

AI-Assisted Frame Completion For Traditional Animation

Dr J Seetha¹, Professor

Department of Computer Science and Business Systems, Panimalar Engineering College,
Chennai, India

Mr Santosh E², Mr Lokesh Saran KB³, Mr Mario Antony Roy R⁴

^{2,3,4} Students, Department of Computer Science and Business Systems
Panimalar Engineering College, Chennai, India

Abstract

Traditional 2D animation requires artists to manually draw a large number of intermediate frames to achieve smooth motion, which is time- consuming and labour-intensive. This paper presents an AI-assisted point-based animation in- Betweening system designed to reduce manual effort by automatically generating intermediate frames from a limited number of keyframes. The proposed system represents animation poses using coordinate points rather than pixel-level images, enabling lightweight computation and structured motion interpolation. The system includes modules for pose data preparation, keyframe input, motion analysis, frame generation, and visualization. Motion between frames is estimated using coordinate interpolation and simple machine learning regression models. Experimental analysis shows that the system produces smooth transitions while maintaining low computational cost. This approach is suitable for beginner animators, educational use, and rapid animation prototyping.

Keywords: computational power, and GPUs. Such approaches are often impractical for small-scale projects or academic environments.

This project proposes a lightweight alternative: a point-based in-Betweening system. Instead of working with full images, animation poses are represented using coordinate points on a drawing board. Motion is modelled as changes in these coordinates over time. This structured Frame interpolation, Motion prediction, Pose- based animation, Coordinate-based motion modelling, Animation synthesis, Deep learning for animation

1. Introduction

Animation is a powerful medium for storytelling, education, and entertainment. Traditional 2D animation is based on drawing a sequence of images where each frame represents a slightly different pose. When played rapidly, these frames create the illusion of movement. However, producing smooth animation typically requires 12–24 frames per second, making manual frame creation extremely demanding.

A common workflow in animation involves drawing keyframes first—these represent major poses or important moments. The intermediate frames, known as in-betweens, are then drawn to smooth the motion. This stage often consumes the majority of production time.

Recent research has explored automated in- Betweening using deep learning and video interpolation. However, many methods rely on pixel-level processing, requiring large datasets, high

representation simplifies computation while still capturing motion patterns.

The primary objective is to design an accessible system that reduces manual effort while preserving motion smoothness and animator control.

2. Related Work

Automated animation and motion interpolation have been extensively explored in computer graphics, computer vision, and machine learning research. Existing approaches can broadly be categorized into pixel-based interpolation, skeleton-based motion

modelling, and coordinate-based motion representation. Each paradigm offers different trade-offs between realism, computational cost, and applicability to hand-drawn animation.

2.1 Pixel-Based Frame Interpolation

Video frame interpolation techniques estimate motion between images using optical flow and deep neural networks. Models such as RIFE and transformer-based methods generate high-quality results but require large datasets and GPU acceleration. These methods operate on raw pixels and are often optimized for video restoration rather than hand-drawn animation.

2.2 Skeleton-Based Motion Modelling

In character animation research, motion is often represented using skeletal joints. Machine learning models predict joint trajectories to synthesize movement. While effective, these approaches target 3D motion capture scenarios and may be unnecessarily complex for simple 2D animation tasks.

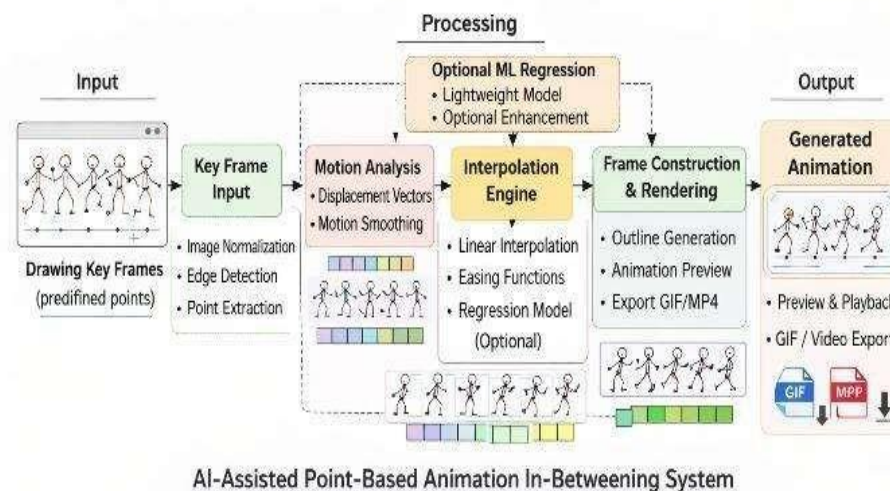
2.3 Coordinate-Based Motion Representation

Recent studies highlight the benefits of representing motion as coordinate sequences. This reduces dimensionality and allows structured interpolation. Such representations align closely with point-based animation workflows.

Compared to prior work, the proposed system focuses on a simplified coordinate-level representation tailored to 2D drawing. It avoids heavy models while maintaining practical usefulness.

3. Methodology

The proposed AI-assisted in-Betweening system converts user-provided keyframes into smooth animation sequences using a coordinate-based approach. Instead of pixel-level processing, the system operates on point representations of poses, reducing computational complexity while preserving motion structure. The methodology consists of five stages: pose data preparation, keyframe acquisition, motion analysis, interpolation, and rendering.



3.1 Pose Data Preparation

User-drawn frames may vary in size and quality, so preprocessing ensures consistency. Input images are normalized to a fixed resolution and converted to grayscale. Basic smoothing and thresholding reduce noise.

Edge detection is applied to identify outlines, from which key coordinate points are

extracted. Each pose is represented as a set of (x, y) points. This coordinate-based format simplifies later computations and maintains structural information.

3.2 Keyframe Acquisition

Users provide a small number of keyframes representing major poses. These may be drawn on a canvas or uploaded. Each keyframe is converted into a structured point set. Point correspondence across frames is maintained using consistent placement or nearest-neighbour matching. A timeline interface allows users to define the number of in-between frames required.

3.3 Motion Analysis

Motion between keyframes is estimated by computing displacement vectors for corresponding points:

$$= (x_2 - x_1, y_2 - y_1)$$

These vectors describe motion direction and magnitude. Smoothing is applied to avoid abrupt transitions and ensure natural movement.

3.4 Interpolation

Intermediate positions are generated using interpolation.

Linear Interpolation

$$= (1 - t)P_1 + tP_2$$

where t is the interpolation factor.

Easing Functions

Ease-in and ease-out functions improve motion naturalness by simulating acceleration and deceleration.

Optional ML Assistance

A simple regression model may predict intermediate positions based on prior motion trends, providing smoother transitions without heavy computation.

3.5 Frame Construction and Rendering

Interpolated points are connected to reconstruct outlines. Each reconstructed pose becomes a frame in the sequence.

The animation is rendered for preview, and users can adjust speed or refine keyframes. Final output can be exported as GIF or video.

4. Results

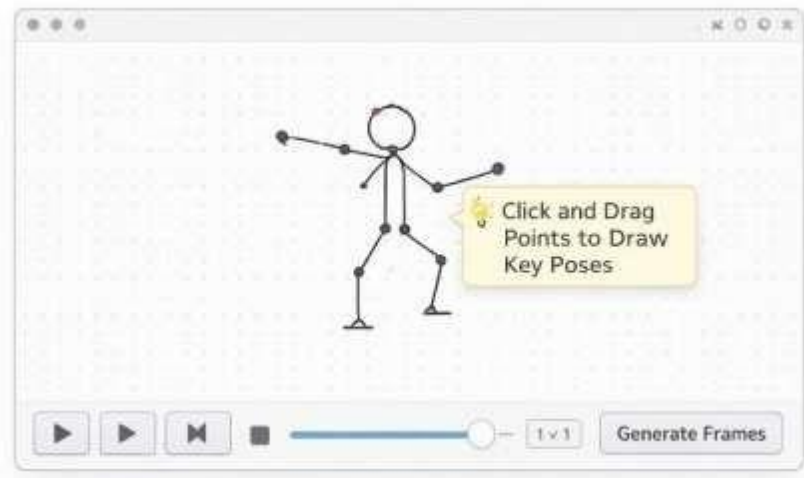
The results of the proposed system are demonstrated through the functional application interface developed for AI-assisted animation in- Betweening. The system provides an interactive workflow where users can create, process, and preview animations. The main output of the project is a working application consisting of multiple functional pages.

4.1 Keyframe Drawing Interface

The application includes a drawing canvas where users can sketch key poses using point-based outlines. Users can place and adjust coordinate points to form character poses. This interface ensures flexibility and allows precise control over keyframe creation.

Each keyframe is stored as a structured set of coordinate points, which serves as input for interpolation.

Fig. 1. Keyframe Drawing Interface



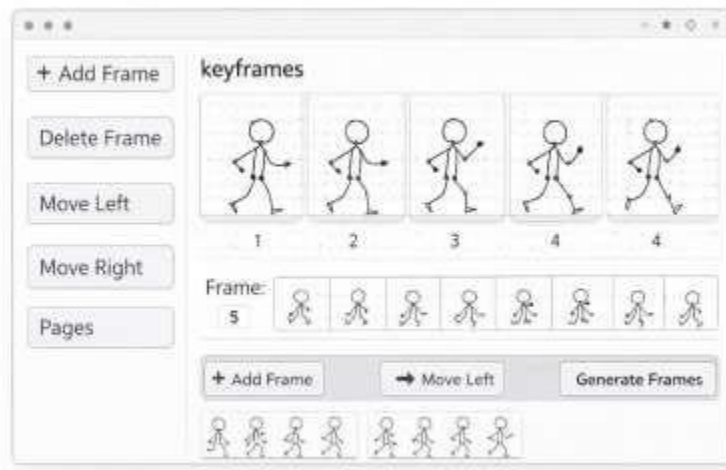
4.2 Keyframe Timeline Page

A timeline interface allows users to arrange keyframes in sequence. Users can:

- Add or delete keyframes
- Reorder frames
- Specify the number of in-between frames

This page ensures proper temporal ordering for motion generation.

Fig. 2. Keyframe Timeline Page



4.3 Motion Processing Page

Once keyframes are finalized, the motion processing module analyses displacement between points. This page shows processing status and prepares data for interpolation.

The system computes motion vectors and applies smoothing to ensure natural transitions.

Fig. 3. Motion Processing Page



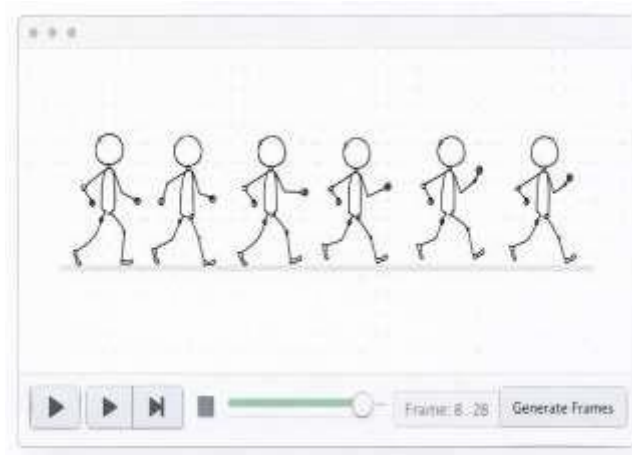
4.4 Animation Preview Page

The preview interface displays generated in- between frames as a continuous animation. Users can:

- Play/Pause animation
- Adjust playback speed
- Review motion quality

This allows users to verify smoothness before exporting.

Fig. 4. Animation Preview Page



4.5 Export Page

The final animation can be exported as:

- GIF format
- MP4 video

This makes the output usable for presentations, projects, and creative work.

Fig. 5. Export Pag



4.6 Overall Outcome

The developed application successfully demonstrates automated in-betweening from minimal user input. The interface is simple and beginner-friendly. The results confirm that point- based interpolation can produce visually smooth animations while requiring low computational resources.

5. Implementation

The proposed AI-assisted in-betweening system was implemented as a prototype application that combines a point-based drawing interface with machine learning-based motion interpolation.

The implementation focuses on simplicity, modularity, and feasibility within academic project constraints.

5.1 Development Environment

The system was developed using Python as the primary programming language due to its strong ecosystem for machine learning and image processing. The main tools and libraries include:

- PyTorch – for implementing and running ML models
- OpenCV – for image processing and frame handling
- NumPy – for numerical operations on coordinate data
- Matplotlib / GUI toolkit (Tkinter or PyQt) – for visualization and interface prototyping
- Google Colab / Local GPU setup – for model training and testing

These tools were selected for accessibility, community support, and compatibility with deep learning workflows.

5.2 System Modules Implementation

1) Keyframe Drawing Interface

A 2D canvas with a grid of points was implemented where users can place and drag points to form stick-figure poses. Each pose is stored as a set of (x, y) coordinates.

This coordinate-based storage significantly reduces data size compared to pixel images and simplifies motion modelling.

2) Keyframe Management

Keyframes are stored sequentially in a timeline structure. Users can:

- Add or delete frames
- Reorder frames
- Preview static poses

Each frame is indexed and linked to its coordinate set for later interpolation.

3) Motion Analysis & Interpolation

Motion between two keyframes is computed by:

- Calculating displacement vectors for each point
- Estimating direction and magnitude of motion
- Applying linear or ML-based interpolation

A simple ML model (e.g., feedforward network or lightweight LSTM) can be trained on coordinate sequences to predict intermediate positions. For a prototype, linear interpolation is used as a baseline and ML interpolation as an enhancement.

4) Intermediate Frame Generation

Given two keyframes and a desired number of in- betweens:

- The system divides motion into equal time steps
- Coordinates are interpolated for each step
- New frames are constructed from predicted points

This ensures smooth transitions and consistent spacing between poses.

5) Preview & Rendering

Generated frames are rendered back into stick- figure visuals on the canvas. A playback module allows:

- Frame-by-frame preview
- Adjustable playback speed
- Animation validation before export

6) Export Module

The final animation sequence is exported as:

- GIF for lightweight sharing
- MP4 video for higher quality playback

OpenCV video writing utilities are used to assemble frames into a video format.

5.3 Training Strategy

If ML interpolation is used:

- Synthetic pose sequences are generated for training
- Input: previous and next keyframe coordinates
- Output: predicted intermediate coordinates
- Loss Function: Mean Squared Error (MSE) between predicted and true positions

Due to project scope limits, lightweight models are preferred over deep transformer architectures.

6. Experimental Evaluation

The proposed point-based in-betweening system was evaluated for motion smoothness, consistency, and computational efficiency. The goal was to verify that the system can generate smooth intermediate frames with low resource usage.

6.1 Evaluation Setup

Tests were conducted on a standard computer without GPU support. Simple 2D motions such as jumping, hand waving, and walking were used.

Each sequence contained 3–5 keyframes with automatically generated in-betweens.

6.2 Evaluation Metrics

The system was evaluated based on:

- Motion Smoothness – continuity between frames
- Structural Consistency – stable point correspondence
- Computational Efficiency – low processing time and memory use
- User Usability – ease of drawing and previewing

6.3 Observations

The system produced smooth transitions for simple motions. Linear interpolation and easing reduced abrupt changes. Because processing is coordinate-based, computation was fast and lightweight.

Evaluation Metric	Observation Method	Result
Motion Smoothness	Visual inspection of transitions	Smooth transitions for simple motions
Structural Consistency	Checking point alignment across frames	Stable point correspondence maintained
Computational Efficiency	Processing time measurement	Near real-time generation
Memory Usage	Resource monitoring	Low memory consumption
User Usability	Informal user feedback	Interface easy to use
Automation Benefit	Comparison with manual drawing	Reduced manual workload

Users found the interface intuitive for quick animation creation.

7. Discussion

The experimental results indicate that point- based interpolation is a practical solution for automating simple 2D animation in-betweening. By operating on coordinate points rather than full images, the system achieves low computational cost while maintaining acceptable motion smoothness. This makes it suitable for academic environments and beginner animators who may not have access to high-end hardware.

The findings also highlight that structured coordinate representations simplify motion modeling and reduce data requirements. Unlike pixel-based deep learning methods, the proposed approach does not rely on large datasets, making it easier to implement and train.

However, the system performs best for simple and moderately varying motions. Complex

movements or large pose differences between keyframes may produce less natural transitions. This suggests that while rule-based and lightweight ML interpolation are effective for basic animation, more advanced models could further improve realism. Overall, the project demonstrates a balance between automation and user control. It supports learning and rapid prototyping while leaving room for future enhancements such as style-aware interpolation or pose extraction from images.

8. Limitations

- Motion quality depends on accurate and consistent point placement by the user.
- Performs best for simple motions; large pose changes may reduce smoothness.
- Requires manual creation of keyframes; no automatic pose extraction.
- Does not capture artistic style elements like shading or line thickness.
- Limited to 2D animation and does not support 3D motion.
- Lightweight interpolation may not match high-end deep learning realism.

9. Future Work

- Integrate automatic pose extraction from images or videos using computer vision.
- Support style-aware interpolation to preserve artistic drawing styles.
- Extend the system to 3D animation and skeletal modelling.
- Enable real-time in-between generation for live animation workflows.
- Improve interpolation using advanced ML models for complex motions.
- Add game engine export support (e.g., Unity or Unreal Engine).

10. Conclusion

This work presented an AI-assisted point-based frame in-betweening system aimed at simplifying the animation process by automatically generating intermediate frames from a small number of keyframes. By representing poses as coordinate points instead of full images, the system reduces computational complexity while maintaining motion continuity.

The proposed approach demonstrates that smooth transitions can be achieved using lightweight interpolation techniques combined with optional machine learning support. Experimental evaluation shows that the system is efficient, user-friendly, and suitable for beginners and educational use. Although it is not intended to replace professional animation tools, it provides a practical solution for rapid prototyping and learning. Overall, the project highlights the potential of coordinate-based motion modelling as an accessible and scalable approach for animation automation, with opportunities for future improvements in realism and functionality.

11. References

1. Zhou et al., "SILK: Smooth Interpolation Framework," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops – Human Motion Generation, 2025.
2. Lin et al., "SimMotionEdit: Text-Based Motion Editing," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.
3. Kwon et al., "Framer: Interactive Frame Interpolation," International Conference on Learning Representations (ICLR), 2025.
4. Huang et al., "AceVFI: Video Frame Interpolation: A Comprehensive Survey," arXiv preprint arXiv, 2025.
5. Zhang et al., "Motion Keyframe Interpolation via Point Cloud Representation," arXiv preprint arXiv, 2024.

6. Liu et al., "Long-Term Motion In-Betweening via Keyframe Prediction," Pacific Conference on Computer Graphics and Applications (Pacific Graphics), 2024.
7. Sun et al., "Advanced Articulated Motion Prediction," Frontiers in Computer Science – Artificial Intelligence Section, Elsevier, 2024.
8. Holden et al., "Motion In-Betweening with Phase Manifolds," Proceedings of the ACM on Computer Graphics and Interactive Techniques (PACM CGIT), ACM, 2023.
9. Mao et al., "Shuffled Autoregression for Motion Interpolation," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023.
10. Martinez et al., "Sequence-to-Sequence Learning for Human Motion Prediction," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2023.
11. Harvey et al., "Conditional Motion In- Betweening," Pattern Recognition, Elsevier, 2022.
12. Kim et al., "Global-Local Motion Transformer," European Conference on Computer Vision (ECCV), Springer, 2022.
13. Chen et al., "Hierarchically Self-Supervised Transformer for Skeleton-Based Representation Learning," European Conference on Computer Vision (ECCV), Springer, 2022.
14. Sridhar et al., "Transformer-Based Motion In-Betweening," International Conference of the Italian Association for Artificial Intelligence, Springer, 2022.
15. Wang et al., "Skeleton-Based Motion Prediction: A Survey," IEEE Access / IEEE Survey-based publication, 2022.
16. Huang et al., "RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation," European Conference on Computer Vision (ECCV), Springer, 2022.
17. Li et al., "Learning Multiscale Correlations for Human Motion Prediction," IEEE International Conference on Development and Learning (ICDL), IEEE, 2021.
18. Park et al., "Controllable Human-Centric Keyframe Interpolation," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021.
19. Tulyakov et al., "TimeLens: Event-Based Video Frame Interpolation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021.
20. Cao et al., "Attention-Based Time Series Imputation for Missing Data," IEEE International Conference on Data Mining (ICDM), IEEE, 2021.