An Enhanced 3D Rainy Stereoscopic Video Stabilization Framework by Using Depth Estimation and SA-LMCCA-CNN

Mehala Rajendran
Department of Computer Applications
Alagappa University
India
mehalaa.researchscholar@gmail.com

Mahesh Krishnan
Department of Computer Applications
Alagappa University
India
mahesh.alagappa@gmail.com

Abstract: *The process of enhancing the video's quality by removing unwanted effects of camera shakes and jitters is called Video Stabilization (VS). However, the 3-Dimensional (3D) rainy stereoscopic video stabilization process was not concentrated on any of the prevailing research work. Therefore, in this framework, an effective 3D rainy stereoscopic video stabilization with depth estimation and Shape Autotuning Liebovitch map Cheetah Chase Algorithm with Convolution Neural Network (SA-LmCCA-CNN) is proposed. Primarily, the input videos are converted into a number of frames. After that, by using Pairnorm L0 Gradient Minimization (Pn-LGM), the raindrops in each frame are removed. Later, the overlapping region and depth estimation are processed, and by using the Liebovitch map Cheetah Chase Algorithm (LmCCA), the energy function is diminished. Likewise, to mitigate the hallucination issue, a mesh is generated by utilizing Alternating Least Squares-Locally Constrained Representations (ALS-LCR). Then, from the hallucination-mitigated image and energy function minimized image, the feature points are extracted. Later, by employing SA-LmCCA-CNN, the stable and unstable frames are classified. If the frame is unstable, then the frame undergoes motion and camera path corrections, followed by raindrop reconstruction; otherwise, raindrop reconstruction is done directly for a stable frame. Lastly, in order to get the stabilized video, the frames are synthesized. The experimental analysis proved the proposed model's robustness in 3D rainy stereoscopic video stabilization by attaining a stability score of 0.93.*

Keywords: *Video stabilization, 3D rainy stereoscopic video, depth estimation, hallucination mitigation, texture mapping, LKT, and LPF.*

Received May 19, 2025; accepted November 17, 2025
https://doi.org/10.34028/iajit/23/1/14

1. Introduction

The growing popularity of videos plays a major role in daily life in the new digital era [22]. The generation of video content achieves a dominant performance with the improvements of digital cameras, smartphones, and some other portable video recording devices [19]. However, owing to the instability of camera movements, the video content might suffer from some visual distortions [9]. Blurring effects, poor video quality, and visual discomfort are caused by this instability [5]. An effective process used for the mitigation of these video instability issues is Video Stabilization (VS) [4]. A technique utilized to diminish the effects of unwanted camera movements for enhancing the video's quality is called VS [33]. VS has the advantage that the unwanted camera movement effects can be effectively removed without disturbing the intentional camera movements [20]. Thus, to remove the high-frequency distortions in numerous applications like security surveillance, filmmaking, videography, surgical procedures, wildlife observations, and so on, VS is essential [26].

Nowadays, a surge in demand for high-quality video experiences are caused by the rapid growth multimedia technology with elevated stereoscopic video content [14]. Still, it is very challenging to capture stable and visually effective stereoscopic videos in some dynamic conditions [31]. Hardware solutions like gimbals and stabilizers, software solutions like motion post-capture correction algorithms, and some hybrid solutions that combine both hardware and software solutions were used by the traditional VS techniques [11, 18]. Nevertheless, these techniques often struggle to eliminate complex motion artifacts and are less effective in handling depth inconsistencies in dynamic conditions [2].

For VS, numerous methodologies were analyzed and developed in the prevailing works. Versatile Quaternion-based Filter algorithm, Principal Component Analysis (PCA), trajectory optimization, Euclidean Distance-Constrained Randomized Sampling Consistency (ED-RSC) algorithm, decomposed motion compensation, gyroscopic measurement, Bezier curve, simple linear iterative clustering, K-means clustering, and mesh generation models are some of the prevailing

models for VS centered on hardware approaches, software approaches, and hybrid approaches [7, 17]. Still, none of these models focused on stabilizing the 3D rainy stereoscopic video. Hence, this framework proposes effective 3D rainy stereoscopic VS by using depth estimation and Shape Autotuning Liebovitch map Cheetah Chase Algorithm with Convolution Neural Network (SA-LmCCA-CNN).

1.1. Problem Statement

Some drawbacks of the prevailing works are explained as follows:

- None of the existing works concentrated on stabilizing the 3D rainy stereoscopic videos.
- The prevailing works had frames' misregistration owing to the overlapping of Wide baseline and low-texture regions in the stereoscopic videos.
- The existing [25] had hallucination problems because of the independent processing of the 2 frame views.
- The mixing motion estimation and path smoothing on the Red, Green, and Blue (RGB) video frames resulted in estimation errors, wobbling, and distortion artifacts in [29].
- The conventional works had few content losses owing to image cropping.

1.2. Objectives

To overcome the limitations of the prevailing works, the major contributions of the proposed framework are presented below:

- In this framework, 3D rainy stereoscopic VS is performed by detecting and removing the raindrops in the frames by utilizing Pn-LGM.
- Here, the overlap region and depth are estimated by employing the Multi-view Stereo (MS) methodology, and the texture mapping is processed using Texture Filtering (TF).
- The hallucination issue is detected by generating mesh and is mitigated by using the proposed ALS-LCR.
- The estimation error is diminished by employing the Lucas Kanade-Triparametric correlation coefficient (LKT), and the camera path smoothing is done by using Low-Pass Filtering (LPF).
- The video is synthesized without any cropping, thereby eliminating the content loss in this model.

The remainder of this paper is organized as follows: the related works is explained in section 2, the proposed framework is described in section 3, a performance analysis is provided in section 4, and finally, section 5 concludes the paper with future work.

2. Related Works

Xu *et al*. [32] established a learning VS system centered on Deep Unsupervised Trajectory (DUT) by simply watching unstable videos. By employing a multi-homography estimation strategy, the key point motions were estimated in the trajectory estimation stage. After that, in the trajectory smoothing stage, dynamic smoothing kernels were utilized. As per the analysis outcomes, the developed model performed better than the other top-notch models both qualitatively as well as quantitatively. Nevertheless, in this work, the transformations betwixt images could not be computed accurately.

Wu *et al*. [30] introduced an enhanced K-means clustering and super pixels-centric VS with robust global motion estimation. In this work, to establish the motion vector space, the super pixels' feature points between 2 adjacent frames were estimated. By using homography transformation, the global motion was determined. The experimental outcomes stated that when compared with the conventional models, the developed model achieved a superior structural similarity index. Nevertheless, mixing motion estimation and path smoothing on the RGB video frames resulted in estimation errors, wobbling, and distortion artifacts in this model.

Souza and Pedrini [25] presented visual rhythms for qualitative evaluation of VS. In this, from the average of the frames' columns, a vertical visual rhythm was constructed, and from the rows of the frames, a horizontal visual rhythm was created. The evaluation outcomes proved that the developed methodology significantly verified the complex movements in the visual rhythms. However, due to independently processing the 2 frame views, this model had hallucination issues.

Zhao and Ling [35] illustrated an adaptively meshed methodology for the shaky video's stabilization centered on the feature trajectories and an adaptive blocking strategy. Initially, centered on the distribution of the feature trajectories in the frames, the triangle mesh was generated. Then, to stabilize the video, the transformation between the shaky frames and their stabilized views over the mesh was analyzed. The robustness of this model was stated in the analysis outcomes. Still, this model had a higher computational burden.

Milanovic [15] stated a Gyroscope-centered VS for the electro-optical long-range surveillance systems. In this, by employing the quaternion domain interpolation, the unwanted motion estimation quality was enhanced. In addition, the gyro bias instability and noise disturbance were eliminated. As per the evaluation outcomes, the developed model had lower computational complexity. Nevertheless, this model's performance was fully centered on the gyroscope technology.

Zhang *et al*. [34] propounded a high-precision satellite VS methodology centered on the Euclidean Distance-constrained Randomized Sampling Consistency (ED-RanSaC) operator. The introduced

model extracted the high-precision features effectively and matched the homologous points to render reliable stabilized video. The dynamic performance of this model in VS was proved by the analysis outcomes. Nevertheless, for synthetic aperture radar video, this model attained poor stabilization efficiency.

Wu *et al*. [29] presented a simultaneous VS and rolling shutter removal process for shaky videos. In this, for the spatially variant inter-frame motion estimation, a neighbor-motion-aware local motion model was employed. After that, to enhance the neighbor motion consistency, a classical mesh-centric model was utilized. The experimental outcomes stated that when compared with other models, this model attained superior performance in VS. However, this model had an issue in the estimation of correct correspondences in blurry and severe noise videos.

Lee [10] depicted a low memory access VS for a low-cost camera system-on-chip. For the video stabilization of the low-end hardware devices, structure and layout methods were developed in this model. The evaluation results proved that when compared with other straightforward methods, this model significantly diminished the memory access amount. Nevertheless, for low-cost camera hardware-based video stabilization, this model was not appropriate.

Lin *et al*. [12] employed a content-and-disparity-aware stereoscopic video stabilization methodology. Primarily, the rotational angle between 2 adjacent frames was determined. By rotating the frames appropriately, the inconsistent angular velocity was solved. As per the analysis results, this model optimized the smoothness and preserved the video contents effectively. However, for quick translation videos, this model had unsatisfactory outcomes.

Luan *et al.* [13] signified an unsupervised video stabilization algorithm centered on key point detection. For the generation of the rich key points, a Deep Neural Network (DNN) based key point detector was developed. In addition, to obtain the unstable motion trajectories, the foreground and background separation approach was employed. Nevertheless, due to image cropping, this model had a content loss issue.

Oliveira *et al*. [19] presented the depth perception and visualization in 360° of each donor. In this perception was noted especially when visualizing donors with different cavities and fossae. The combination of 3D techniques is of paramount importance for neuroanatomy education. Stereoscopic projections could provide a valuable tool for neuroanatomy instruction directed at clinical trainees and could be especially useful when access to laboratory-based learning is limited.

Morichon *et al*. [16] utilizing the smartphone- based 360° photogrammetry, virtual camera recording, and stereoscopic display. The results demonstrate that the 3D models obtained feature a complete mesh with a high level of detail and a realistic texture. Additionally, stereoscopic animations were both feasible and effective in enhancing depth perception. The simplicity and affordability of this method position it as a technique of choice for creating easily photorealistic anatomical models combined with stereoscopic depth visualization.

Chen *et al*. [3] complete 3D-DIC simulation method involving optical simulation and mechanical simulation and integrating 3D-DIC, virtual stereo vision, and image super-resolution reconstruction technology. Virtual stereo vision can reduce hardware costs and eliminate camera-synchronization errors. Image super-resolution reconstruction can compensate for the decrease in precision caused by image-resolution loss.

Pathak *et al*. [21] presented a method for accurate all-round 3D reconstruction of an indoor environmentin one-shot using a system of trinocular 360-degree cameras. Binocular 360-degree stereo is unableto reconstruct in all directions due to lack of disparity along epipolar directions. Thus, a third cameraalong a perpendicular epipolar direction is introduced to cover for this, making the system trinocular.

Al Mokhtar and Dawwd [1] presented VP based on decreasing design complexity and producing good results. This produces the VP model based on a Three-Dimensional Variational Auto Encoder (3D VAE). presented to builds all layers depending on 3D convolutional layers. This leads to better extraction of spatiotemporal information and decreases the design complexity. The Kullback Leibler Loss (KL Loss) is enhanced by a 3D sampling stage which allows to calculation of the 3D latent loss. The 3D sampling represents a good regularizer in the model.

3. 3D Video Stabilization Model

In this framework, by using Pn-LGM and SA-LmCCA-CNN, the 3D rainy stereoscopic video is stabilized. Primarily, the input video is converted into a number of frames, followed by removing raindrops in each frame. After that, to classify the unstable and stable frames, the raindrop-removed frames are further processed. Lastly, the raindrops are reconstructed, resulting in a stabilized video. Figure 1 indicates the proposed framework's structure.

3.1. Frame Conversion

Primarily, the input video data Ψ is taken from the dataset. Here, Ψ is converted into a number of image frames and is explained as:

$$\Psi_{fra} = \{\Psi_1, \Psi_2, \dots, \Psi_F\} \quad (1)$$

Here, the converted frames are denoted as Ψ_{fra}, and the number of frames in Ψ_{fra} is represented as F.

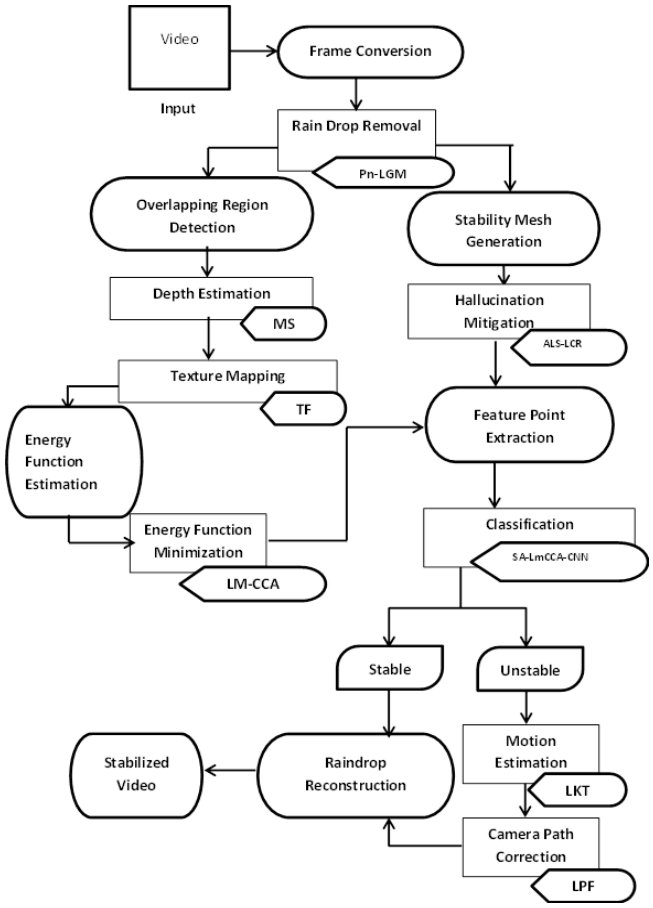


Figure 1. Architecture of the proposed 3D video stabilization model.

3.2. Raindrop Removal

After that, raindrops are removed from Ψ_{fra} for ensuring a clearer view of the scene, which is crucial for video stabilization. The raindrops can generate false motion cues, causing poor stabilization outcomes. Therefore, by utilizing Pn-LGM, the raindrops are removed. An edge-aware image smoothing methodology that aims to remove small-scale oscillations while preserving salient structures is termed conventional L_0 Gradient Minimization (LGM). Nevertheless, the major edges are sharpened by the LGM, which might inadvertently blur or weaken details. Hence, Pairnorm is used in LGM to overcome the issues. The objective function of Pn-LGM to reduce the non-zero gradients in Ψ_{fra} is formulated as:

$$\Psi_{rain} = \min_{f} \hat{\gamma} \left\| f - \Psi_{fra} \right\|^2 + \lambda \|\nabla f\|_0 \tag{2}$$

Where, the raindrop removed frames are denoted as Ψ_{rain}, $\|.\|$ and $\|.\|_0$ depict the L_2 norm and L_0 norm, respectively, the filtered output is represented as f, the gradient of f is depicted as ∇f, the control parameter for controlling the level of sparseness in f is denoted as λ, and $\hat{\gamma}$ denotes the Pairnorm vector, which is depicted as:

$$\hat{\gamma} = \dot{s}\sqrt{g} \cdot \frac{\Psi_{fra}}{\sqrt{\left\|\Psi_{fra}\right\|_{\tilde{F}}^2}} \tag{3}$$

Here, the pairwise distance is represented as g, the hyperparameter is denoted as $\dot{s}$, and the Frobenius norm is depicted as $\tilde{F}$. If the λ is higher, then the output is obtained as a coarser result with a lesser gradient. After that, the objective function is written as:

$$\Psi_{rain} = \min_{f} \hat{\gamma} \sum_{i=1}^{I} \left\| f_i - \Psi_{fra} \right\|^2 + \frac{\lambda}{2} \sum_{j \in J_i} \left\| f_i - f_j \right\|_0 \tag{4}$$

Here, the pixels are denoted as i and j, the number of lengths of f is depicted as I, and the neighboring set of the i^{th} pixel is represented as J_i. Here, $\frac{\lambda}{2}$ denotes the relationship between f_i and f_j and is counted twice. Therefore, the non-zero gradients (i.e., raindrops) in Ψ_{fra} are removed.

3.3. Overlap Region Detection

Then, centered on the pixel values, position, and coordinates of the pixels in the frames, the overlapping regions betwixt each frame are detected from Ψ_{rain}. The detected overlapped regions are signified as Ψ_{over}.

3.4. Depth Estimation

Then, by using the MS method, the depth of the detected Ψ_{over} is assessed for determining how deeply the regions overlap. Here, the pixel coordinates (P) of Ψ_{over} are signified as:

$$c_m(P) = s_m D_m(P) \tilde{P} \tag{5}$$

Where, the 3D pixel coordinates of the m^{th} frame are denoted as c_m, the per-frame scale coefficients are depicted as s_m, the depth map is represented as D_m, and the homogeneous augmented pixel coordinate is denoted as $\tilde{P}$. After that, the 3D coordinates are pointed into the camera coordinate system of another frame (n), and it is articulated as:

$$c_{m \to n}(P) = \xi_n \Re_n^{\tilde{T}} (\Re_m \xi_m^{-1} c_m(P) + \tau_m - \tau_n) \tag{6}$$

Here, the intrinsic, rotation, and translation of frames are represented as ξ, $\Re$ and τ, respectively, and the transpose of the frame rotation vector is denoted as $\tilde{T}$. After that, the objective function (o) of MS is to diminish the re-projection loss (Γ^{rep}) of every pixel in Ψ_{over}, and it is signified as:

$$O = arg \min_{\theta^{depth}} \sum_{(m,n) \epsilon P} \sum_{P} \Gamma_{m \to n}^{rep}(P) \tag{7}$$

Here, θ^{depth} denotes the optimization variable. After that, Γ^{rep} is determined as:

$$\Gamma_{m \to n}^{rep}(P) = \Gamma_{m,n}^{sim}(c_{m \to n}(P), c_n\left(\tilde{f}_{m \to n}(P)\right)) \tag{8}$$

Where, the re-projection similarity loss is represented as $\Gamma_{m,n}^{sim}$ and the flow re-projection is notated as $\tilde{f}_{m \to n}(P)$. Therefore, the depth of overlap area is obtained in O.

3.5. Texture Mapping

Later, by using TF, the texture of O is mapped. While mapping the textures of O in 3D by employing a

parametric function (Θ), TF enhances the quality of the frames and is explained as:

$$\theta: O \rightarrow \varpi \tag{9}$$

Here, the 3D Euclidean space is symbolized as ϖ. After that, Ψ_{text} represents the 3D texture-mapped image.

3.6. Energy Function Estimation

Afterward, to determine the weighted terms of the pixels in Ψ_{text}, the energy function of Ψ_{text} is estimated. By using the sparse displacement and similarity transformation factors, the weighted terms are estimated. The estimated weighted terms are depicted as Ψ_{ene}.

3.7. Energy Function Minimization

After that, to evaluate the deviation between the frames, the energy function Ψ_{ene} is minimized by employing Lm-CCA. Since the Cheetah Chase Algorithm (CCA) has awesome capacities in the exploitation and exploration phases, it is chosen. CCA conducts exploration and exploitation separately, which means that it doesn't need to balance the 2 forces. Nevertheless, CCA's drawback is that overemphasizing exploration may hinder convergence, making it challenging to identify the global optimum. The Liebovitch map technique is included in the exploration phase to solve this issue. Therefore, the proposed methodology is named as Lm-CCA. Based on searching for prey, sitting and waiting, and attacking strategies, this method is carried out. Here, to avoid getting stuck in local optimal points, leaving the prey and returning home strategy is also introduced. The process of proposed Lm-CCA is explained as follows:

1. Initialization*:* primarily, the populace of Cheetah (*C*) (i.e., energy function of Ψ_{ene}) is initialized as:

$$C = \{C1, C2, \ldots, Cz\} \tag{10}$$

Here, the number of cheetahs in the population is denoted as *z*. After that, the fitness function (ε) for the cheetah's position is deemed as the minimum (*min*) energy (*l*), and it is illustrated as:

$$\varepsilon = min\{l\} \tag{11}$$

Then, the cheetah's position in the search space (*d*) is estimated as:

$$C_{t,a}^{d} = v_{lo}^{d} + \varepsilon \cdot rand(v_{up}^{d} - v_{lo}^{d}) \tag{12}$$

Where, the position of the a^{th} cheetah in *d* at t^{th} hunting time is denoted as $C_{t,a}^{d}$, and the upper and lower bounds of *d* are notated as v_{up}^{d} and v_{lo}^{d}, respectively.

2. Searching strategy: grounded on the surrounding environmental conditions and hunting behavior, the position of the a^{th} cheetah in the search space (*d*) is estimated and is mathematically represented as:

$$C_{t+1,a}^{d} = C_{t,a}^{d} + (\hat{\vartheta}_{t,a}^{-1} \cdot a_{t,a}^{d}) \tag{13}$$

Here, the new position of the a^{th} cheetah is denoted as $C_{t+1,a}^{d}$, the randomization parameter and step length are depicted as $\hat{\vartheta}_{t,a}^{-1}$ and $a_{t,a}^{d}$, respectively, and the next hunting time is represented as *t*+1. The randomization parameter is just a normally distributed random number. The step length is articulated as:

$$a_{t,a}^{d} = 0.001 \times \frac{t}{t_{max}}(v_{up}^{d} - v_{lo}^{d}) \tag{14}$$

Here, the length of hunting time is notated as t_{max}. Based on the distance of the a^{th} cheetah and arbitrarily selected e^{th} cheetah, the random step length is estimated for other members in *C* and is articulated as:

$$a_{t,a}^{d} = 0.001 \times \frac{t}{t_{max}}(C_{t,a}^{d} - C_{t,e}^{d}) \tag{15}$$

3. Sitting and waiting strategy: then, the cheetah's sit-and-wait strategy to get the prey close enough for hunting is arithmetically modeled as:

$$C_{t+1,a}^{d} = C_{t,a}^{d} \tag{16}$$

4. Attacking strategy: then, based on speed and flexibility to catch the prey, the cheetah's attacking strategy is determined. Centered on the fleeing prey and the leader's position, the position of the cheetah is adjusted and is signified as:

$$C_{t+1,a}^{d} = C_{t,a}^{d} \cdot \hat{\theta} + \breve{\vartheta}_{t,a} \cdot X_{t,a}^{d} \tag{17}$$

Where, the turning factor and interaction factor are denoted as $\breve{\vartheta}_{t,a}$ and $X_{t,a}^{d}$, respectively, and the Liebovitch mapping vector is depicted as $\hat{\theta}$ and is described as:

$$\hat{\theta} = \begin{cases} \sigma C_{t,a}^{d}, if\ 0 < C_{t,a}^{d} \leq dy_1 \\ \dfrac{dy_2 - C_{t,a}^{d}}{dy_2 - dy_1}, else\ if\ dy_1 < C_{t,a}^{d} \leq dy_2 \\ 1 - \sigma(1 - C_{t,a}^{d}), else\ dy_2 < C_{t,a}^{d} \leq 1 \end{cases} \tag{18}$$

Here, the control parameter is symbolized as σ, and the additional dynamic parameters are depicted as dy_1 and dy_2. The interaction factor determines the interaction betwixt the cheetah and leader in the capturing mode and is determined as:

$$\breve{\vartheta}_{t,a} = |\vartheta_{t,a}| exp\left(\frac{\vartheta_{t,a}}{2}\right) \sin(2\pi\vartheta_{t,a}) \tag{19}$$

Here, the randomly chosen value from the normal distribution is symbolized as $\vartheta_{t,a}$. In addition, the turning factor assesses the sharp turns of the cheetahs in the capturing mode and is signified as:

$$X_{t,a}^{d} = C_{t,e}^{d} - C_{t,a}^{d} \tag{20}$$

To attain the best solution, the position of the cheetah is updated centered on the above strategies. Sometimes, to prevent the system from falling into local optimal solutions, leave the prey and return to home strategy is applied. Therefore, the energy function is diminished, and it is notated as Ψ_{min}. The pseudo-code for Lm-CCA is presented in Algorithm (1) below.

Algorithm 1: Pseudo-code for Lm-CCA.

Input: Weighted terms (Ψ_{ene})
Output: Minimized weighted terms (Ψ_{min})
Begin
Initialize Cheetah population (C),ε, maximum iteration (t_{max}), d , v_{up}^{d} and v_{lo}^{d}
While ($t \leq t_{max}$)
For each Ψ_{ene} do
Evaluate fitness function (ε)
$C_{t,a}^{d} = v_{lo}^{d} + \varepsilon \cdot rand(v_{up}^{d} - v_{lo}^{d})$
Estimate
 # Searching strategy
$C_{t+1,a}^{d} = C_{t,a}^{d} + (\hat{\vartheta}_{t,a}^{-1} \cdot a_{t,a}^{d})$
Update
Determine step length
 # Sitting and waiting strategy
Formulate $C_{t+1,a}^{d} = C_{t,a}^{d}$
 # Attacking strategy
Update the position
Compute the Liebovitch map
If ($0 < C_{t,a}^{d} \leq dy_1$)
{
Obtain $\sigma C_{t,a}^{d}$
} *Else if* ($dy_1 < C_{t,a}^{d} \leq dy)_2$
{
Attain $\frac{dy_2 - c_{t,a}^{d}}{dy_2 - dy_1}$
} *Else*
{
Get $1\text{-}\sigma(1\text{-}C_{t,a}^{d})$
}
End if
Evaluate turning factor and interaction factor
Estimate leave the prey and go back home strategy
End For
End while
Return minimized weighted terms(Ψ_{min})
End

After that, for the identification of stable and unstable frames, the features of Ψ_{min} are employed further.

3.8. Stability Mesh Generation

Likewise, to enhance the reliability rate of the video stabilization, the hallucination is removed from Ψ_{rain}. For Ψ_{rain}, a stability mesh is generated to mitigate the image hallucination. Here, Ψ_{mesh} signified the generated mesh.

3.9. Hallucination Mitigation

After that, by employing ALS-LCR, the hallucination effect is reduced from Ψ_{mesh}. To mitigate the hallucination, the conventional Locally Constrained Representations (LCRs) are utilized. This is because LCR encourages the pixel representations to be predictable by representing neighboring pixels. Nevertheless, enforcing locality constraints can be challenging in LCR when dealing with irregular or sparse data. Therefore, to overcome this issue, Alternating Least Squares (ALS) is employed for reducing the sparse data in LCR. Initially, to generate the locality constraints (L) for Ψ_{mesh}, the ALS is employed and is articulated as:

$$L_{x,y} = \min_{(x,y)\in\hat{R}} \wp'^{\left(\hat{R}_{x,y} - \Psi_{mesh,x}^{T}\Psi_{mesh,y}\right)^{2}} + \lambda(\left\|\Psi_{mesh,x}\right\|_{\tilde{F}}^{2} + \left\|\Psi_{mesh,y}\right\|_{\tilde{F}}^{2}) \quad (21)$$

Where, the observed rating of the data points (x, y) in Ψ_{mesh} is denoted as $\hat{R}$, the transpose vector is depicted as T, the Frobenius norm is represented as $\tilde{F}$, and the regularization parameter is denoted as λ. Here, to reduce the sparse data, the ALS vector ($\wp'$)is employed and is signified as:

$$\wp' = (\Psi_{mesh}^{T}\Psi_{mesh} + \lambda\ddot{I})^{-1}\Psi_{mesh}^{T} \quad (22)$$

Here, the identity matrix is signified as $\ddot{I}$. Later, to represent L, the diagonal ($\tau_{\tilde{d}}$) and Laplacian matrix ($\tau_{\tilde{l}}$) are generated, and it is depicted as:

$$\tau_{\tilde{d}} = \sum_{x,y} L_{x,y} \quad (23)$$

$$\tau_{\tilde{l}} = \tau_{\tilde{d}} - L_{x,y} \quad (24)$$

Later, to preserve the local structure of Ψ_{mesh}, the objective function ($\wp$) of ALS-LCR is determined and articulated as:

$$\wp = \min_{L} \sum_{x,y} \left\|\Psi_{mesh,x} - \Psi_{mesh,y}\right\|^{2} L_{x,y} \quad (25)$$

Therefore, by choosing the eigenvectors corresponding to the smallest eigenvalues of $\wp$, the final representation of the hallucination-mitigated image is obtained. Thus, the ALS-LCR ensures that the learned representation preserves the local geometric structure of Ψ_{mesh} to diminish the hallucination effects in Ψ_{mesh}. Therefore, Ψ_{hall} depicts the hallucination mitigated image.

3.10. Feature Points Extraction

Later, features like scale-invariant feature transform, speeded-up robust features, homogeneity, contrast, entropy, and angular second moment are extracted from Ψ_{min} and Ψ_{hall}. To estimate the motion of the camera, these feature points in every single frame are tracked over successive frames. Here, λ_{ext} denotes the extracted feature points.

3.11. Classification

By using SA-LmCCA-CNN, the stable and unstable frames are classified centered on the feature points(λ_{ext}). The conventional CNN can process high-dimensional data and share information betwixt layers. However, since CNN has several processing layers, it takes a longer time for the training process. Therefore, shape autotuning activation is used in CNN to overcome the drawback, and by using Liebovitch map Cheetah Chase Algorithm (LmCCA), the weight is optimized. Figure 2 presents the SA-LmCCA-CNN's structure.

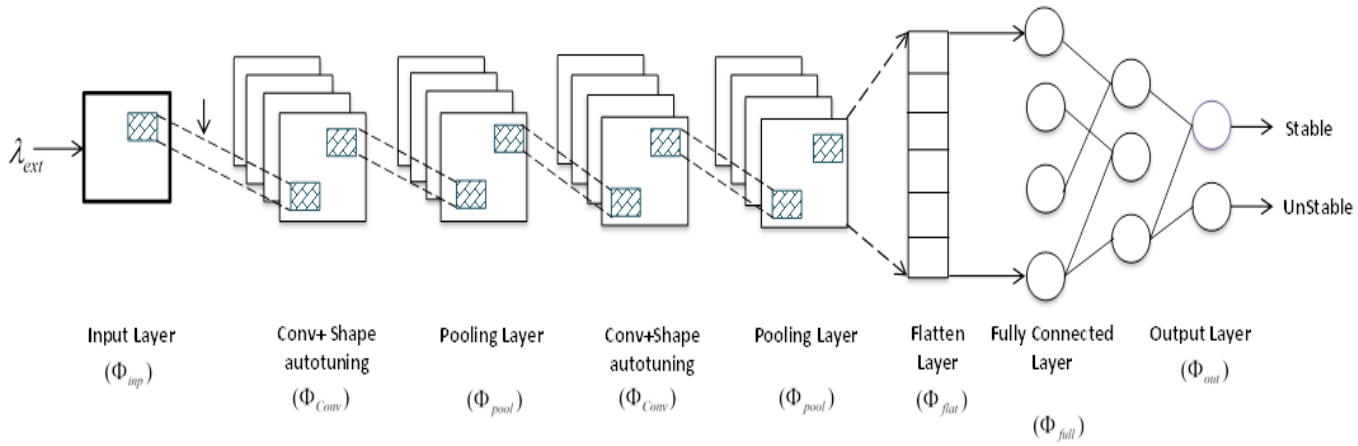


Figure 2. Structure of SA-LmCCA-CNN.

1. Input layer: the feature points (λ_{ext}) are fed as input to the input layer (Φ_{inp}) of SA-LmCCA-CNN, and it is explained as:

$$\Phi inp \rightarrow \{\lambda\, ext\} \tag{26}$$

After that, the output of Φ_{inp} is fed as input to the convolutional layer (Φ_{conv}).

2. Convolutional layer: Φ_{conv} is accountable for the feature extraction by utilizing some kernels, which are convolved with Φ_{inp} for capturing the relevant features of Φ_{inp}, and it is articulated as:

$$\Phi_{conv} = \wedge \cdot \delta\left(\frac{\Phi_{inp} + 2ps - y}{\varsigma}\right) + 1 \tag{27}$$

Here, the convolutional padding size is signified as *ps*, the convolutional kernel size is notated as γ, and the convolutional stride size is symbolized as ς. Later, the shape autotuning activation function (δ) is given as:

$$\delta = \frac{\Phi_{inp}}{\frac{\Phi_{inp}}{a"} + exp(-\frac{\Phi_{inp}}{\beta"})} \tag{28}$$

Here, a pair of trainable non-negative parameters is depicted as a" and β". Also, the weights ($\wedge$) of SA-LmCCA-CNN are determined by the LmCCA function, which is briefly explained in section 3.7.

3. Pooling layer: afterward, in the pooling layer (Φ_{pool}), the spatial dimension of Φ_{conv} is diminished by retaining the significant information from Φ_{conv} and discarding the unwanted information. Therefore, the $\Phi_{pool}\Phi_{pool}$ operation is articulated as follows:

$$\Phi_{pool} = \max_{\varsigma}(\Phi_{conv}) \tag{29}$$

4. Flatten layer: the flatten layer (Φ_{flat}) aids the network in learning the complex patterns from Φ_{pool}. The Φ_{flat} converts the feature maps from Φ_{pool} to a format that is understood by the fully connected layer (Φ_{ful}).

$$\mathcal{Q}\Phi pool \rightarrow \Phi flat \rightarrow \tilde{\mathcal{Q}}\Phi flat \tag{30}$$

Here, the multi-dimensional and one-dimensional feature arrays are represented as $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$, respectively.

5. Fully connected layer: The fully connected layer (Φ_{flat}) utilizes the features from $\tilde{\mathcal{Q}}\Phi_{flat}$ and predicts the output classes. Then, from Φ_{flat}, the feature information is connected to the output layer via Φ_{ful}.
6. Output layer: In the output layer (Φ_{out}), the corresponding outcomes with the pre-trained labels for the input data are obtained. The SA-LmCCA-CNN's output is obtained as the stable and unstable frames. The output attained from the proposed SA-LmCCA-CNN is depicted as:

$$\Phi_{out} = \{\Phi_1, \Phi_2\} \tag{31}$$

Here, the stable and unstable frames are denoted as Φ_1 and Φ_2, respectively. Lastly, the reliability level of the predicted outputs and the true labels of the input are quantified by the loss function ($\Diamond$). The pseudo-code for SA-LmCCA-CNN is given Algorithm (2) below.

Algorithm 2: Pseudo-code for SA-LmCCA-CNN.

Input: Features points (λ_{ext})
Output: Stable or unstable frames (Φ_{out})
Begin
Initialize padding size (ps), kernel size (γ), and stride size (ς)
For each λ_{ext} *do*
Compute convolutional operation
Activate $\delta = \frac{\Phi_{inp}}{\frac{\Phi_{inp}}{a"} + exp\left(-\frac{\Phi_{inp}}{\beta"}\right)}$
Determine weights
Formulate pooling function
Evaluate
$\mathcal{Q}\Phi_{pool} \rightarrow \Phi_{flat} \rightarrow \tilde{\mathcal{Q}}\Phi_{flat}$
Estimate fully connected layer (Φ_{ful})
Obtain output
Evaluate $\Diamond$
If $\left(\Diamond = \min\right)$ *{*
Terminate
} Else {
Adjust the parameters
}
End If
End For
Return Stable or unstable frames
End

The frames are processed and synthesized centered on the classification results for generating stabilized video.

3.12. Motion Estimation

If Φ_2 is obtained, then the video is stabilized by assessing the motion in Φ_2 utilizing LKT. A widely utilized differential methodology for optical flow estimation is the existing Lucas-Kanade method. By using the least squares criterion, this method solves the fundamental optical flow equations for all the pixels in the neighborhood. However, the algorithm may converge to suboptimal solutions when image pairs have significant distortions or complex motions. In this method, the Triparametric correlation coefficient is included to solve this issue. The process of LKT is illustrated as follows:

- The LKT assumes that the displacement between the nearby frames (*M* and *N*) is small with the window size (*ws*) of pixels (*k*) for the motion estimation. Afterward, the optical flow matrix ($\hbar$) holds all pixels (*S*) of coordinates within *ws* and is signified as:

$$\tilde{I}_M(S)\hbar_M + \tilde{I}_N(S)\hbar_N = c\tilde{I}_{\hat{t}}(S) \tag{32}$$

Where, the intensity at position (*M*, *N*) is denoted as $\tilde{I}$ and the time is signified as $\hat{t}$.

- The optical flow matrix provides the linear equation and is articulated as:

$$\varphi\hbar = \beta \tag{33}$$

Here, the matrix that contains the image gradient components is denoted as φ, the vector that represents the optical flow is represented as $\hbar$, and the image derivative vector is notated as β. These vectors are signified as:

$$\varphi = \hat{\Gamma}\begin{bmatrix} \tilde{I}_M(S_1)\,\tilde{I}_N(S_1) \\ \tilde{I}_M(S_2)\,\tilde{I}_N(S_2) \\ \tilde{I}_M(S_k)\,\tilde{I}_N(S_k) \end{bmatrix} \tag{34}$$

$$\hbar = \begin{bmatrix} \hbar_M \\ \hbar_N \end{bmatrix} \tag{35}$$

$$\beta = \begin{bmatrix} \tilde{I}_{\hat{t}}(S_1) \\ \tilde{I}_{\hat{t}}(S_2) \\ \tilde{I}_{\hat{t}}(S_k) \end{bmatrix} \tag{36}$$

Here, $\hat{\Gamma}$ implies the Triparametric correlation coefficient vector and is illustrated as:

$$\hat{\Gamma} = \frac{\nabla(M,N)}{\sqrt{\omega(M)}\sqrt{\omega(N)}} \tag{37}$$

Where, the correlation betwixt *M* and *N* is depicted as $\nabla(M, N)$ and $\sqrt{\omega(M)}$ and $\sqrt{\omega(N)}$ signify the informational energies of *M* and *N*, respectively.

- Here, the LKT estimates the least square solution to avoid over-determination and is articulated below:

$$\varphi^{Tr}\varphi\hbar = \varphi^{Tr}\beta \tag{38}$$

Where, the transpose of φ is denoted as φ^{Tr}.

- Thus, the least square solution for the optical flow of Φ_2 is obtained as:

$$\hbar = (\varphi^{Tr}\varphi)^{-1}\varphi^{Tr}\beta \tag{39}$$

$$\begin{bmatrix} \hbar_M \\ \hbar_N \end{bmatrix} = \begin{bmatrix} \sum_{\bar{\imath}=1}^{k} \tilde{I}_M(S_{\bar{\imath}})^2 & \sum_{\bar{\imath}=1}^{k} \tilde{I}_M(S_{\bar{\imath}})\,\tilde{I}_N(S_{\bar{\imath}}) \\ \sum_{\bar{\imath}=1}^{k} \tilde{I}_N(S_{\bar{\imath}})\,\tilde{I}_M(S_{\bar{\imath}}) & \sum_{\bar{\imath}=1}^{k} \tilde{I}_N(S_{\bar{\imath}})^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{\bar{\imath}=1}^{k} \tilde{I}_M(S_{\bar{\imath}})\,\tilde{I}_{\hat{t}}(S_{\bar{\imath}}) \\ -\sum_{\bar{\imath}=1}^{k} \tilde{I}_N(S_{\bar{\imath}})\,\tilde{I}_{\hat{t}}(S_{\bar{\imath}}) \end{bmatrix} \tag{40}$$

If the central matrix $\varphi^{Tr}\varphi$ is invertible and well-conditioned, then the optical flow ($\hbar$) of Φ_2 is well-estimated.

3.13. Camera Path Correction

Later, to determine the motion of the camera, the camera path is estimated during video capture. The unwanted motions can easily be differentiated from the intentional camera movements by estimating the camera path. Therefore, it is estimated and corrected by employing the LPF, and it is described as follows:

$$\phi(\hbar^*) = \frac{1}{1 + \Im(\hbar^* \times 2\pi R'C')} \tag{41}$$

Where, the transfer function is denoted as $\emptyset$, the frequency of $\hbar$ is represented as $\hbar^*$, the imaginary unit is depicted as $\Im$, and the cutoff frequency with resistor (*R*') and capacitor (*C*') values is denoted as $2\pi R'C'$. Thus, $\hbar_{path}$ signifies the camera path corrected image.

3.14. Raindrop Reconstruction

Later, the raindrops are reconstructed for both Φ_1 and $\hbar_{path}$ to get the clear original frames for the input video. Lastly, to get the stabilized 3D stereoscopic video, the raindrop reconstructed frames are synthesized. The proposed 3D rainy stereoscopic video stabilization framework's performance is discussed further.

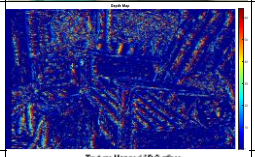
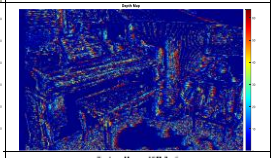
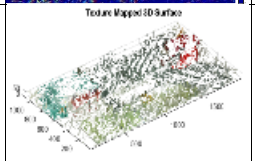
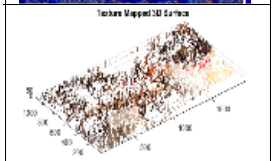
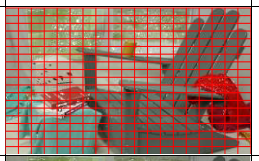
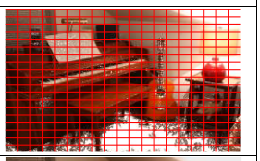
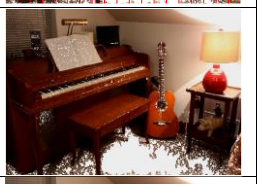
4. Result and Discussion

Here, the robustness of the proposed framework is proved by analogizing it with the prevailing works. The proposed work is implemented in the working platform of MATLAB.

4.1. Dataset Description

To evaluate the proposed system's efficacy, the Middlebury Stereo Dataset (MSD) is employed. The MSD has high-resolution stereo videos with complex geometry and pixel-accurate ground truth video data. It comprises 47 stereoscopic video data. Among them, 80% and 20% of the data are utilized for training and testing purposes, respectively. In Table 1, the sample image results attained by the proposed model are presented. The sample image results obtained from the proposed 3D rainy stereoscopic video stabilization framework are illustrated in Table 1.

Table 1. Sample image results.

S.no.	Process	Image 1	Image 2
1	Input		
2	Raindrop removal		
3	Depth estimation		
4	Texture mapping		
5	Stability mesh generation		
6	Hallucination mitigation		
7	Raindrop reconstruction		

4.2. Performance Analysis

To prove the superiority of the proposed models' video stabilization process, the proposed methodologies' performance is analyzed and compared with the prevailing models in this subsection.

4.2.1. Performance Evaluation of Proposed LKT

In this section, based on the performance metrics, such as the Sum of Squared Differences (SSD), Mean Absolute Difference (MAD), Sum of Absolute Difference (SAD), Peak Signal-to-Noise Ratio (PSNR), and Mean Squared Error (MSE), the proposed LKT's performance is analyzed and compared with the existing models like Lucas-Kanade (LK), Horn-Schunck (HS), Camshift Algorithm (CA), and Block Matching Algorithm (BMA).

The comparative analysis of the proposed LKT and the prevailing models based on performance metrics like SSD, MAD, SAD, and PSNR is indicated in Table 2. By employing the Triparametric correlation coefficient, the proposed LKT reduces the suboptimal solution. Hence, the SSD, MAD, SAD, and PSNR of the proposed model are 5082.13, 13.707, 8793.2, and 41.01, respectively. However, the existing models' average SSD, MAD, SAD, and PSNR were 7918.07, 27.096, 10143.1, and 37.59, respectively. Nevertheless, when analogized to the proposed model, the performances of the existing models were considerably lower. Hence, the analysis demonstrates the efficacy of the proposed LKT.

Table 2. Comparison analysis of proposed LKT.

Technique	SSD	MAD	SAD	PSNR
Proposed LKT	5082.13	13.707	8793.2	41.01
LK	6891.64	19.645	9112.3	40.56
HS	7213.78	23.697	9657.3	38.69
CA	8113.64	29.913	10564.9	36.98
BMA	9453.21	35.128	11237.7	34.15

Figure 3 exhibits the MSE analysis of the proposed and the prevailing models. The MSE of the existing LK was 0.01236, HS was 0.03751, CA was 0.0689, and BMA was 0.1364. But, the proposed approach uses a Triparametric correlation coefficient function with the gradient component. Hence, the proposed model's MSE is 0.0079, which is better compared to all existing models. Hence, the analysis results demonstrated that the proposed model is less error-prone.

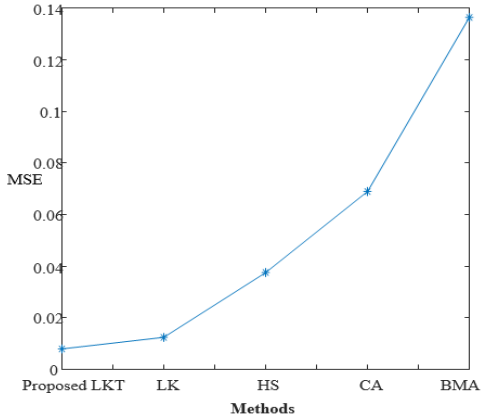
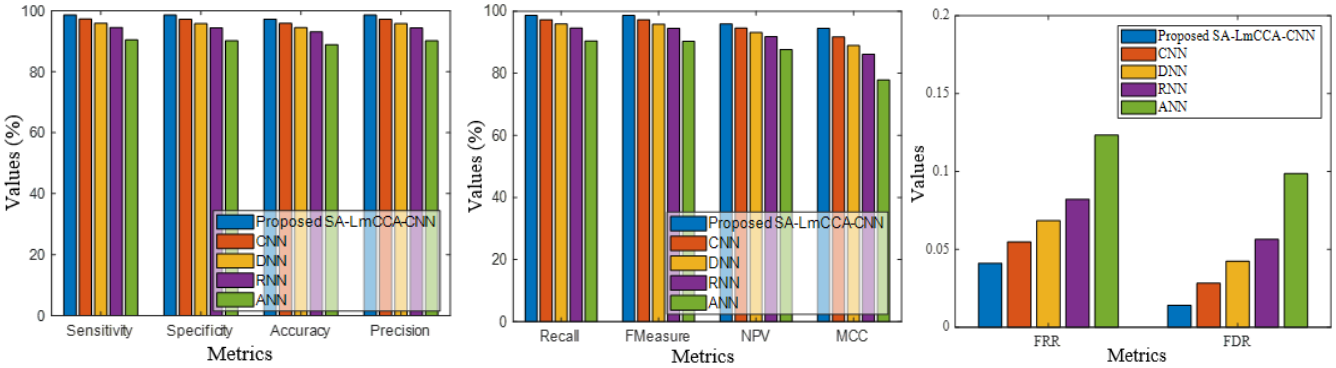


Figure 3. MSE analysis of proposed LKT.

4.2.2. Performance Evaluation of Proposed SA-LmCCA-CNN

In this phase, the performance analysis of the proposed SA-LmCCA-CNN and the prevailing models like Convolutional Neural Network (CNN), DNN, Recurrent Neural Network (RNN), and Artificial Neural Network (ANN) is performed. The analysis is centered on performance metrics, namely sensitivity, specificity, accuracy, precision, recall, F-measure, Negative Predictive Value (NPV), Matthews Correlation Coefficient (MCC), False Reject Rate (FRR), False Discovery Rate (FDR), False Positive Rate (FPR), and False Negative Rate (FNR).



a) Sensitivity, specificity, accuracy and precision. b) Recall, FMeasure, NPV and MCC. c) FRR and FDR

Figure 4. Performance assessment of proposed SA-LmCCA-CNN.

The performance assessment of the proposed SA-LmCCA-CNN and the prevailing models based on sensitivity, specificity, accuracy, precision, recall, F-measure, NPV, MCC, FRR, and FDR is depicted in Figures 4-a), (b) and (c). The proposed model attains the sensitivity, specificity, accuracy, precision, recall, F-measure, NPV, MCC, FRR, and FDR of 98.63%, 98.59%, 97.22%, 98.59%, 98.63%, 98.61%, 95.89%, 94.48%, 0.041, and 0.014, respectively. Nevertheless, the sensitivity, specificity, accuracy, precision, recall, F-measure, NPV, MCC, FRR, and FDR of the prevailing CNN was 97.26%, 97.18%, 95.83%, 97.18%, 97.26%, 97.22%, 94.52%, 91.70%, 0.054, and 0.028, respectively, which were considerably lower when analogized to the proposed model. Likewise, when analogized to the proposed model, other prevailing models also attained lower performances. Thus, the proposed model's robustness is proved. In Table 3, the FPR and FNR analysis of the proposed SA-LmCCA-CNN and the prevailing models are described. To reduce the training time of the network, the proposed SA-LmCCA-CNN effectively optimized the weights of the network and employed the shape autotuning activation function. Thus, the FPR and FNR analysis of the proposed SA-LmCCA-CNN are 0.014085 and 0.041096, respectively. Nevertheless, the existing DNN and RNN had FPRs of 0.042254 and 0.056338. Also, they had FNRs of 0.068493 and 0.082192, respectively. These performances are highly ineffective compared to the proposed model. In addition, other conventional models had lower performances. Therefore, from the analysis, the efficacy of the proposed SA-LmCCA-CNN is proven.

Table 3. FPR and FNR analysis.

Techniques	FPR	FNR
Proposed SA-LmCCA-CNN	0.014085	0.041096
CNN	0.028169	0.054795
DNN	0.042254	0.068493
RNN	0.056338	0.082192
ANN	0.098592	0.123288

4.2.3. Performance Analysis of Proposed Lm-CCA

Grounded on fitness vs. Iteration analysis, the performance of the proposed Lm-CCA and the prevailing CCA, Snow Leopard Optimization Algorithm (SLOA), Pelican Optimization Algorithm (POA), and Aquila Optimization (AO) is compared in this section.

Figure 5 compares the fitness vs. iteration analysis of the proposed and the prevailing methodologies. The fitness of the Lm-CCA is attaining minimized energy. For 10 iterations, the proposed model achieves a fitness of 0.5. This is owing to the utilization of the Liebovitch map for attaining global optimal solutions. Nevertheless, for 10 iterations, the existing CCA, SLOA, POA, and AO had the fitness of 0.59, 0.68, 1.39, and 1.9, respectively. These are considerably lower when analogized to the proposed model. Therefore, it is stated that the proposed model's efficacy is better than the existing models.

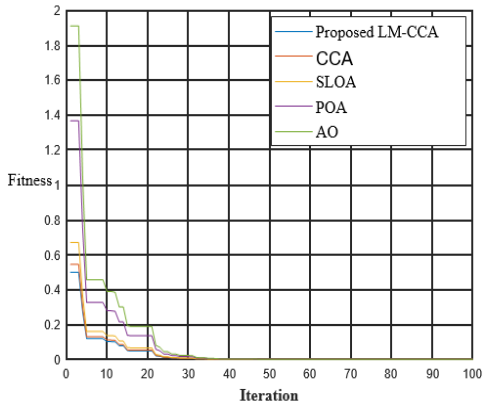


Figure 5. Fitness vs. iteration analysis.

4.2.4. Comparative Analysis with Related Works

The comparative analysis of the proposed methodology and related works is performed in this subsection. The comparative analysis with the related works is indicated in Table 4.

The comparative analysis of the proposed work and the related works in terms of different video stabilization techniques and datasets utilized in video stabilization is presented in Table 4. The robustness of the proposed work is determined by analogizing it with existing works based on stability score analysis. The existing (Huang *et al*. [6], Jang *et al*. [8], Shang and Chu [24], Valero *et al*. [27], and Ren *et al*. [23]) attained stability scores of 0.91, 0.82, 0.91, 0.92, and 0.86, respectively. Nevertheless, the proposed work's stability score is 0.93, which is considerably superior to other existing works. This dynamic performance of the proposed methodology is owing to the proper handling of the rainy stereoscopic video by determining the depth of the overlapped regions and mitigating the hallucination effects. In addition, the unstable frames are effectively processed by the proposed model. Therefore, this robust performance in video stabilization is attained by the proposed work. Thus, from the comparative analysis, the efficacy of the proposed work is demonstrated.

Table 4. Comparative analysis with related works.

Author name	Technique	Dataset	Stability score
Proposed	SA-LmCCA-CNN	MSD	0.93
Huang *et al*. [6]	Decomposed motion compensation	The National University of Singapore and DeepStab datasets	0.91
Jang *et al*. [8]	Dual-Modality Cross-Interaction	Inertial measurement unit sensor dataset	0.82
Shang and Chu [24]	Low-rank constraint and trajectory optimization	Video dataset	0.91
Valero *et al*. [27]	Thermal Infrared	Wildland fire video dataset	0.92
Ren *et al*. [23]	Gyroscope-based Self-Calibration Spherical	Gyroscope data	0.86

5. Conclusions and Future Work

In this framework, a robust 3D rainy stereoscopic video stabilization model is proposed by employing depth estimation and SA-LmCCA-CNN. The proposed model's efficacy was proved by implementing it in the working platform of MATLAB. As per the experimental analysis, the proposed methodology attained an accuracy of 97.22% and an F-measure of 98.61% for the classification of stable and unstable video frames, respectively. In addition, the proposed SA-LmCCA-CNN obtained the minimum FPR and FNR of 0.014085 and 0.041096, correspondingly. After that, the proposed model attained an SSD, MAD, SAD, and PSNR of 5082.13, 13.707, 8793.2, and 41.01, respectively. When analogized with the prevailing model, the proposed model's MSE was less, thus indicating that the model was less error-prone. Therefore, the proposed model's superiority was stated by the overall analysis.

Conclusion of the future work, the 3D rainy stereoscopic videos can be improved by the real-time climate data and advanced simulations. Weather effects like rain, snow and lighting can be made more realistic and dynamic.

Data Availability

The datasets are available at https://vision.middlebury.edu/stereo/data/scenes2014/zip/.

Reference

[1] Al Mokhtar Z. and Dawwd S., "3D VAE Video Prediction Model with Kullback Leibler Loss Enhancement," *International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 879-888, 2024. https://doi.org/10.34028/iajit/21/5/9

[2] Bellavia F., Fanfani M., Colombo C., and Piva A., "Experiencing with Electronic Image Stabilization and PRNU Through Scene Content Image Registration," *Pattern Recognition Letters*, vol. 145, pp. 8-15, 2021. https://doi.org/10.1016/j.patrec.2021.01.014

[3] Chen H., Li H., Liu G., and Wang Z., "A Novel Simulation Method for 3D Digital-Image Correlation: Combining Virtual Stereo Vision and Image Super-Resolution Reconstruction," *Sensors*, vol. 24, no. 13, pp. 1-26, 2024. https://doi.org/10.3390/s24134031

[4] Chen Y., Tseng, K., Lee Y., Chen, C., and Hung, Y., "PixStabNet: Fast Multi-Scale Deep Online Video Stabilization with Pixel-Based Warping," *in Proceedings of the International Conference on Image Processing*, Anchorage, pp. 1929-1933, 2021. https://doi.org/10.1109/ICIP42928.2021.9506801

[5] Guilluy W., Oudre L., and Beghdadi A. "Video Stabilization: Overview, Challenges and Perspectives," *Signal Processing: Image*

Communication, vol. 90, pp. 1-40, 2021. https://doi.org/10.1016/j.image.2020.116015

[6] Huang Q., Liu J., Jiang C., and Wang Y., "DMCVS: Decomposed Motion Compensation-Based Video Stabilization," *IET Image Processing*, vol. 18, no, 6, pp. 1422-1433, 2024. https://doi.org/10.1049/ipr2.13035

[7] Ito M. and Izquierdo E., "Deep Homography-Based Video Stabilization," *in Proceedings of the 25th International Conference on Pattern Recognition*, Milan, pp. 8819-8825, 2021. https://doi.org/10.1109/ICPR48806.2021.9413034

[8] Jang J., Ban Y., and Lee K., "Applied Sciences Dual-Modality Cross-Interaction-Based Hybrid Full-Frame Video Stabilization," *Applied Sciences*, vol. 14, no. 10, pp. 1-16, 2024. https://www.mdpi.com/2076-3417/14/10/4290#

[9] Keyang C., Shichao L., Lan R., Wenshan W., and et al., "Video Stabilization via Prediction with Time-Series Network and Image Inpainting with Pyramid Fusion," *Chinese Journal of Electronics*, vol. 30, no. 6, pp. 1103-1110, 2021. https://doi.org/10.1049/cje.2021.08.006

[10] Lee Y., "Low Memory Access Video Stabilization for Low-Cost Camera SoC," *Sensors*, vol. 22, no. 6, pp. 1-18, 2022. https://doi.org/10.3390/s22062341

[11] Lemaire P., Crispim-Junior C., Robinault L., and Tougne L., "Registering Unmanned Aerial Vehicle Videos in the Long Term," *Sensors*, vol. 21, no. 2, pp. 1-19, 2021. https://doi.org/10.3390/s21020513

[12] Lin S., Le T., Wu P., and Lee T., "Content-and-Disparity-Aware Stereoscopic Video Stabilization," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1545-1564, 2020. https://doi.org/10.1007/s11042-020-09767-9

[13] Luan Y., Han C., and Wang B., "An Unsupervised Video Stabilization Algorithm Based on Key Point Detection," *Entropy*, vol. 24, no. 10, pp. 1-15, 2022. https://doi.org/10.3390/e24101326

[14] Luchetti A., Zanetti M., Kalkofen D., and De Cecco M., "Stabilization of Spherical Videos Based on Feature Uncertainty," *Visual Computer*, vol. 39, no. 9, pp. 4103-4116, 2023. https://doi.org/10.1007/s00371-022-02578-z

[15] Milanovic P., Popadic, I., and Kovacevic B., "Gyroscope-Based Video Stabilization for Electro-Optical Long-Range Surveillance Systems," *Sensors*, vol. 21, no. 18, pp. 1-23, 2021. https://doi.org/10.3390/s21186219

[16] Morichon A., Dannhoff G., Barantin L., Destrieux C., and Maldonado I., "Doing More with Less: Realistic Stereoscopic Three-Dimensional Anatomical Modeling from Smartphone Photogrammetry," *Anatomical Sciences Education*, vol. 17, no. 4, pp. 864-877, 2024. https://doi.org/10.1002/ase.2402

[17] Morikawa C., Kobayashi M., Satoh M., Kuroda Y., and et al., "Image and Video Processing on Mobile Devices: A Survey," *Visual Computer*, vol. 37, no, 12, pp. 2931-2949, 2021. https://doi.org/10.1007/s00371-021-02200-8

[18] Nooralishahi P., Lopez F., and Maldague X., "A Drone-Enabled Approach for Gas Leak Detection Using Optical Flow Analysis," *Applied Science*, vol. 11, no. 4, pp. 1-19, 2021. https://doi.org/10.3390/app11041412

[19] Oliveira A., Leonel L., LaHood E., Nguyen T., and et al., "Projection of Realistic Three-Dimensional Photogrammetry Models Using Stereoscopic Display: A Technical Note," *Anatomical Sciences Education*, vol. 17, no. 1, pp. 39-46, 2024. https://doi.org/10.1002/ase.2329

[20] Pant S., Nooralishahi P., Avdelidis N., Genest M., and et al., "UAV-Based Active Infrared Thermography Application," *Sensors*, vol. 21, no. 5, pp. 1-16, 2021. https://www.mdpi.com/1424-8220/21/5/1604

[21] Pathak S., Hamada T., and Umeda K., "Trinocular 360-Degree Stereo for Accurate all-Round 3D Reconstruction Considering Uncertainty," *Advanced Robotics*, vol. 38, no. 15, pp. 1038-1051, 2024. https://doi.org/10.1080/01691864.2024.2376022

[22] Reichert G., Pieras M., Marroquim R., and Vilanova A., "Stabilization and Visual Analysis of Video-Recorded Sailing Sessions," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 26, pp. 1-10, 2021. https://doi.org/10.1186/s42492-021-00093-x

[23] Ren Z., Fang M., and Chen C., "Self-Calibration Spherical Video Stabilization Based on Gyroscope," *Information*, vol. 12, no. 8, pp. 1-13, 2021. https://doi.org/10.3390/info12080299

[24] Shang Z. and Chu Z., "Video Stabilization Based on Low-Rank Constraint and Trajectory Optimization," *IET Image Processing*, vol. 18, no. 7, pp. 1768-1779, 2024. https://doi.org/10.1049/ipr2.13062

[25] Souza M. and Pedrini H., "Visual Rhythms for qualitative evaluation of video stabilization," Eurasip Journal on Image and Video Processing, vol. 2020, no. 19, pp. 1-19, 2020. https://doi.org/10.1186/s13640-020-00508-4

[26] Taormina V., Raso G., Gentile V., Abbene L., and et al., "Segmentation in Videocapillaroscopy," *Sensors*, vol. 23, no. 18, pp. 1-23, 2023. https://doi.org/10.3390/s23187674

[27] Valero M., Verstockt S., Butler B., Jimenez D., and et al., "Thermal Infrared Video Stabilization for Aerial Monitoring of Active Wildfires," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2817-2832, 2021. https://doi.org/10.1109/JSTARS.2021.3059054

[28] Verma K., Ghosh D., and Kumar A., "Visual Tracking in Unstabilized Real Time Videos Using SURF," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 1, pp. 809-827, 2024. https://doi.org/10.1007/s12652-019-01249-7

[29] Wu H., Xiao L., and Wei Z., "Simultaneous Video Stabilization and Rolling Shutter Removal," *IEEE Transactions on Image Processing*, vol. 30, pp. 4637-4652, 2021. https://doi.org/10.1109/TIP.2021.3073865

[30] Wu R., Xu Z., Zhang J., and Zhang L., "Robust Global Motion Estimation for Video Stabilization Based on Improved K-Means Clustering and Superpixel," *Sensors*, vol. 21, no. 7, pp. 1-17, 2021. https://doi.org/10.3390/s21072505

[31] Xin Z., Kong S., Wu Y., Zhan G., and Yu J., "A Hierarchical Stabilization Control Method for a Three-Axis Gimbal Based on Sea-Sky-Line Detection," *Sensors*, vol. 22, no. 7, pp. 1-14, 2022. https://doi.org/10.3390/s22072587

[32] Xu Y., Zhang J., Maybank S., and Tao D., "DUT: Learning Video Stabilization by Simply Watching Unstable Videos," *IEEE Transactions on Image Processing*, vol. 31, no. 8, pp. 4306-4320, 2022. https://doi.org/10.1109/TIP.2022.3182887

[33] Yagoubi M., Amirshahi S., Moan S., Beghdadi A., and Rodner E., "Blind Video Stabilization Quality Assessment Based on Convolutional LSTM," *in Proceedings of the 11th European Workshop on Visual Information Processing*, Gjovik, pp. 1-6, 2023. https://doi.org/10.1109/EUVIP58404.2023.10323043

[34] Zhang F., Li X., Wang T., Zhang G., and et al., "High-Precision Satellite Video Stabilization Method Based on ED-RANSAC Operator," *Remote Sensing*, vol. 15, no. 12, pp. 1-17, 2023. https://doi.org/10.3390/rs15123036

[35] Zhao M. and Ling Q., "Adaptively Meshed Video Stabilization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 9, pp. 3504-3517, 2021. https://doi.org/10.1109/TCSVT.2020.3040753

Mehala Rajendran received her M.C.A degree in Alagappa University, Karaikudi, India in 2010 and M.Phil degree in Alagappa University, Karaikudi, India in 2013. She is currently pursuing Ph.D. at the Department of Computer Applications, Alagappa University, Karaikudi, India. Her areas of interest are Video Processing and Deep Learning Algorithms.

Mahesh Krishnan received the Ph.D. degree in Computer Science from Alagappa University, India in 2012. He is formerly professor and has 33 years of experience in the Department of Computer Applications, Alagappa University, Karaikudi, India. His research interest includes Video Processing Image Processing, and Cloud Security.