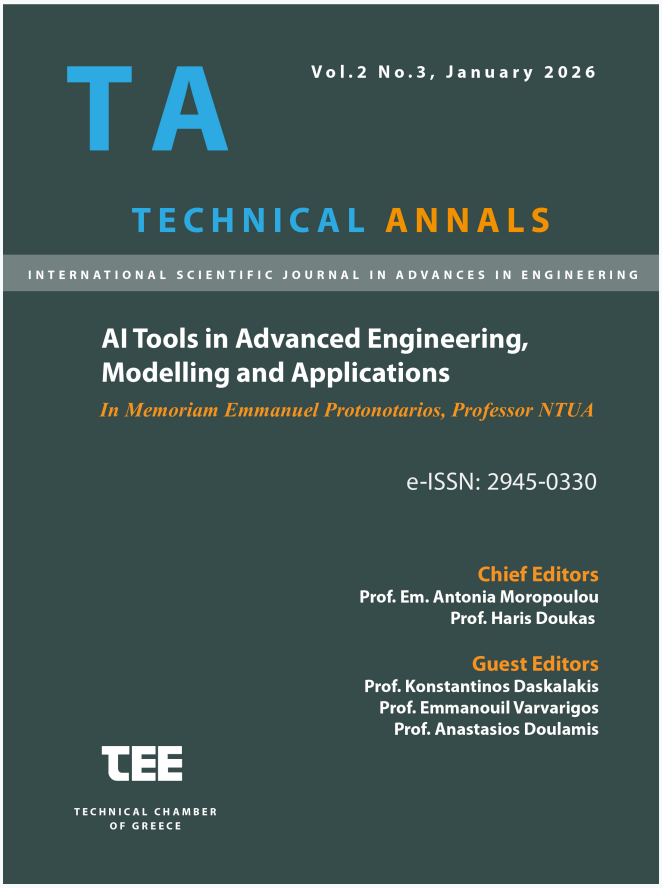
Technical Annals

Vol 2, No 3 (2026)

Technical Annals




# Image-to-Video Synthesis Using Generative Adversarial Networks and the Poisson Equation

*Christophoros Nikou, Michalis Vrigkas, Vasileios Stergiou*

doi: 10.12681/ta.45925



**To cite this article:**

Nikou, C., Vrigkas, M., & Stergiou, V. (2026). Image-to-Video Synthesis Using Generative Adversarial Networks and the Poisson Equation. *Technical Annals*, *2*(3). https://doi.org/10.12681/ta.45925

# Image-to-Video Synthesis Using Generative Adversarial Networks and the Poisson Equation

Vasileios Stergiou[1[0009-0009-7031-3559]], Michalis Vrigkas[2[0000-0001-5888-6949]], Christophoros Nikou[1[0000-0003-1388-6915]]

[1]Department of Computer Science & Engineering, University of Ioannina, Ioannina, Greece,
[2]Department of Communication and Digital Media, University of Western Macedonia, Kastoria, Greece,
vstergiou@uoi.gr, cnikou@uoi.gr, mvrigkas@uowm.gr

**Abstract.** Deep generative models have revolutionized video synthesis by enabling the animation of static source images through the extraction of motion patterns from driving sequences. Despite these advancements, a persistent challenge in the field remains the mitigation of structural degradation, where generated outputs frequently exhibit high-frequency frame noise and boundary artifacts that undermine visual quality. Adopting the First Order Motion Model (FOMM) as a baseline, we propose a post-processing pipeline that integrates the Poisson equation-based seamless cloning technique, guided by estimated occlusion maps and Gaussian filtering, to suppress synthesis noise and smooth edge discontinuities. To maintain temporal consistency and prevent the blending of background elements into the moving foreground, we further introduce a custom thresholding mask, as the main part in applying the Poisson-based technique. Our experimental results on the Taichi and Fashion datasets demonstrate that this approach preserves structural integrity more effectively than standard filtering, yielding substantial improvements in Structural Similarity (SSIM) metrics.

**Keywords:** Deep Learning, FOMM, Seamless Cloning, Video Reconstruction, Image Filtering

## 1 Introduction

Artificial neural networks are structurally modeled after the biological configuration of the human brain, serving as an algorithmic simulation of natural neural structures [12]. These systems mirror the brain's capacity to learn via examples under supervised or unsupervised learning paradigms. They process target responses, self-correct errors based on error propagation, retain information derived from historical inputs, and generalize behavior when presented with novel stimuli. This computational knowledge is stored across distributed network configurations via adjustable synapses known as weights. Reflecting these properties, a neural network organizes its units into discrete layers, including an input layer, an output layer, and intermediate hidden layers. The application of artificial neural networks is one of the most widespread methods for solving complex problem sets across modern advanced scientific fields, spearheaded

by breakthroughs in computer science, automated industrial manufacturing [1-4] and clinical medicine [5]. The continuous growth of these neural architectures stems from rapid hardware developments over recent decades, which have optimized processing speeds on large-scale parallel computing networks. Simultaneously, extensive research has broadened their application to non-traditional domains, resolving structural challenges that were previously computationally intractable [7]. A major subsection of these challenges involves tracking spatial and temporal changes within computer vision [8]. Specifically, transferring motion coordinates onto a static source image involves generating a sequence of frames where regional movements identified in a driving video are mapped directly onto the base asset. This process synthesizes completely new frames using the source image as a geometric foundation, aligning the internal motion vectors with those observed in the corresponding frame of the driving sequence. In digital image processing terminology, these complex spatial modifications are defined as affine transformations. To ensure a high-fidelity mapping of these localized movements onto the source target, deep learning networks are trained extensively on datasets consisting of human trajectories, ranging from macro-level skeletal animations to micro-level facial expressions. By utilizing specialized video sets containing complex bodily or facial kinematics, it is possible to generalize this motion mapping to abstract designs, artistic sketches, or non-human spatial layouts, such as the BAIR dataset used during the architectural development of advanced motion frameworks [7,8]. Generating video from a single static view remains a significant challenge within computer vision. It requires high-precision identification of target structures (e.g., face or body geometry) within 2D or 3D coordinate space, followed by the accurate extraction of motion patterns to ensure natural temporal transitions [9]. Modern architectures frequently address these problems using deep learning frameworks, such as generative adversarial networks (GANs) [10] and variational autoencoders (VAEs) [11]. These networks operate under supervised or self-supervised learning paradigms, mapping target conditions to ensure structural alignment. However, despite their capacity to transfer motion coordinates effectively, deep generative models often suffer from structural degradation, producing visible high-frequency frame noise and boundary artifacts. This paper investigates the optimization of the first order motion model (FOMM) framework through systematic post-processing [8]. Specifically, we investigate the application of the Poisson equation-based seamless cloning technique, alongside lowpass Gaussian filtering, to suppress synthesis noise and smooth edge discontinuities [20]. The purpose of our research is to propose an algorithmic approach on the post-processing step of the outputs videos, instead of a potential high-cost machine learning one, while aiming to maintain the same levels, or even improve, the metrics that the method was evaluated on the previous research material. Particularly, the resulting generations are quantitatively evaluated $L_1$ loss and structural similarity (SSIM) metrics. The structure of this paper is organized as follows: Section 2 explores the Monkey-Net architecture [7]. Section 3 breaks down the first order motion model (FOMM). Section 4 details the seamless cloning methodology and custom foreground extraction techniques [8,25]. Section 5 presents the comprehensive experimental results, and section 6 provides concluding remarks.

## 2 Related Work

Image animation, the process of synthesizing a video by transferring motion patterns from a driving video onto a source image, has become a cornerstone of modern computer vision. Its applications span film production, digital entertainment, and e-commerce. Early approaches to this problem were constrained by the need for strong prior knowledge, often relying on 3D morphable face models or category-specific constraints to reconstruct and animate subjects [12]. While these methods achieved high fidelity within controlled settings (e.g., facial reenactment), they lacked the architectural flexibility to generalize across diverse object categories, such as human bodies, animals, or complex articulated structures.

Recent advancements in the field have shifted focus toward higher-order geometric modeling and improved occlusion handling, moving the field closer to the goal of high-fidelity, real-time animation for any object class. Specifically, motion-representation-based articulated animation (MRAA) provides better stability in articulated motion, ensuring that the movement of connected limbs remains coherent even during complex, high-velocity sequences [13]. Simultaneously, continuous piecewise-affine Based (CPAB) transformations have been proposed to offer highly expressive diffeomorphism spaces, allowing for more precise motion transfer in scenarios where traditional affine transformations fail to capture the curvature of object movement. Furthermore, landmark-guided residual Modules have significantly improved the quality of facial expression animation by focusing on specific residual features, which effectively capture subtle nuances of facial muscle movement that general motion models often overlook. Building on these insights, X2Face [14] demonstrated that a network could learn to control facial expressions and poses using audio or visual driving samples in a self-supervised manner. However, X2Face often struggled to generate fine details due to the limitations of its warping mechanism. Simultaneously, generative models such as MoCoGAN [15] attempted to decompose video into static content and stochastic motion; while innovative, these models often suffered from unnatural object appearance and poor background consistency because the content and motion were not perfectly disentangled. The field reached a major milestone with the introduction of Monkey-Net, the first framework to employ a keypoint detector, a dense motion network, and a motion transfer generator in an end-to-end unsupervised pipeline. Monkey-Net [7] proved that motion transfer could be achieved without predefined models, simply by extracting self-supervised keypoints from a driving video to guide the animation of a source object. By modeling the motion of keypoints, the framework allowed the system to learn the structural dynamics of an object without manual labeling.

Building upon this, the first order motion model (FOMM) significantly improved the handling of complex motions by using local affine transformations in the neighborhood of keypoints rather than treating them as isolated points. FOMM [8] introduced the use of Jacobians to model the transformation of a local patch around a keypoint, allowing for more expressive and natural movement. Furthermore, FOMM introduced an occlusion-aware generator, which employs an occlusion mask to seamlessly handle image regions that are hidden in the source but visible in the driving frame. Because of

its ability to model non-rigid and subtle motions accurately, FOMM became a standard benchmark in the field.

While prior methods like Monkey-Net and the First Order Motion Model (FOMM) have significantly advanced self-supervised video animation through sparse keypoint tracking, dense motion fields, and local affine transformations, they struggle with structural degradation, high-frequency frame noise, and boundary artifacts. Moreover, existing literature has focused on heavy architectural modifications or standard filtering methods (such as standalone Gaussian smoothing) that often blur fine details, reduce Structural Similarity (SSIM) performance in complex dynamic environments, and fail to preserve temporal consistency. While these studies have explored specialized motion composition strategies, a clear gap remains in achieving high-fidelity artifact suppression without requiring costly deep learning re-training. This work is oriented into bridging that gap by demonstrating how a targeted, lightweight post-processing pipeline, combining discrete Poisson equation-based seamless cloning, Gaussian filtering, and custom occlusion-aware thresholding masks, can be directly appended to baseline frameworks like FOMM to mitigate edge discontinuities and enhance reconstruction metrics.

# 3 The Monkey-Net Model Architecture

## 3.1 Model Structure

The Monkey-Net model provides a self-supervised framework for generating video sequences from static source images by tracking structural motion trajectories [7]. The tracking workflow, as shown in Figure 1 below, operates in two primary phases: first, a specialized keypoint detector $\Delta$ isolates coordinates representing facial landmarks or skeletal junctions across both the source asset and driving frames. By computing the difference between these keypoint sets, the model isolates the relative motion vectors required to transform the source geometry into the driving frame's configuration. Second, a dense motion network $M$ translates these sparse coordinate deltas into a continuous dense motion field. This field essentially serves as a dense warping function that models regional structural changes relative to the source image configuration. The output from the dense motion field is passed to a motion transfer generator $G$ alongside the raw source image. The generator employs an encoder-decoder architecture with a latent bottleneck $d$.

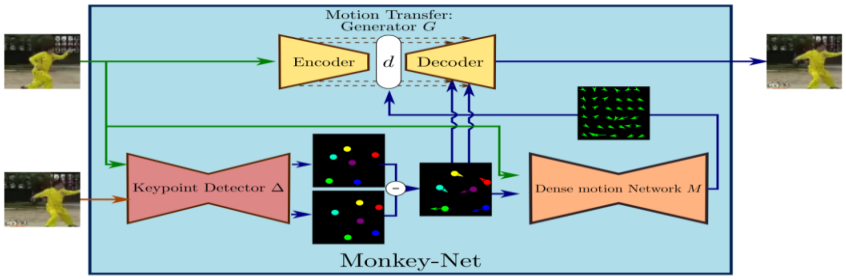



**Fig. 1.** The Monkey-Net model architecture as presented by Siarohin et al. [7,8], illustrating the integration of the encoder-decoder modules, the keypoint detector $\Delta$, and the dense motion network $M$ for synthesizing motion based on a static input and a driving video

The encoder compresses the static source features into a latent representation, which is then spatially modulated by the dense motion field before being reconstructed into the final synthesized frame by the Decoder. This bottleneck forces the network to learn a disentangled representation, separating the structural identity of the source subject from the temporal dynamics imposed by the driving sequence. A defining characteristic of Monkey-Net is its self-supervised training strategy, which eliminates the need for manual annotations or ground-truth motion labels. During training, the model uses a pair of frames from the same video sequence; it treats one as the source and another as the driving frame. The network is then optimized by minimizing the reconstruction loss between the generated frame and the original, withheld ground-truth driving frame. This approach encourages the model to discover robust, generalizable landmarks that effectively capture the motion characteristics of the target object class, such as human bodies or faces, without explicit human supervision.

### 3.2 Network Architecture and Activations

Hidden layers process spatial features and transfer data through predefined activation functions [17]. Non-linear activation functions are universally applied within hidden layers because, according to the universal approximation theorem, a network configured with at least one non-linear hidden layer can approximate any continuous function within a finite number of training steps. Among the widely implemented formulations, the logistic sigmoid $\sigma(x)$ and hyperbolic tangent $\tanh(x)$ functions are defined as:

$$\sigma(x) = \frac{1}{1 + e^{-ax}}, a \in \mathbb{R}$$

$$\tanh(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}. a \in \mathbb{R}$$

For any given neuron, the cumulative input excitation $u(x)$ and the corresponding functional output $o(x)$ are governed by:

$$u(x) = \sum_{i=1}^{d} w_i\, x_i + w_0, w_i \in \mathbb{R} \tag{1}$$

$$o(x) = g(u(x)) \tag{2}$$

where $w_0$ represents the baseline neuron bias associated with a constant static input of 1.

### 3.3 The Gradient Descent Optimization Algorithm

Network parameter optimization relies on the backpropagation algorithm to distribute errors from the output layer back toward the initial hidden connections [18,19]. The gradient descent optimization algorithm is universally favored for training deep neural architectures because it only requires the first derivative of the activation functions. This minimization process helps the network achieve high generalization capabilities on validation sets outside the training distribution. For a system containing $H$ hidden layers, the backpropagation pass defines the localized error gradient $\delta_i^{H+1}$ at the output layer $(H+1)$ across linear or sigmoidal distributions as:

$$\delta_i^{(H+1)} = g'(u_i^{(H+1)})(o_i - t_{ni}), \quad i = 1, \ldots, p \tag{3}$$

$$g'\left(u_i^{(H+1)}\right) = \begin{cases} 1 & \text{for linear activations} \\ o_i(1-o_i) & \text{for sigmoidal activations} \end{cases} \tag{4}$$

This gradient is sequentially propagated backward through preceding hidden layers $h = H, \ldots, 1$ for all constituent nodes $i = 1, \ldots, d^h$ according to the following recursive formulations:

Depending on the batch size configuration, these partial derivatives are accumulated

$$\delta_i^{(h)} = g'(u_i^{(h)}) \sum_{j=1}^{d_{h+1}} w_{jl}^{(h+1)}\, \delta_j^{(h+1)}, \quad i = 1, \ldots, d_h \tag{5}$$

$$g'(u_i^{(h)}) = \begin{cases} 1 & \text{for linear activations} \\ g(u_i^{(h)})(1 - g(u_i^{(h)})) & \text{for sigmoidal activations} \end{cases} \tag{6}$$

to determine the total error gradient relative to individual network weights $(\frac{\partial E^n}{\partial w_{ij}^{(h)}})$ [18,19] and node biases:

$$\frac{\partial E^n}{\partial w_{ij}^{(h)}} = \delta_i^{(h)} g(u_j^{(h-1)}), \tag{7}$$

$$\frac{\partial E^n}{\partial w_0^{(h)}} = \delta_i^{(h)} \tag{8}$$

Using these gradients, parameters are updated at iteration $t + 1$ using an empirical learning rate $\eta$, adopted from the existing FOMM architecture:

$$w_i(t+1) := w_i(t) + \eta \frac{\partial E^n}{\partial w_i^{(h)}} \tag{9}$$

### 3.4 Mathematical Calculation of Keypoints

The keypoint detector is built upon a modified U-Net architecture, which uses symmetric convolutional blocks to execute pixel-level segmentation [24]. Instead of predicting singular coordinates, which is prone to high variance, the network classifies whether individual pixels correspond to target tracking landmarks by outputting localized heatmaps $H_k \in [0,1]^{H \times W}$ for each keypoint $k$ [7]. To derive precise landmark locations, the model treats the heatmap as a probability distribution. The spatial characteristics are parameterized by two primary metrics: the expected spatial coordinate $h_k$ and the structural coordinate covariance $\Sigma_k$, defined over the image pixel domain $U$ as:

$$h_k = \sum_{p \in U} H_k\,[p] \cdot p \tag{10}$$

$$\Sigma_k = \sum_{p \in U} H_k\,[p] \cdot (p - h_k) \cdot (p - h_k)^T \tag{11}$$

The model approximates the probability density of a keypoint's location using a bivariate Gaussian distribution. Figure 2 showcases that the keypoints assignment for any pixel $p$ is determined using a normalization scalar $\alpha$, which ensures the sum of probabilities across the heatmap equals unity:

$$H_k(p) = \frac{1}{\alpha} e^{-(p-h_k)\Sigma_k^{-1}(p-h_k)} \tag{12}$$

**Fig. 2.** Keypoint detection visualization on the Taichi dataset. The colored points represent the expected coordinates $h_k$ derived from the learned Gaussian heatmaps, identifying structural landmarks such as joints and the head

By defining landmarks through heatmaps rather than fixed coordinates, the model gains robustness to occlusions and motion blur. The covariance matrix $\Sigma_k$ effectively captures the uncertainty of a keypoint's position; a larger variance indicates a less confident detection, which allows the subsequent dense motion network to assign lower weights to that specific keypoint when calculating the warping field. This probabilistic approach ensures that even if a landmark is partially obscured during complex movements, the global structural configuration is maintained by the aggregate influence of all visible keypoints.

### 3.5 Dense Motion Field Calculation

The dense motion field $F$ is computed by combining a coarse motion field $F_{coarse}$ with a localized refinement field $F_{residual}$ [7]. Assuming that individual keypoint neighborhoods do not overlap, Monkey-Net derives the coarse trajectory distribution by aggregating $K$ distinct tracking masks $M_k \in \mathbb{R}^{H\times W}$ alongside a dedicated background mask $M_{K+1}$:

$$F_{coarse} = \sum_{k=1}^{K+1} M_k \otimes \rho(h_k) \tag{13}$$

The spatial parameter $\rho(h_k)$ represents a coordinate displacement vector mapped across the frame dimensions. To account for static elements, the background mask $M_{K+1}$ is evaluated against a zero-motion vector $\rho(0)$, isolating the stationary background from the dynamic foreground transformations.

### 3.6 Frame Synthesis Pipeline

To generate a novel frame, the model computes the spatial coordinates of the target keypoints by determining the relative distance between the current driving frame $t$ and the initial baseline driving frame 1. This motion delta is directly added to the source keypoint locations:

$$h_k^{s'} = h_k^s + (h_k{}^t - h_k^1) \tag{14}$$

These findings contribute to a wide body of research on deep pose-guided image animation and scene mapping [20-22].

# 4 The First Order Motion Model (FOMM)

## 4.1 Structural Advances Over Monkey-Net

While Monkey-Net effectively models simple linear motion, its performance degrades when handling complex affine transformations or large displacements between sequential keypoints [8]. To address these limitations, the FOMM replaces sparse point-tracking with local first-order Taylor expansions. Formally, for a source image $S$ and a driving frame $D$, the model estimates a dense motion field $\mathcal{T}_{S\leftarrow D}$ by approximating the local motion around each keypoint $p_k$ using an affine transformation $A_k$. The displacement is modeled as:

where $z$ represents local coordinates. This approach allows the network to model

$$\mathcal{T}_{S\leftarrow D}(z) \approx p_k + A_k(z - p_k) \tag{15}$$

complex mathematical transformations—such as rotation, scaling, and shearing, directly within the neighborhood of each keypoint, rather than relying on global rigid motion assumptions. Additionally, FOMM introduces an occlusion map $\hat{O}_{S\leftarrow D} \in [0,1]^{H'\times W'}$ within its dense motion pipeline. This map acts as a soft-masking mechanism, identifying regions in the generated frame that are either newly revealed or obscured in the source image. By explicitly predicting this map, the model provides a signal to the generator, allowing it to ignore inaccurate warped source pixels and instead rely on the inpainting capabilities of the GAN architecture to realistically reconstruct missing background details or occluded body parts.

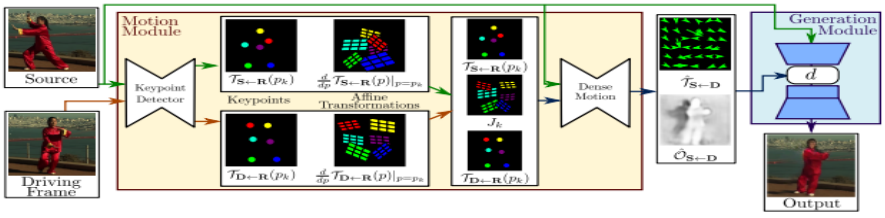



**Fig. 3.** The successor of the Monkey-Net model architecture, the FOMM model, as presented by Siarohin et al. [7,8], illustrating the adapted keypoint detection and dense motion field modules, including the occlusion map and the GAN network for final output calculation

The integration of these motion features occurs within the generation module, where the source image $S$ is warped according to the estimated dense motion field $\mathcal{T}_{S\leftarrow D}$. The

generator, typically implemented as an encoder-decoder network with skip connections, takes the warped source image and the predicted occlusion map as inputs.

Through this adversarial training process, the model learns to synthesize high-fidelity frames that maintain the appearance of the source identity while faithfully replicating the pose and temporal dynamics of the driving sequence. By decoupling appearance (the source frame) from motion (the driving frame), FOMM enables robust animation across diverse subjects, effectively bridging the gap between sparse keypoint tracking and dense pixel-level video generation. The full pipeline of the FOMM architecture can be summarized in Figure 3, where all the mentioned modules, their respective tasks and outputs are presented, along with the formulas on each step.

### 4.2 First-Order Motion Estimations

FOMM assumes the existence of an abstract reference frame $R$ to define coordinate transformations between the source image $S$ and the driving frame $D$ [8]. By applying a first-order Taylor series expansion around each tracked keypoint location $p_k$, the spa-

$$T_{X\leftarrow R}(p) = T_{X\leftarrow R}(p_k) + \left(\frac{d}{dp}T_{X\leftarrow R}(p)|_{p=p_k}\right)(p - p_k) + o(\| p - p_k \|) \tag{16}$$

tial transformation is approximated as:

The localized spatial behavior is captured by estimating both the coordinate positions and their corresponding Jacobian matrices, which represent local affine transformations:

$$T_{X\leftarrow R}(p) \approx \left\{\{T_{X\leftarrow R}(p_1), J_1\}, \ldots, \{T_{X\leftarrow R}(p_K), J_K\}\right\} \tag{17}$$

By computing the inverse transformation $T_{R\leftarrow D} = T_{D\leftarrow R}^{-1}$, the final coordinate mapping from the source space directly to the target driving frame is formulated as:

$$T_{S\leftarrow D}(z) = T_{S\leftarrow R}(p_k) + J_k(z - T_{D\leftarrow R}(p_k)) \tag{18}$$

$$J_k = \left(\frac{d}{dp}T_{S\leftarrow R}(p)|_{p=p_k}\right)\left(\frac{d}{dp}T_{D\leftarrow R}(p)|_{p=p_k}\right)^{-1} \tag{19}$$

$$\hat{T}_{S\leftarrow D} \approx M_0 + \sum_{k=1}^{K} M_k \left(T_{S\leftarrow R}(p_k) + J_k(z - T_{D\leftarrow R}(p_k))\right) \tag{20}$$

The final transformation matrix $\hat{T}_{S\leftarrow D}$ integrates these keypoint-specific Jacobian fields with the background mask $M_0$ to establish a continuous motion field across the entire image domain:

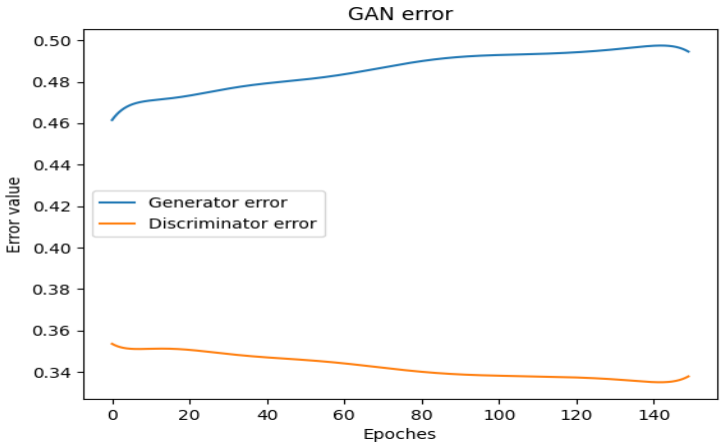



**Fig. 4.** Training dynamics of the GAN module. The convergence of Generator and Discriminator errors indicates the stabilization of the motion field estimation process throughout the training epochs

The efficacy of the estimated motion field $\hat{T}_{S\leftarrow D}$ relies heavily on the adversarial training of the GAN framework. During training, the Generator is tasked with synthesizing the driving frame by applying the motion field to the source image, while the Discriminator distinguishes between real video frames and generated ones. As shown in Figure 4, the evolution of the Generator and Discriminator errors provides a metric for training health. Initially, the model undergoes a period of rapid optimization as it learns to approximate the local affine transformations ($J_k$). As training progresses, the Discriminator error decreases while the Generator error reaches a relative equilibrium, indicating that the motion estimation network has successfully learned to disentangle the motion (deformation) from the appearance (source pixels). This steady-state performance confirms that the first-order approximation sufficiently captures the temporal dynamics required for realistic video synthesis without introducing severe artifacts. Visually, the correlation between the static input image, the driving frames, the extracted keypoints and the occlusion maps, is demonstrated in Figure 5 below.

**Fig. 5.** The full pipeline process of the FOMM architecture, regarding the transition from a static input image towards a frames sequence, guided by the keypoints detected in videos that contain human poses. From left to right, the first column depicts the static images and their respective keypoints, while the second column shows the driving video frames with their keypoints. On each step, the position of the driving frame keypoints indicates the regions the static image ones and the pixel values will be directed to. The final image denotes the form of the occlusion map that is generated on each step, alongside the predicted frame

### 4.3 Feature Warping and Occlusion-Driven Inpainting

Direct pixel mapping often creates spatial misalignments and artifacts due to coordinate discretization errors [8]. To smooth these transitions, FOMM applies a feature warping operator $f_w$ to a latent feature map $\xi \in \mathbb{R}^{H' \times W'}$ extracted via down-sampling convolutional layers. The final latent representation $\xi'$ is generated by computing the Hadamard product ($\odot$) between the warped feature map and the estimated Occlusion Map $\hat{O}_{S \leftarrow D}$:

$$\xi' = \hat{O}_{S \leftarrow D} \odot f_w(\xi, \hat{T}_{S \leftarrow D}) \tag{21}$$

This operation suppresses misaligned features and highlights occluded regions, allowing the decoder to accurately inpaint missing content based on surrounding contextual details. The final combined feature mapping scales boundaries via localized Hadamard tensor operations [23,24].

## 5 Methodology

### 5.1 Continuous and Discrete Poisson Image Editing

Spatial transformations often introduce high-frequency boundary noise and blending artifacts. To address this, we apply the seamless cloning method on the FOMM model Shiarohin et al. [8] introduced. This algorithm models image blending as a bounded optimization problem over a spatial domain $\Omega$ with boundary $\partial\Omega$ [30]. Given a source image patch $f$, its thresholded representation $f^*$ and a destination background image $g$, the algorithm interpolates the pixel values within the target region by minimizing the gradient discrepancy relative to a guidance vector field $\boldsymbol{v}$:

$$\min_{f} \iint_{\Omega} |\nabla f - \boldsymbol{v}|^2 \quad \text{subject to} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{22}$$

where $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]^T$ denotes the gradient operator. The solution to this minimization problem satisfies the Euler-Lagrange equation, yielding the classic Poisson equation with Dirichlet boundary conditions [30]:

$$\Delta f = \text{div}\, \boldsymbol{v} \quad \text{on } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{23}$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ represents the Laplacian operator, and div $\boldsymbol{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of the guidance field $\boldsymbol{v} = (u, v)$. To enforce seamless blending, a correction function $\tilde{f} = f - g$ is applied, transforming the system into a standard Laplace equation [25]:

$$\Delta \tilde{f} = 0 \quad \text{on } \Omega, \quad \text{with} \quad \tilde{f}|_{\partial\Omega} = (f^* - g)|_{\partial\Omega} \tag{24}$$

To implement this continuous model on a discrete pixel grid, let $S$ denote the set of all image pixels, and let $N_p$ define the 4-connected neighborhood of a pixel $p \in S$. The boundary of the discrete cloned region $\Omega$ is defined as $d\Omega = \{p \in S \backslash \Omega : N_p \cap \Omega \neq \emptyset\}$ [25]. The discrete optimization problem is formulated as a quadratic minimization task:

$$\min_{f|_{\Omega}} \sum_{\langle p,q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 \quad \text{with} \quad f_p = f_p^* \quad \forall p \in d\Omega \tag{25}$$

where $v_{pq} = \boldsymbol{v}(\frac{p+q}{2}) \cdot \overrightarrow{pq}$. This optimization system yields a set of linear equations solved for each pixel within the interior of $\Omega$ [25].

$$|N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap d\Omega} f_q^* + \sum_{q \in N_p} v_{pq}\,, \forall p \in \Omega \tag{26}$$

When the region $\Omega$ encounters the physical boundary of the image space, the neigh-

$$\forall p \in \Omega, \quad |N_p| f_p = \sum_{q \in N_p} v_{pq} \tag{27}$$

borhood size scales down ($|N_p| < 4$), reducing the linear system to:

### 5.2 Post-Processing Generated Video Sequences

In this pipeline, the first synthesized frame serves as the structural source image $f$, while subsequent frames are treated as destination targets $g$ [8]. To enhance the visual fidelity of the results, the model employs Poisson Image Editing. This technique allows for seamless integration of the generated foreground into the original background by solving the Poisson partial differential equation, which preserves the gradient of the source image while respecting the boundary conditions of the target domain. The cloning domain Ω is defined directly by the estimated Occlusion Map $\hat{O}_{s\leftarrow D}$. However, directly applying seamless cloning can cause parts of the initial frame's background to bleed into the moving foreground, creating ghosting artifacts. To isolate the moving foreground from these background anomalies, a custom thresholding mask is applied. For each pixel $p$ within a frame $F$, a binary threshold $t$ separates the foreground and background elements:

$$M(p) = \begin{cases} 255 & \text{if } F(p) \leq t \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

A logical bitwise XOR operation ($\oplus$) is then applied between this binary mask $M$ and the frame features to extract a clean foreground region $F^*$, eliminating background distortion around the moving subject and resulting in the post-processed frame comparison that Figure 6 provides:

$$F^* = F \oplus M \tag{29}$$

**Fig. 6.** Visual results of the reconstruction process using the Taichi dataset. The figure demonstrates the final output after applying Poisson blending, showing the successful reduction of ghosting artifacts through the integration of the occlusion map and thresholding

The combination of thresholding and Poisson blending is critical for maintaining temporal consistency. By using the occlusion map to guide the Poisson solver, the system effectively ignores areas where the model is uncertain—such as when a limb moves behind the body or off-camera. The bitwise operations act as a secondary filter,

ensuring that high-frequency noise from the background is not inadvertently cloned onto the moving subject. This post-processing layer essentially acts as a final sanity check, ensuring that the synthesized movement remains structurally coherent with the static background of the scene.

# 6 Experimental Results and Discussion

## 6.1 Experimental Configuration and Metrics

The deep motion frameworks were evaluated using approximately 3,000 video sequences from the standard *Taichi* and *Fashion* public datasets, with resolutions ranging from $256 \times 256$ to $512 \times 512$ pixels [7,8]. Both the keypoint detector and the dense motion network were trained for $N = 150$ epochs using a learning rate of $\eta = 2 \cdot 10^{-4}$, as provided by the existing FOMM architecture, over 150 iterations. Training was executed on high-performance compute clusters over a continuous 5-day period. Three basic quantitative metrics were utilized to evaluate the output quality:

- $L_1$ **Loss:** Measures the absolute pixel-level difference between the generated frame $\hat{y}$ and the ground-truth target $y$ over channels $C$, rows $H$, and columns $W$:

$$L_1 = \frac{||\hat{y} - y||_2^2}{C \cdot H \cdot W} \tag{30}$$

- **Average Euclidean Distance (AED):** Measures the mean structural coordinate tracking shift across $N$ continuous video frames:

$$AED = \frac{\sum_{i=1}^{N} L_1\left(\hat{y}_i, y_i\right)}{N} \tag{31}$$

- **Structural Similarity Index (SSIM):** Measures perceptual image quality by evaluating luminance ($l$), contrast ($c$), and structural composition ($s$) between two image domains:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{32}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{33}$$

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x \cdot \sigma_y + c_3} \tag{34}$$

$$SSIM(x,y) = \| \, l(x,y) \cdot c(x,y) \cdot s(x,y) \, \| \tag{35}$$

where $\mu_x$ and $\mu_y$ represent the average pixel intensities of images $x$ and $y$, $\sigma_x^2$ and $\sigma_y^2$ denote their respective variances, and $\sigma_{xy}$ is the covariance between $x$ and $y$, while $c_1$, $c_2$, and $c_3$ are stabilization constants based on the dynamic pixel range $L = 255$.

### 6.2 Evaluation of Post-Processing Configurations

We evaluated multiple post-processing configurations, ranging from baseline generation and standalone Gaussian filtering to a unified pipeline that integrates Poisson seamless cloning with localized Gaussian filtering, comparing them to the ablation study performed on the previous work [31], that the FOMM model and its variations were examined for their efficiency in video generation, given a static image and a source video.

These experiments were designed to measure the efficacy of noise reduction while maintaining the structural fidelity of the synthesized frames. Applying standalone Gaussian filtering effectively mitigated high-frequency artifacts; however, this approach introduced clear trade-offs in structural retention. On the Fashion dataset, standalone filtering yielded an SSIM of 89.36%,a modest improvement over the 88.85% baseline. Conversely, on the more complex Taichi dataset, standalone filtering resulted in a decrease in SSIM from 65.81% to 63.17%. This degradation indicates that, without structural reinforcement, smoothing operations aggressively suppress fine details in highly dynamic scenes, leading to a loss of temporal and spatial sharpness. In contrast, the unified Poisson-Gaussian pipeline consistently outperformed all comparative methods, demonstrating superior structural reconstruction across both environments. In addition, in the Fashion dataset, the hybrid approach achieved a significantly heightened SSIM of 95.52% with a controlled $L_1$ error of 0.0142%. The performance gains were even more pronounced in the Taichi environment: the cloning pipeline effectively mitigated the noise introduced during synthesis, elevating the SSIM from a baseline of 65.81% to 90.48%, while maintaining a low $L_1$ error of 0.0487%.These results indicate that solving the discrete Poisson equation is critical for preserving sharp edge boundaries while simultaneously suppressing synthesis artifacts, a feat not achievable through standard filtering alone. By leveraging the Poisson equation to maintain structural integrity during the reconstruction of gradient fields, this hybrid methodology successfully minimizes residual texture noise. Consequently, this leads to perceptually natural video synthesis, confirming that the combination of Poisson-based optimization and Gaussian smoothing provides a robust solution for high-fidelity generative modeling.

**Table 1.** Evaluation metrics comparison on the Monkey-Net and FOMM architectures, along with our proposed post-processing method on the FOMM, with the metrics of interest, consisting of the L1, AED and SSIM metrics. [7, 8, 31]

| Dataset | Model | Metric | Gaussian Noise | | Poisson+Gaussian | |
|---|---|---|---|---|---|---|
| | | | W/O | With | W/O | With |
| Fashion | FOMM [8, 31] | L1 (%) | 2.48000 | 2.26000 | N/A | N/A |
| | | AED (%) | 9.70000 | 9.88000 | N/A | N/A |
| | | SSIM (%) | 91.52 | 92.38 | N/A | N/A |
| | Proposed | L1 (%) | 0.0142 | 0.08189 | 0.0142 | 0.01422 |
| | | AED (%) | 0.02001 | 161.00461 | 0.02001 | 27.96285 |
| | | SSIM (%) | 88.85 | 89.36 | 88.85 | 95.52 |
| Taichi | Monkey-Net [7] | L1 (%) | 7.70000 | N/A | N/A | N/A |
| | | AED (%) | 22.80000 | N/A | N/A | N/A |
| | FOMM [8, 31] | L1 (%) | 5.63000 | 5.79000 | N/A | N/A |
| | | AED (%) | 15.50000 | 15.52000 | N/A | N/A |
| | | SSIM (%) | 76.50 | 75.46 | N/A | N/A |
| | Proposed | L1 (%) | 0.003 | 0.06789 | 0.003 | 0.048 71 |
| | | AED (%) | 2.85e-07 | 133.47093 | 2.85e-07 | 95.76302 |
| | | SSIM (%) | 65.81 | 63.17 | 65.81 | 90.48 |

Despite the quantitative performance gains achieved by the proposed hybrid pipeline, certain visual constraints and time constraints must be acknowledged. Visually, while the discrete Poisson solver and custom thresholding masks successfully suppress high-frequency background noise and ghosting artifacts, extreme driving poses or rapid, self-occluding movements can occasionally cause the occlusion map to misidentify boundaries.

This results in localized blurring or minor blending inconsistencies around complex structural edges. From a temporal perspective, solving the discrete Poisson equation iteratively across sequential video frames introduces a computational overhead compared to direct, feed-forward inference. Consequently, while the pipeline avoids the heavy resource burdens of deep learning re-training, processing high-resolution video streams or long frame sequences can be challenging in accurately preserving the visual information that is migrated into the next frame.

## 7 Conclusion

This paper demonstrates that combining deep motion synthesis models with optimization-based post-processing significantly improves the visual quality and structural fidelity of generated video sequences. While deep frameworks like the First Order Motion Model (FOMM) successfully capture complex motion trajectories, they inherently suffer from structural degradation, high-frequency frame noise, and boundary artifacts. Through our experimental evaluation, we show that integrating a discrete Poisson equation solver, guided by estimated occlusion maps and custom foreground thresholding masks, effectively suppresses synthesis noise and eliminates edge discontinuities without the heavy computational burden of deep learning re-training. Furthermore, quantitative assessments on the Taichi and Fashion datasets confirm that this lightweight pipeline preserves structural integrity, yielding substantial improvements in Structural Similarity (SSIM) and Average Euclidean Distance (AED) metrics. Ultimately, these results highlight that heavy architectural modifications are not the sole path to elevating video generation performance. Instead, targeted, mathematically grounded post-processing can efficiently bridge the gap between sparse keypoint tracking and high-fidelity output. This makes the proposed methodology highly practical for real-world applications, such as in film production process, digital entertainment, and even animation where both visual consistency and computational efficiency are essential, especially when dealing with a great number of frames and time restraints.

## References

1. Wei, C.-C., Yeh, C., -H, Wang, I., Lin, Y., -C.: Deep neural networks for new product form design. In: ICEIS (2019)
2. Syberfeldt, A., Vuoluterä, F.: Image processing based on deep neural networks for detecting quality problems in paper bag production. Procedia CIRP. 93, 1224–1229 (2020)

3. Papacharalampopoulos, A., Petridis, D., Stavropoulos, P.: Experimental investigation of rubber extrusion process through vibrational testing. Procedia CIRP. 93, 1236–1240 (2020)
4. Wu, H., Zhou, Z.: Using convolution neural network for defective image classification of industrial components. Mobile Inf. Syst. 2021, 1–8 (2021)
5. Weiss, R., Karimijafarbigloo, S., Rödiger, S.: Applications of neural networks in biomedical data analysis. Biomedicines. 10, 1469 (2022)
6. Nagy, B., Galata, D., L., Farkas, A., Nagy, Z., K.: Application of artificial neural networks in pharmaceutical manufacturing. AAPS J. 24, 42 (2022)
7. Siarohin, A., Lathuiliere, S., Tulyakov, S., Ricci, E., Sebe, N.: Animating arbitrary objects via deep motion transfer. In: IEEE CVPR (2018)
8. Siarohin, A., Lathuiliere, S., Tulyakov, S., Ricci,E., Sebe, N.: First order motion model for image animation. NeurIPS (2020)
9. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2D & 3D face alignment problem? In: IEEE ICCV (2017)
10. Aggarwal, A., Mittal, M., Battineni, G.: Generative adversarial network: An overview of theory and applications. Int. J. Inf. Manage. Data Insights. 1, 100004 (2021)
11. Kingma, D., Welling, M.: Auto-encoding Variational Bayes. In: ICLR (2013)
12. Zollhöfer,M., Thies, J., Bradley, D., Garrido, P., Beeler, T., Péerez, P., Stamminger, M., Nießner, M., Theobalt, C.: "State of the art on monocular 3d face reconstruction, tracking, and applications," 2018
13. Cao, C., Hou, Q., Zhou, K.: "K.: Displaced dynamic expression regression for real-time facial tracking and animation," in In ACM TOG (Proc. SIGGRAPH) (2014)
14. Wiles, O., Koepke, A., S., Zisserman, A.: "X2face: A network for controlling face generation by using images, audio, and pose codes," CoRR, vol. abs/1807.10550, 2018
15. Tulyakov, S., Liu, M.-Y., Yang, X., Kautz, J.: "MoCoGAN: Decomposing motion and content for video generation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1526–1535
16. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Niessner, M.: "Face2face: Real-time face capture and reenactment of rgb videos," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
17. Lu, Y., Lu, J.: A universal approximation theorem of deep neural networks for expressing distributions. arXiv preprint arXiv:2004.04169 (2020). Tang, G., Chaoshun
18. L., Xiaoming, X., Xinbiao, C., Ruoheng., W., Chu, Z.: The short-term interval prediction of wind power using the deep learning model with gradient descent optimization. Renewable Energy. 155, 345–358 (2020)
19. Li, C., Tang, G., Xue, X., Chen, X., Wang, R., Zhang, C.: Deep interval prediction model with gradient descent optimization method for wind power forecasting. IEEE Trans. Sustainable Energy (2019)
20. Hanczar, B., Zehraoul, F., Issa, T., Aries, M..: Biological interpretation of deep neural network for phenotype prediction based on gene expression. BMC Bioinformatics. 21, 432 (2020)
21. Yang, C., Vadlamani, A., Soloviev, A., Veth, M., Taylor, C.: Feature matching error analysis and modeling for consistent estimation in vision-aided navigation. Navigation. 65, 265–278 (2018)
22. van Dyck, L., Kwitt, R., Denzler, S., Gruber, W..: Comparing object recognition in humans and deep convolutional neural networks. Front. Neurosci. 15, 750639 (2021)
23. Kim, H., Kim, J., Jung, H.: Convolutional neural network based image processing system. J. Inf. Commun. Converg. Eng. 16, 160–165 (2018)

24. Yin, X., Sun, L., Fu, Y., Lu R., Zhang, Y.: U-Net-based medical image segmentation. J. Healthc. Eng. 2022, 1–16 (2022)
25. Grigorev, A., Sevastopolsky, A., Vakhitov, A., Lempitsky, V.: Coordinate-based texture inpainting for pose-guided human image generation. In: IEEE CVPR (2019)
26. Drobyshev, N., Chelishev, J., Khakhulin, T.: MegaPortraits: One-shot megapixel neural head avatars. ACM Trans. Graph (2022)
27. Villegas, R., Yang., J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: ICML (2017)
28. Kishka, Z., Saleem, M., Abdalla, M., Elrawy, A.: On Hadamard and Kronecker products over matrix of matrices. Gen. Lett. Math. 4, 13–22 (2018)
29. Weitao, J., Hu, H.: Hadamard product perceptron attention for image captioning. Neural Process. Lett. 1–18 (2022)
30. Droniou, J., Eymard, R., Gallouët, T., Guichard, C., Herbin, R.: Dirichlet boundary conditions. In: The Coregradient Discretisation Method, pp. 35–62. Springer, description space (2018)
31. Vrigkas, M., Tagka, V., Plissiti, M. E., Nikou, C. (2023, June). Composition of motion from video animation through learning local transformations. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE