columns to ensure object (human) stability.



(a) Before stabilization. (b) After stabilization.

Fig. 4: Comparison of average object trajectory.

B. Results

Fig. 3 and 4 illustrate the effectiveness of our approach in stabilizing objects in multi-object video scenes. More specifically, in Fig. 3, we show that while conventional video stabilization techniques remove shakiness [28], they often fail to keep object in consistent positions (unaligned red crosses). OCViS, however, stabilizes object (indicated by aligned red crosses) while still leaving some shakiness. Our method not only stabilizes objects but also removes shakiness, resulting in perceptually effective outcomes.

In addition, Fig. 4 illustrates the smooth trajectory of object achieved using our method across a range of synthetic shakiness levels, from 5 to 30. This is attained through simultaneous object-centric stabilization and shakiness removal.

For establishing quantitative efficacy, we show the performance of our approach across videos with various synthetic shakiness levels (syn. shaky in Table I) and real shake videos in Table I. It is worth noting that we achieve improvement by a large margin in comparison to a standard stabilization technique [28]. For example, 88.18% ($65.88 \rightarrow 7.79$) and 14.35% ($2.16 \rightarrow 1.85$) improvement is obtained in average object and motion errors, respectively. Our method facilitates object-centric video stabilization followed by shakiness removal leading to an improved overall performance. Note that unlike our approach, existing works [27], [30] often center on deep learning-based methods, and primarily focus on shakiness removal rather than object stability. Consequently, there are no common grounds for comparison between our method and these existing approaches.

We also conduct an ablation study (Table II) highlighting the contributions of OCViS and OFTrAnS. OCViS enhances object stability but may introduce shakiness, while OFTrAnS effectively removes shakiness but may neglect object stability. Integrating both ensures optimal performance, combining object stability and shakiness removal.

TABLE II: ABLATION STUDY ON OCViS AND OFTrAnS.

OCViS	OFTrAnS	Avg. Object Error ↓	Avg. Motion Error ↓
✓		0.48 ± 0.83	1.28 ± 0.77
	✓	67.11 ± 15.34	0.97 ± 0.48
✓	✓	7.30 ± 4.56	0.81 ± 0.32

V. Conclusion

In this paper, we proposed a novel end-to-end method, combining ***O**bject-**C**entric **Vi**deo **S**tabilization (**OCViS**)* and ***O**ptical **F**low-based **Tr**ansformation **A**ccumulatio**n** and **S**moothing (**OFTrAnS**)* algorithms, for object-centric multi-object video stabilization. Our approach effectively detects, tracks, and stabilizes specific objects while reducing shakiness in each frame. Moreover, the proposed approach ensures smooth and steady footage, even in complex scenarios, without relying on computationally intensive deep neural networks or ground-truth data, making it practical for real-world applications. For future work, we aim to enhance the reliability of our method in scenarios with highly varying illumination environments, which currently represents a limitation of our approach.

REFERENCES

[1] C. Deng, S. He, Y. Han, and B. Zhao, "Learning dynamic spatial-temporal regularization for uav object tracking," *IEEE Signal Processing Letters*, vol. 28, pp. 1230–1234, 2021.

[2] H. Tiwari, V. K. Kurmi, V. K. Subramanian, and Y. S. Chen, "Distilling knowledge for occlusion robust monocular 3d face reconstruction," *Image and Vision Computing*, vol. 137, p. 104763, 2023.

[3] L. Kong, D. Huang, J. Qin, and Y. Wang, "A joint framework for athlete tracking and action recognition in sports videos," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 2, pp. 532–548, 2019.

[4] Y. Wang, Q. Huang, and Z. Miao, "Video stabilization: A comprehensive survey," *Neurocomputing*, vol. 516, 2023. Published on 1 November 2022.

[5] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. Published on 05 June 2006.

[6] S. S. Hussain, M. Nazir, S. K. Sharma, and M. A. R. Khan, "A robust video stabilization algorithm for global motion estimation using block matching," *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, 2020. Published on 14 January 2020.

[7] W. Guilluy, L. Oudre, and A. Beghdadi, "Video stabilization: Overview, challenges and perspectives," *Signal Processing: Image Communication*, vol. 90, p. 116015, 2021. Published in July 2021.

[8] W. G. Aguilar and C. Angulo, "Real-time model-based video stabilization for microaerial vehicles," *Neural Process Lett*, vol. 43, pp. 459–477, 2016. Published on 02 June 2015.

[9] Y. H. Chen, H. Y. S. Lin, and C. W. Su, "Full-frame video stabilization via sift feature matching," in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 361–364, IEEE, 2014.

[10] S. Erturk, "Multiplication-free one-bit transform for low-complexity block-based motion estimation," *IEEE Signal Processing Letters*, vol. 14, no. 2, pp. 109–112, 2007.

[11] C. Jia and B. L. Evans, "Constrained 3d rotation smoothing via global manifold regression for video stabilization," *IEEE Transactions on Signal Processing*, vol. 62, no. 13, pp. 3293–3304, 2014.

[12] C.-H. Chu, "Video stabilization for stereoscopic 3d on 3d mobile devices," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2014.

[13] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 450–462, 2012.

[14] Q. Rao, X. Yu, S. Navasardyan, and H. Shi, "Sim2realvs: A new benchmark for video stabilization with a strong baseline," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5406–5415, 2023.

[15] T. Liu, G. Wan, H. Bai, X. Kong, B. Tang, and F. Wang, "Real-time video stabilization algorithm based on superpoint," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[16] W. Zhao, X. Li, Z. Peng, X. Luo, X. Ye, H. Lu, and Z. Cao, "Fast full-frame video stabilization with iterative optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23534–23544, 2023.

[17] W. Yan, Y. Sun, W. Zhou, Z. Liu, and R. Cong, "Deep video stabilization via robust homography estimation," *IEEE Signal Processing Letters*, 2023.

[18] C. Huang, X. Pan, J. Cheng, and J. Song, "Deep image registration with depth-aware homography estimation," *IEEE Signal Processing Letters*, vol. 30, pp. 6–10, 2023.

[19] S. K. Kumawat, M. Biswas, S. Chinara, A. K. Tripathy, K. C. Li, J. P. Sahoo, and A. K. Mishra, "Srgan with 3d cnn model for video stabilization," *Advances in Distributed Computing and Machine Learning*, vol. 660, 2023. Published on 28 June 2023.

[20] J. Yu and R. Ramamoorthi, "Learning video stabilization using optical flow," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. Published on 05 August 2020.

[21] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Trans. Graph.*, vol. 28, August 2009. Published in August 2009.

[22] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, "Video stabilization with a depth camera," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–95, 2012. Published on 26 July 2012.

[23] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," *ACM Trans. Graph.*, vol. 31, August 2012. Published in August 2012.

[24] Z. Zhao and X. Ma, "Video stabilization based on local trajectories and robust mesh transformation," in *IEEE International Conference on Image Processing (ICIP)*, pp. 4092–4096, 2016. Published on 19 August 2016.

[25] F. Hu, J. Ma, L. Shen, and H. Du, "Digital video stabilization based on multilayer gray projection," *Signal Processing: Image Communication*, vol. 68, 2018. Published on 10 July 2018.

[26] M. Sharif, S. Khan, T. Saba, M. Raza, and A. Rehman, "Improved video stabilization using sift-log polar technique for unmanned aerial vehicles," in *IEEE International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–7, 2019. Published on 16 May 2019.

[27] Y. Xu, J. Zhang, S. J. Maybank, and D. Tao, "Dut: Learning video stabilization by simply watching unstable videos," *IEEE Transactions on Image Processing*, 2022. Published on 20 June 2022.

[28] A. Spannbauer, "Vidstab: A Python library for video stabilization." https://github.com/AdamSpannbauer/python_video_stab, Sep 13, 2021. A standard python library.

[29] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM transactions on graphics (TOG)*, vol. 32, no. 4, pp. 1–10, 2013.

[30] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, and S.-M. Hu, "Deep online video stabilization with multi-grid warping transformation learning," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2283–2292, 2018.

[31] J. Yu and R. Ramamoorthi, "Selfie video stabilization," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 551–566, 2018.