

GENERATIVE AI FOR STOP-MOTION ANIMATION

by

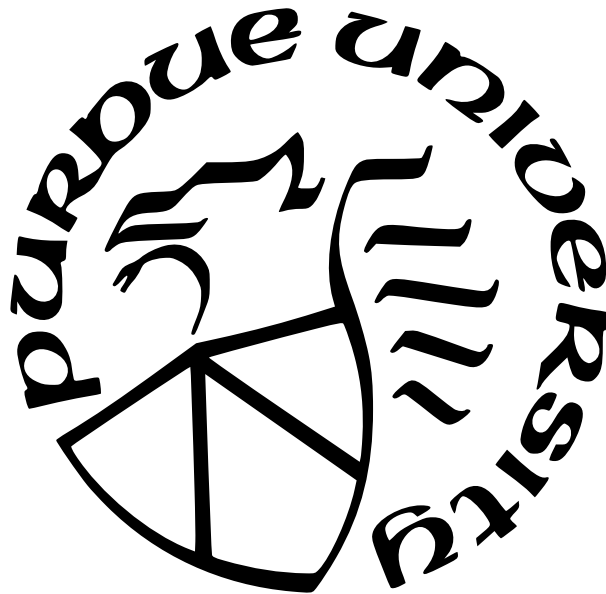
Chun Meng Yu

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science



Department of Computer Graphics Technology

West Lafayette, Indiana

December 2025

**THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL**

Dr. Stephen Baker, Chair

School of Applied and Creative Computing

Dr. Tim McGraw

School of Applied and Creative Computing

Dr. Christos Mousas

School of Applied and Creative Computing

Dr. Liang He

The University of Texas at Dallas

Dr. Angus Forbes

Nvidia Corporation

TABLE OF CONTENTS

LIST OF TABLES	5
LIST OF FIGURES	6
ABSTRACT	9
1 INTRODUCTION	10
1.1 Background	10
1.1.1 Challenges of Traditional Stop-motion Animation	10
1.1.2 Other Animation Techniques	11
1.1.3 Existing Generative AI Models for Animation	13
1.2 Objectives	14
2 REVIEW OF EXISTING WORK	15
2.1 Text-to-video models	15
2.2 Motion-controllable models	16
2.3 Frame Interpolation models	17
2.4 Pose transfer approaches	19
2.5 Diffusion models for pose transfer	20
3 METHODOLOGY	22
3.1 User Input	24
3.1.1 Pose Reference Frames	24
3.1.2 Character reference images	24
3.2 Pose detection	26
3.3 Finetuning	28
3.4 Pose transfer	29
3.5 User Interface	29
3.5.1 Timeline	30
3.5.2 Assets Panel	31
3.5.3 Render Panel	35
3.5.4 Properties Panel	36
4 RESULTS	38
4.1 Representative Examples	38

4.1.1	Simple cases	38
4.1.2	Complexity of physical characteristics	43
4.1.3	Complexity of motion	45
4.2	Quantitative Evaluation	47
4.2.1	Evaluation Data	47
4.2.2	Evaluation Metrics	49
4.2.3	Quantitative Results	50
4.2.4	Human-in-the-Loop Improvements	52
4.3	Key findings	54
5	USE CASES	55
5.1	Creating Animations with Pose-to-Pose	55
5.2	Creating Animations with Motion-Capture	58
5.3	Editing Existing Animations	62
6	LIMITATIONS AND FUTURE WORK	65
6.1	Non-Humanoid Characters	65
6.2	Character Interactions	66
6.3	Facial Features and Expressions	67
7	CONCLUSION	70
	REFERENCES	71
A	APPENDIX A	77

LIST OF TABLES

4.1	Results	50
A.1	Results Breakdown for 10 Animations in the Evaluation Dataset	77
A.2	Comparison of results for 10 characters in the evaluation dataset	78
A.3	Sources	81

LIST OF FIGURES

2.1	Result from Stable Video Diffusion using reference images of a figurine	16
2.2	Input and output from the Go-with-the-Flow model [37]. There are distortions on the ducks face when it rotates to face the camera. The model also adds water dripping down in the background which wasnt specified in the input.	17
2.3	Result from the FILM model	18
2.4	Result from Interpolating between Images with Diffusion Models [39]	18
2.5	Result from Meta AI Animated Drawings [40]	19
2.6	Input and output of the Text2Video-Zero model [41]	19
3.1	Flowchart of the system pipeline	22
3.2	Example input images of a figurine	25
3.3	Pose detection of the motion reference video	26
3.4	Pose detection for a more complex pose	27
3.5	Pose detection failure due to shadow	28
3.6	The user interface containing the Timeline, Assets Panel, Render Panel, and Properties Panel	30
3.7	The user interface with multiple layers in the Timeline	31
3.8	User interface for adding character reference images	32
3.9	Adding images and videos in the Assets Panel	33
3.10	Uploading a motion-capture video as pose reference	34
3.11	Uploading a rough sketches as pose reference	35
3.12	The generated frame overlayed with the detected pose, the original pose reference frame, and the onion-skinning of previous and subsequent frames	36
4.1	Input images of a realistic human figurine	39
4.2	Frames from the input motion reference video of a man walking	39
4.3	Baseline results without finetuning	40
4.4	Results with finetuning	40
4.5	Input images of a Spiderman figurine	41
4.6	Baseline results without finetuning	41
4.7	Results with finetuning	42

4.8	Results using three input images instead of two	42
4.9	Input images of a Hulk figurine	43
4.10	Results for different body types	43
4.11	Input images of a figurine with an additional feature	44
4.12	Results for a figurine with an additional feature	44
4.13	Input images of an abstract figurine	45
4.14	Results for an abstract figurine	45
4.15	Frames from a video of a ballet performance to input as the motion reference video	46
4.16	Four of the nine input images of a figurine in different poses	46
4.17	Results for complex motion (ballet)	47
4.18	Evaluation data example: Walk cycle animation	48
4.19	Evaluation data example: Input images	48
4.20	More examples from the evaluation data	49
4.21	Distribution of evaluation metrics scores	51
4.22	Metrics improvements after fix the pose	53
4.23	Metrics improvements after adding one more character reference image	53
4.24	Metrics improvements after editing the image	54
5.1	Character reference images for the snowangel pose-to-pose use case	55
5.2	Generating keyframes from uploaded sketches	56
5.3	Editing incorrectly detected poses and regenerating the keyframes	56
5.4	Comparing two ways to generate intermediate frames from keyframes	57
5.5	Generating keyframes from uploaded sketches	57
5.6	The final results of the snowangel pose-to-pose use case	58
5.7	Frames from the completed animation in the snowangel pose-to-pose use case . .	58
5.8	Character reference images of the Spiderman figurine	59
5.9	Character reference images of the Mary-Jane figurine	59
5.10	Uploading a pose reference video of a ballet performance	60
5.11	Uploading a pose reference video that the user recorded of themselves dancing .	60
5.12	Changing the character reference images to fix an incorrectly generated frame .	61

5.13	Using an image editor to erase an artifact from a generated frame and replacing the frame in the animation	61
5.14	Editing an incorrectly detected pose to match the pose reference frame and re-generating the frame with the corrected pose	62
5.15	Final results of the motion-capture use case of two figurines dancing	62
5.16	Uploading an existing animation	63
5.17	Editing a pose in an existing animation	63
5.18	Editing the appearance of a character in an existing animation	64
6.1	Example output from DeepLabCut [63], an animal pose detection model (image source: https://deeplabcut.medium.com/)	65
6.2	Multi-pose detection with DWPose [27] (original image source: https://torontoobserver.ca/2013/02/14/worlds-next-stop-for-ice-dance-pair-gilles-poirier/)	66
6.3	Comparing faces of the original character reference image and the generated frame for realistic-looking characters	67
6.4	Comparing faces of the original character reference image and the generated frame for unrealistic-looking characters	68
6.5	Example results from FSRT [66], a facial expression transfer model (image source: https://github.com/andrerochow/fsrt)	69

ABSTRACT

Stop-motion animation is traditionally created using the straight-ahead approach, where animators incrementally adjust physical objects and capture each frame sequentially. This process is a time-consuming and repetitive process that limits the ability to plan or edit complex scenes.

This research introduces a system that leverages recent advances in AI, including pose detection and pose transfer models, to generate stop-motion animations in a more flexible and efficient way. The proposed pipeline takes as input a video or sketches describing the motion of a character (pose reference frames) and photos of a figuring (character reference images). The appearance of the resulting animation would be fully controllable using the character reference images, and the motion would be fully controllable using the pose reference frames. The user can then make detailed edits to the animation using tools provided by the user interface.

Through qualitative evaluation on representative examples and quantitative evaluation on a dataset of animations, the system demonstrates that it can generate high-quality animations. The system can support animators through the entire stop-motion animation process, enabling use cases such as creating new animations through pose-to-pose or motion-capture techniques, or refining existing animations at the frame level. This would greatly reduce the workload, allowing more animators to explore stop-motion animation.

1. INTRODUCTION

Stop-motion is a form of animation that could be applied as a practical tool to many domains, such as storytelling, film [1], education [2][3][4][5], architecture [6], communication [7], and personal growth [8]. However, because the animation process is time-consuming and difficult to master, there are many barriers preventing this artform from being more widespread.

There are many types of stop-motion animation, such as animating physical puppets or figurines, creating timelapse videos of natural phenomenon and astronomy, or animating drawings created in various mediums such as paint, sand, or paper cutouts [9]. This research focuses mainly on the animation of figurines.

Traditionally, to create stop-motion animations of figurines, animators need to shift the scene little-by-little for each frame sequentially from beginning to end [10]. This is known as the straight-ahead approach to animation [11]. The animators need to know exactly how the entire sequence should look beforehand in terms of timing and positioning, since they would not be able to go back and change the previous frames.

This project aims to address these challenges by developing a system that allows animators to use different approaches to create animations. Instead of creating each frame one after the other, this systems would allow them to use motion-capture videos or keyframes with automated inbetweening.

1.1 Background

1.1.1 Challenges of Traditional Stop-motion Animation

The drawbacks of the straight-ahead approach for stop-motion animation include the amount of time and labor required [12], the difficulties in animating realistic motion, and the inability to edit previous frames in the animation sequence [13].

Firstly, stop-motion animation is a repetitive and labor-intensive process that requires each frame in the animation to be created individually. Since the standard frame rate (fps) for animation is 24 fps [14], this would require animators to create 24 frames for every second of their animation, making it slow and costly to create animations that are minutes or hours

long. Additionally, professional animation studios, such as Laika, would need to create many copies of the same puppet with different facial expressions and carefully arrange the character movements to maintain smooth motion. This is a time-consuming process. For example, "the film *Wallace and Gromit: The Curse of the Were-Rabbit*, took over a year and a half with nearly 30 animators on the set each day to make." [12]

Secondly, it is difficult to animate physically accurate and natural-looking movement. According to animation professor, Stephen Baker, it is easier to animate large movements, such as characters fighting, than subtle movements that involve small shifts in posture, and beginners often have trouble getting the timing of the movements to look realistic. The animator would need to be very experienced to create animations of a character moving in a way that a human would, with the correct micro-movements and timing. Otherwise, the movements would look stiff or unnatural.

Lastly, if the animators want to change or insert frames in the animation, they would need to recapture the entire scene to make adjustments or fix errors [13]. The animator cannot simply go back and replace previous frames that need correction, since the recaptured frames would not align smoothly into the rest of the motion sequence. According to animator, Graham G. Maiden, "Even when we started shooting, we thought we had the finished *Corpse Bride*, but we found you still had to make little tweaks as you go along" [12]. Thus, the ability to make post-production edits would be very useful to animators and would reduce the time they need to spend planning, storyboarding, and reshooting the animation.

Because of these difficulties, a lot of skill and experience is needed to create professional stop-motion animations at studios such as Laika [15]. Existing software used by professional animators, including Dragonframe [10], does not address these challenges. New technologies that allow stop-motion animation to be created through other techniques would lower the barrier to entry, allowing more animators to explore this artform.

1.1.2 Other Animation Techniques

Stop-motion is just one form of animation, alongside 2D and 3D animation, and there are many techniques and approaches for creating animations. Among the 12 Principles of

Animation [16] are the straight-ahead and pose-to-pose approaches. The straight-ahead approach involves sequentially creating each frame in order from beginning to end, while the pose-to-pose approach involves creating keyframes (blocking) and filling in the intermediate frames (inbetweening). Additionally, motion-capture is another common approach that involves recording the movement of real-world actors to map onto characters in the animation. [17]

While 2D and 3D animation are often created using pose-to-pose animation techniques [18], traditional stop-motion can only be created using the straight-ahead approach. According to the article, *Reviewing and Updating The 12 Principles of Animation* [13], "Stop-motion animation by its very nature is produced with the straight-ahead technique and cannot be achieved by pose to pose as the in-betweens cannot be inserted later. The entire animation must be finished in one go."

Additionally, while software such as Cascadeur [19] can create animations using motion-capture to map AI-detected poses onto a character [20], this only works for virtual 3D models rather than physical figurines, so it is only used for 3D animation rather than stop-motion. Thus, stop-motion animations can only be created with the straight-ahead approach, greatly limiting the techniques available for the animator to use.

Unlike straight-ahead, the pose-to-pose approach allows the animators to easily edit and replace previous frames in the sequence. They can ensure that the changes result in smooth motion through onion-skinning [21], while allows them to compare previous and subsequent frames to the frame they're editing.

For 3D animation, commonly used software such as Maya and Houdini, can automatically create in-between frames from two keyframes for a 3D model [22]. This reduces the animator's workload since they would not need to create in-between frames manually.

In addition, the motion-capture approach addresses the challenge of animating realistic motion by using real actors to control the movement of the characters [23]. This greatly reduces the amount of time and labor required since the animator only needs to transfer the movement from a recorded video rather than creating the animation frame-by-frame.

These techniques are currently not available for stop-motion animation, so a system that could enable pose-to-pose and motion-capture approaches would eliminate the challenges of the straight-ahead approach.

1.1.3 Existing Generative AI Models for Animation

With the development of generative AI models, it is now possible to automate repetitive parts of the animation process, allowing animators to focus on the creative aspects.

Most of the existing research on generative AI for animation focuses on generating videos using text prompts [24][25], where the results would be non-deterministic and difficult to control. The animator would not be able to specify exactly how the character looks or the exact motion they would like the character to perform, since the text prompt would be too vague, making it difficult to obtain the desired result that they had in mind. Additionally, many text-to-video models produce results that have inconsistencies between frames, where the appearance of the same character would change throughout the video.

In addition to generative AI, computer vision models [26][27] have also emerged that could detect the pose skeleton from a photo or sketch, containing the position and orientation of joints and body parts. This makes it possible to capture the motion of an actor from a simple video without any specialized hardware traditionally used for motion-capture, such as motion-capture suits, reflective markers, or sensors [28]. These detected poses can then be used to transfer the actor’s movements to animate a figurine.

Lastly, pose transfer [29][30] is an AI technique that enables the transferring of a human pose from one subject to another, allowing the generation of images combining the physical characteristics of one subject with the pose of another subject. This can be applied to stop-motion animation by using pose transfer models to map real-world motion onto figurines, reducing manual workload while improving the realism of the animations.

Unlike video generation models where the output is controlled by a text prompt, the pose transfer method allows animators to control detailed aspects of the appearance and motion of their characters so they can produce exactly what they envisioned. By providing the option to use a video of a real person moving, this approach ensures that the generated animations

exhibit natural, lifelike movement that is difficult to achieve with traditional stop-motion animation.

1.2 Objectives

The goal of this research is to develop a generative AI pipeline for creating stop-motion animation through pose-to-pose and motion-capture techniques that allow users to edit and improve the output. Key contributions include automation of the repetitive work required for inbetweening in traditional animation, ensuring realistic-looking motion in the generated animation, and giving users the ability to edit and change the generated output.

The system would use a pose detection model to detect poses from videos of human actors or images of keyframe sketches. A pose transfer model would use photos of a figurine to generate frames in the animation of the figurine posed according to the detected poses. With these models, the interface would allow the animator to create stop-motion animations through the pose-to-pose or motion-capture approaches, reducing barriers and enabling faster workflows, so the animator could focus on artistic expression.

2. REVIEW OF EXISTING WORK

Many tools and systems have been developed to make the stop-motion animation process more efficient, such as streamlining the creation of physical puppets through 3D printing [31][32], or reducing the workload using digital 3D models [33]. With the advent of generative AI, more approaches are now available.

There are many existing AI video generation methods. The most popular ones are text-to-video approaches rather than pose transfer, where the only way to control the output is through a text prompt or reference images, so it is not possible to make exact changes. Existing pose transfer approaches focus on generating videos of photorealistic humans rather than stylized figurines and do not provide an easy way for the user to make changes and fix problems in the output. Our approach aims to address these limitations.

2.1 Text-to-video models

Text-to-video models [24][25], including Sora [34], Stable Video Diffusion [35], and AnimatedDiff [36], can be used to generate stop-motion animation by entering a text prompt describing the motion, along with reference images of the figurine to be animated.

Although this approach is widely used for generating AI videos, it would not be able to generate the exact video that the user wants. For example, a text-to-video model can generate an animation of Spiderman walking, but it would be unable to animate a specific figurine of Spiderman walking in a specific way. This would limit the animators ability to control the work they produce and express themselves creatively.

Additionally, while the animator could input reference images of the figurine into the Stable Video Diffusion model [35], the quality of the output would be low, resulting in a lot of artifacts and unwanted extra movements in the generated animation as seen in Figure 2.1.



(a) Input



(b) Generated

Figure 2.1. Result from Stable Video Diffusion using reference images of a figurine

2.2 Motion-controllable models

With the recent release of the Go-with-the-Flow model [37], it is possible to generate animations with video diffusion models where the animator could control the motion of an object by moving it around in an image.

However, with this approach, there are distortions and artifacts in the generated video. It also adds unwanted motion to other parts of the image, such as the splashing and dripping water shown in Figure 2.2.

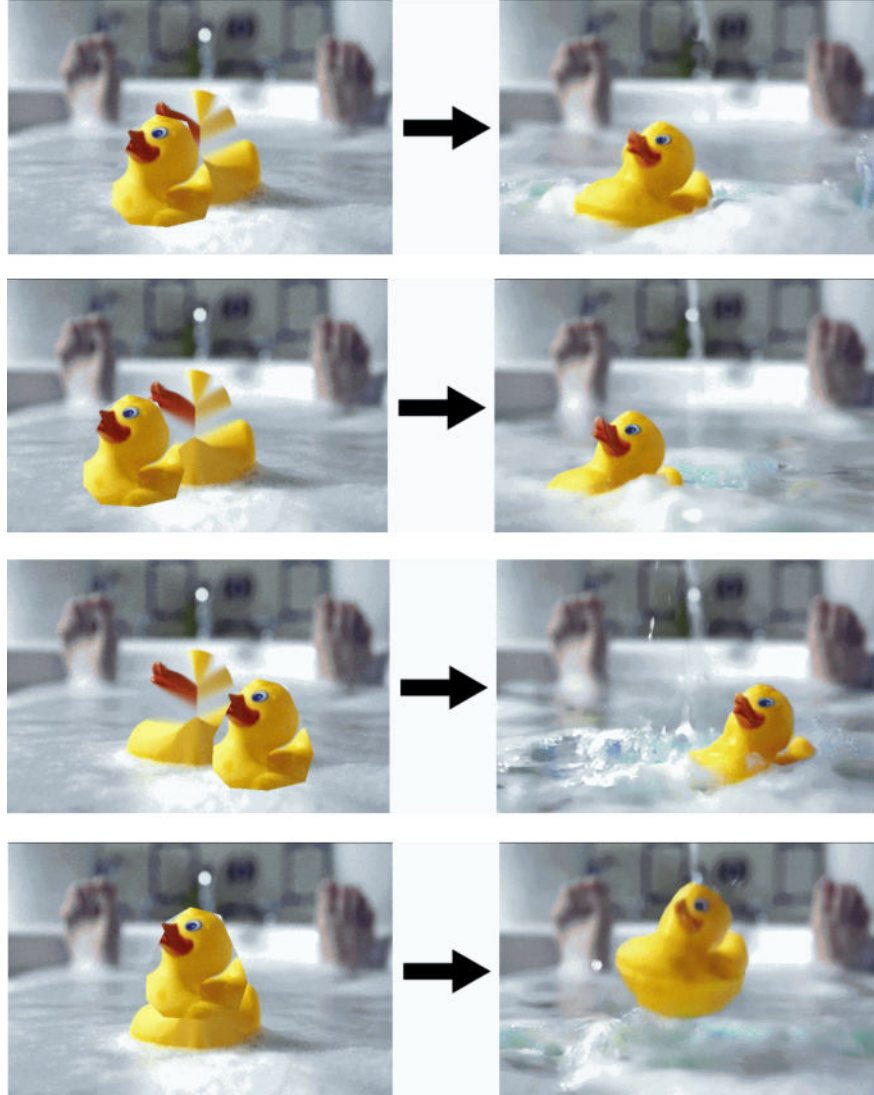


Figure 2.2. Input and output from the Go-with-the-Flow model [37]. There are distortions on the ducks face when it rotates to face the camera. The model also adds water dripping down in the background which wasnt specified in the input.

2.3 Frame Interpolation models

Frame interpolation models interpolate between two images to generate an animation which starts with the first input image and ends with the last input image. This could be applied to stop-motion animation by allowing the animator to only create the keyframes and use the frame interpolation model to generate the frames in between.

Frame Interpolation for Large Motion (FILM) [38] is a model that could accurately interpolate two images. However, the result only looks accurate if the difference between the start and end images are not too large. Otherwise, the result would be distorted, as seen in Figure 2.3. It is also difficult to ensure that the movements look natural, since it moves from the start to end poses at constant speed, whereas a human would speed up and slow down when they move.

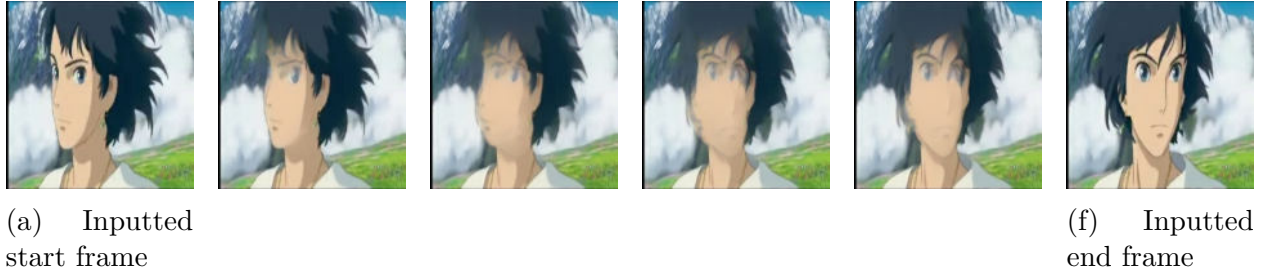


Figure 2.3. Result from the FILM model

The paper *Interpolating between Images with Diffusion Models* [39] describes a frame interpolation approach that focuses on interpolating between images that are very different, which should address the limitations of the FILM model. However, the interpolated frames do not retain the appearance of the character from the inputted images and look inconsistent between frames, as seen in Figure 2.4.



Figure 2.4. Result from Interpolating between Images with Diffusion Models [39]

From these results, the existing approaches that use frame interpolation are unsuitable for stop-motion animation unless the user inputs many keyframes in the animation. Our approach aims to address these limitations, allowing it to work with fewer inputs and generate animations with more realistic motion.

2.4 Pose transfer approaches

Meta AI Animated Drawings [40] animates a user-inputted image of a character using a motion reference video. It detects the skeleton in the inputted image and matches it to the poses in the video. At a high level, this approach is similar to our approach. However, the results look very distorted, as seen in Figure 2.5, and it does not work for 3D rotations such as head turns or limb movements.



Figure 2.5. Result from Meta AI Animated Drawings [40]

Compared to Meta AI’s approach, Text2Video-Zero [41] is capable of generating animations with fewer distortions. This approach transfers the style from the character reference image to the contours of the motion reference video to output an animation of the character moving according to the inputted video. However, the results do not retain the exact physical characteristics of the reference image and add unwanted changes such as the red background in Figure 2.6.



Figure 2.6. Input and output of the Text2Video-Zero model [41]

Our approach aims to produce higher-quality results than Meta AI Animated Drawings and Text2Video-Zero by generating animations that closely match the physical characteristics of the figurines and minimize distortions in the output.

2.5 Diffusion models for pose transfer

Recent advances in image-to-image diffusion models have significantly impacted how well images or videos can be generated from input images. There are models that can transfer the physical characteristics of one person to the pose of another person [42][43][44][45][46], make local changes to the appearance of a person in a photo [47], generate animated videos from a still image [48][49][50][51][52], and generate videos of realistic humans based on the motion of a person in a video [53][53]. For example, Dreambooth [54] can generate images of a character in different poses and scenarios using a few images of the character as input. The *Everybody Dance Now* framework [55] can transfer the motion of a person dancing to the video of another person. The liquid warping GAN [56] is a framework for motion imitation that can transfer the physical characteristics or background of the source image to the pose, perspective, or physical characteristics of the target image. PCDMs [29] and CFLD [30] are models that take a reference image of a person and an image of the desired pose and output an image of the same person positioned according to the input pose. Finally, Animate-Anyone 2 [57] is a state-of-the-art model that could transfer poses that accurately maintain the appearance of the character and generate animations with smooth and realistic motion.

These methods show how diffusion models can be used for animation. They could be leveraged into a framework for generating stop-motion animation using photos of the figurine as the character reference image and a video or sketches as the pose reference frames, running the pose transfer model for each pose to generate all of the frames in the animation.

Despite all the available models, there is still much room for improvement. Many of the existing pose transfer models can only generate videos of photorealistic humans instead of figurines or drawings of characters since they are only trained on images and videos of real humans. An out-of-the-box model would not work well for unrealistic and unique-looking figurines, since they have not been trained on images of the figurine.

Additionally, many of these existing approaches struggle with inconsistencies between the frames in the generated animation without providing a way for users to control and modify the output. Even for state-of-the-art models such as Animate-Anyone 2, if there is a problem in the output, such as distortions or inconsistencies, the user does not have a way to easily improve it. With our approach, the user can simply add more reference images or edit the detected poses for more accurate results. This way, they can make changes to the generated animation until they are satisfied with it.

Our approach will address many of the issues in the existing work, such as retaining details from the character reference images and preventing distortions, along with providing tools for the user to modify the output.

3. METHODOLOGY

The pipeline for generating a stop-motion animation of a figurine uses only photos of the figurine (the "character reference images") and either a video of a human actor or sketches of the keyframes (the "pose reference frames"). The main mechanism involves pose transfer using a PCDMs model [29] that is finetuned to be specialized for the figurine. The chart in Figure 3.1 describes the pipeline.

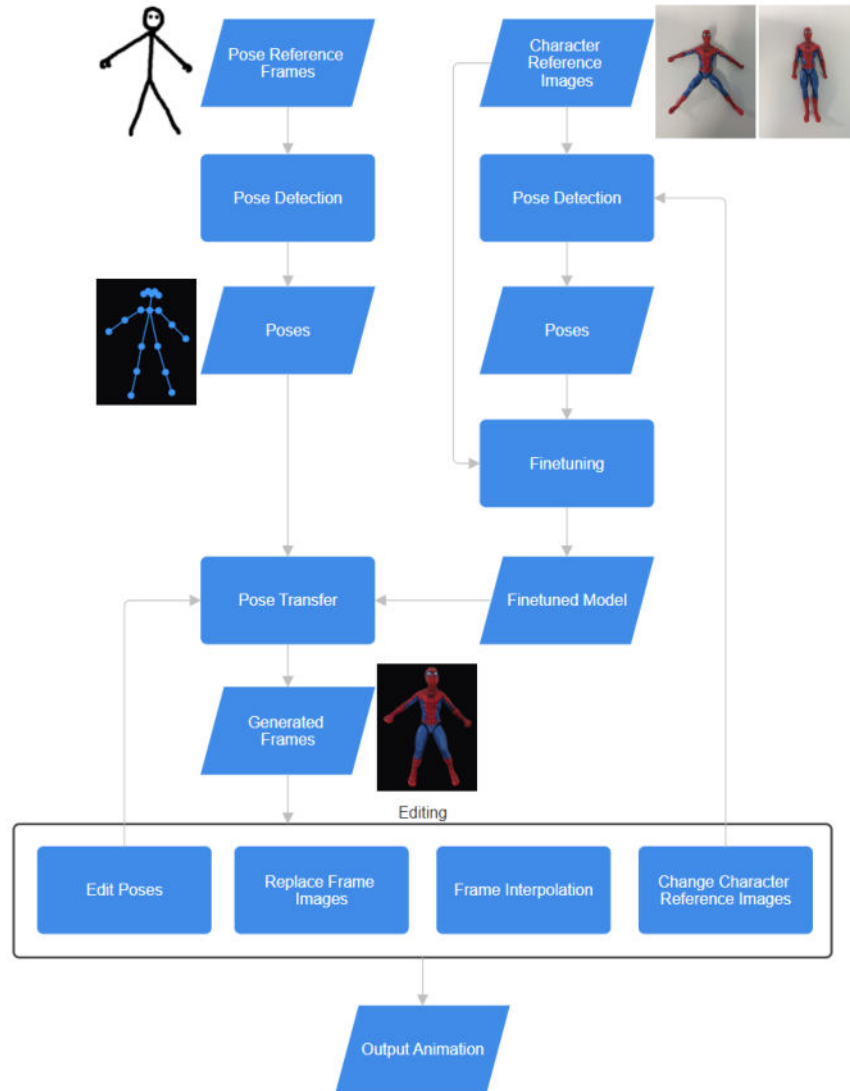


Figure 3.1. Flowchart of the system pipeline

In Figure 3.1, the input to the system in this case would be sketches of stick figures ("pose reference frames") and two photos of a Spiderman figurine in poses that look similar to the key poses of the desired animation ("character reference images"), specifically with the limbs together and limbs apart. The system detects the poses in each pose reference frame. It also finetunes the pose transfer model using the character reference images and their detected poses. Finally, the finetuned model would transfer the physical characteristics of the figurine to the poses from the pose reference frames, generating the keyframes of the animation. The user can then edit the generated frames to fix inaccuracies or obtain the intermediate frames.

The pipeline could be described with the following pseudocode:

Algorithm 1 Pseudocode of the system

```

1: function ANIMATE(poseRef, charRef)
2:   poses  $\leftarrow$  POSE_DETECTION(f), f  $\in$  poseRef
3:   model  $\leftarrow$  IMPORT_PRETRAINED_PCDMS
4:   for img  $\in$  charRef do
5:     pose  $\leftarrow$  POSE_DETECTION(img)
6:     model  $\leftarrow$  FINETUNE(charRef0, pose, img)
7:   end for
8:   output  $\leftarrow$  POSE_TRANSFER(model, charRef0, pose), pose  $\in$  poses
9:   return output
10: end function

```

In Line 2 of the pseudocode, the system detects the poses in each pose reference frame.

In Lines 4 to 7, it finetunes the PCDMs model [29] using the character reference images. Specifically, it trains the model to generate each character reference image using the first image, *img*₀, and the detected poses of the other character reference images, *pose*(*img*_i), as inputs to the model. It would then use *img*_i, the original character reference image corresponding to the pose, as the expected output. This would train the model to generate

results that look the same as each target image, img_i , when given img_0 and the detected pose of img_i .

Finally, in Line 8, the system uses the finetuned model for pose transfer using only the first character reference image, img_0 , and the poses from the pose reference frames to generate frames of the animation.

After the animation is generated, each frame is displayed as a timeline in the interactive user interface. The animator can select each frame to view and edit.

3.1 User Input

To generate a stop-motion animation, the animator needs to input a set of photos of the figurine ("character reference images") and a video or a set of sketches specifying the desired motion of the figurine ("pose reference frames").

3.1.1 Pose Reference Frames

The animator can input either a video of an actor moving or a set of rough sketches of the key poses of the desired animation.

The video does not need to be a professional motion-capture video with specialized equipment. It would simply be a video of a person moving, since the pose detection model [27] could detect the poses from the video directly. The only limitations are that there should only be one person in the video, the person should not wear baggy clothing, there should not be clear shadows of people in the background, and the background should not look too messy. Otherwise, the pose detection model would fail to detect the correct pose.

If the animator decides to input sketches of the key poses instead, the sketches do not need to look like the figurine. Even stick figures would work, as long as the pose is clearly discernible.

3.1.2 Character reference images

The photos of the figurine should be in various angles and poses that are similar to the poses in the desired animation (See Figure 3.2 for an example). There should be at least two

images, with additional images needed for more complex motion, since one of them is used as the source image and the others are used as the target images in the finetuning stage. For example, to create a side-view animation of a figurine walking, it would only require two side-view images of the figurine with the limbs together and limbs apart. In contrast, for more complex motion such as dancing, it would need images of the figurine posed according to each main pose in the dance and viewed from different angles. The poses of the figurine do not need to match the key poses in the pose reference frames exactly, but they should look similar.

If the character reference images are insufficient, for example, if the character should be seen in front-view in the desired animation, but only side-view photos of the figurine are provided, the resulting animation would fail to show the character from the desired angle and would instead show the character from the perspectives of the provided reference images distorted to fit the poses.



Figure 3.2. Example input images of a figurine

3.2 Pose detection

Pose detection models, such as OpenPose [26] or DWPose [27], can detect the pose of a person from a photo or video. The pose consists of 18 keypoints on the persons body, such as the waist, shoulders, knees, etc, and 22 keypoints on the hands. Figure 3.3 shows an example of pose detection.

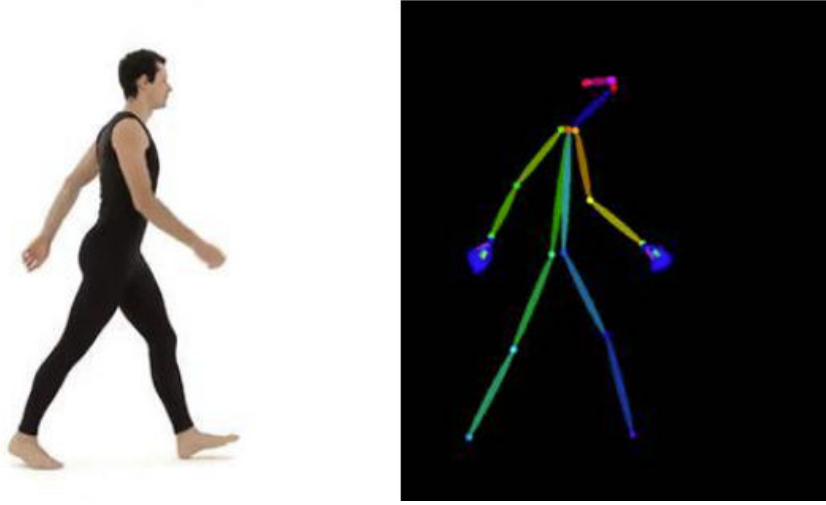


Figure 3.3. Pose detection of the motion reference video

Our pipeline uses the more accurate DWPose [27] to detect the poses in each pose reference frame along with all of the character reference images. The poses from the pose reference frames would be used by the pose transfer model to specify the poses of the frames in the output animation, and the poses from the character reference images would be used for finetuning the pose transfer model to map each pose to the original reference image.

Since DWPose only works for humans figures, the subject in the image needs to be humanoid, but the image can be stylized in different art styles instead of a realistic photo, and the subject can be wearing elaborate costumes or have unrealistic appearances.

In Figure 3.4, the pose detection models work well even for complex poses with a messy background.

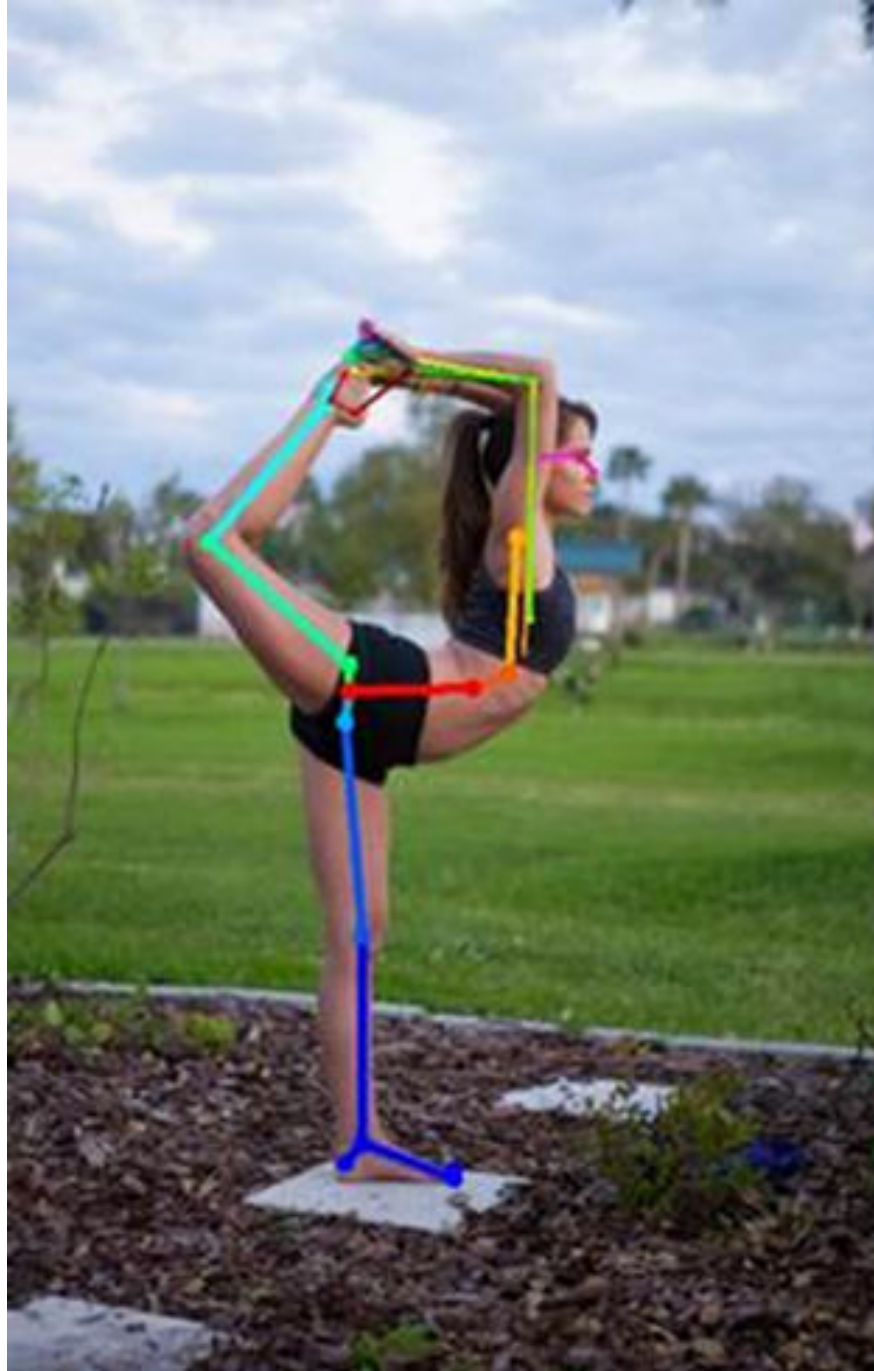


Figure 3.4. Pose detection for a more complex pose

However, if there is a clear shadow in the background as seen in Figure 3.5, it would detect the shadow as a second person in the image.

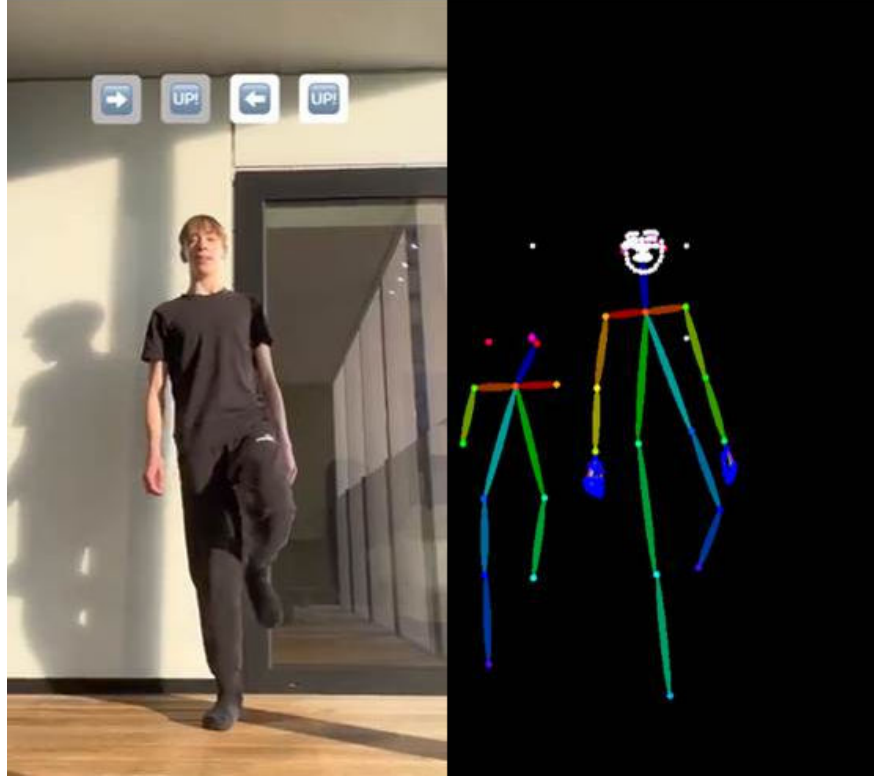


Figure 3.5. Pose detection failure due to shadow

Therefore, pose detection would not work well if the video has a clear shadow and the actor in the video should avoid wearing baggy clothing for better results.

3.3 Finetuning

The finetuning stage trains the PCDMs pose transfer model [29] to be specialized for the figurine that is being animated. This is a conditional diffusion model that takes two images as input (a detected pose and a character reference image) and outputs the character posed according to the detected pose.

The model was pretrained on the DeepFashion dataset [58], consisting of photos of real humans dressed in casual clothing seen from various angles. Since the figurines would look different from the images that the model was originally trained on, the pose transfer results would not look accurate without finetuning, especially if the figurine is stylized to not look

like a realistic human. It would try to generate an output image that looks similar data it was trained on and fail to include the physical characteristics of the figurine.

To finetune the model, the gradient descent training algorithm uses the first character reference image and the detected poses of the other character reference images as inputs to the pose transfer model. The model would generate an image of the character posed according to each input pose. The loss function calculates the image similarity (SSIM) score between the generated image and the original character reference image corresponding to each pose and adjusts the weights of the model to generate images that look more similar to the expected output.

The final model should be able to generate images of the character in poses that look similar to the ones it was finetuned on. This step causes the output to look more like the ground-truth images provided during finetuning.

3.4 Pose transfer

The pose transfer stage generates the frames of the output animation. It uses the finetuned PCDMs model [29] to transfer the poses of each pose reference frame onto the first character reference image. Since the model was already trained during the finetuning stage to transfer poses from the character reference images that look similar to the poses from the pose reference frames, the results should look accurate.

Each frame generated from the pose transfer model would become a frame in the final animation. Once pose transfer is applied to all pose reference frames, The system outputs the generated animation frames to the frontend to display them in the user interface.

3.5 User Interface

The user interface shown in Figure 3.6 consists of three panels and a timeline of frames from the animation. This would allow the user to easily manage assets, modify poses, and edit individual frames of the generated animation. The production process would be split into separate steps so the system would not have to reprocess the entire pipeline every time the user makes changes in one of the steps.

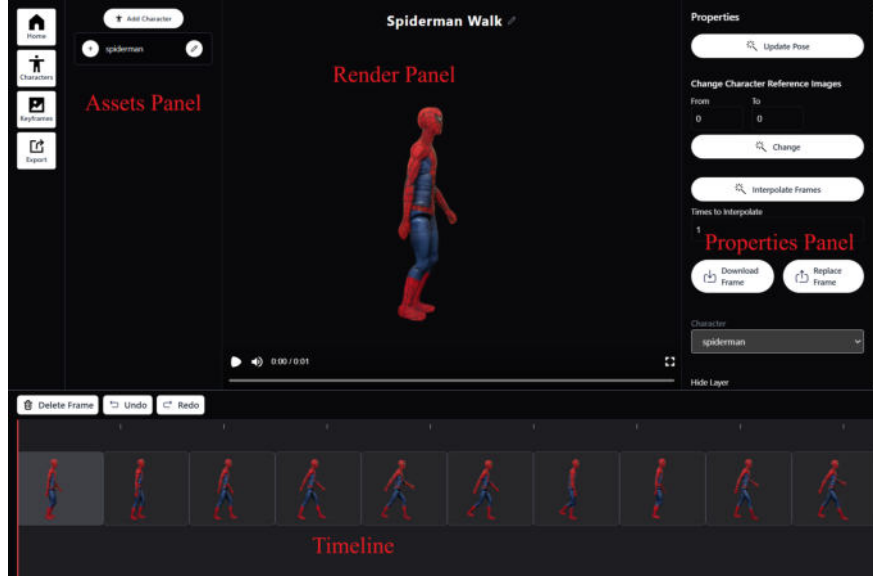


Figure 3.6. The user interface containing the Timeline, Assets Panel, Render Panel, and Properties Panel

3.5.1 Timeline

The timeline shows a row of keyframes for each layer in the animation. It is designed to resemble the interface of video editing software so the user could intuitively understand how to use it. They could select frames from different layers to edit or playback the animation.

Since the pose transfer model only works when there is only one figurine, having multiple layers makes it possible for there to be multiple characters in the same scene, where each layer contains the generated animation of one figurine. This is shown in Figure 3.7

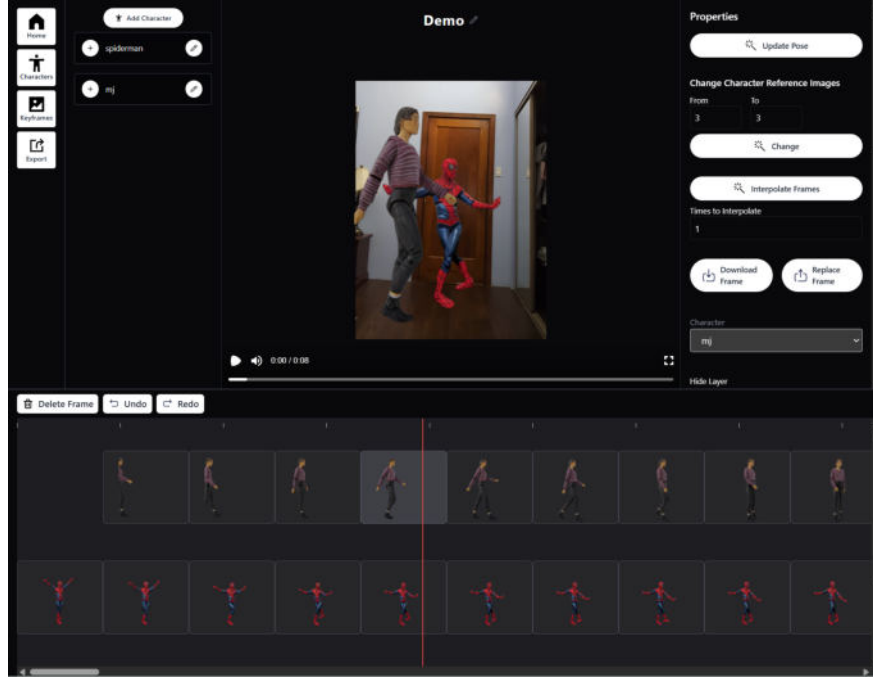


Figure 3.7. The user interface with multiple layers in the Timeline

3.5.2 Assets Panel

The assets panel contains all of the figurines that the user can add to their animation. When they click on an asset, they would have the option to add character reference images to it or generate a new animation with it, shown in Figure 3.8. Each time the user adds a new asset or adds more reference images, the system will finetune the pose transfer model with the new images.

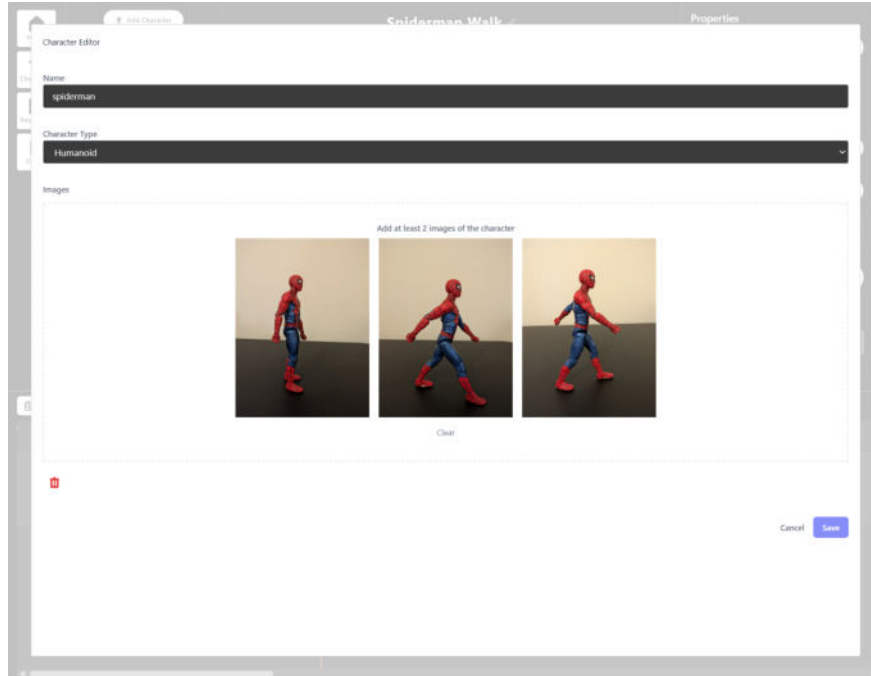


Figure 3.8. User interface for adding character reference images

Since finetuning is the slowest step in our pipeline and is independent from the pose transfer stage, this allows our system to reuse the finetuned model across different animations and only run this step when new character reference images are added.

Additionally, the user can also add images and videos directly to the system through the assets panel shown in [Figure 3.9](#).

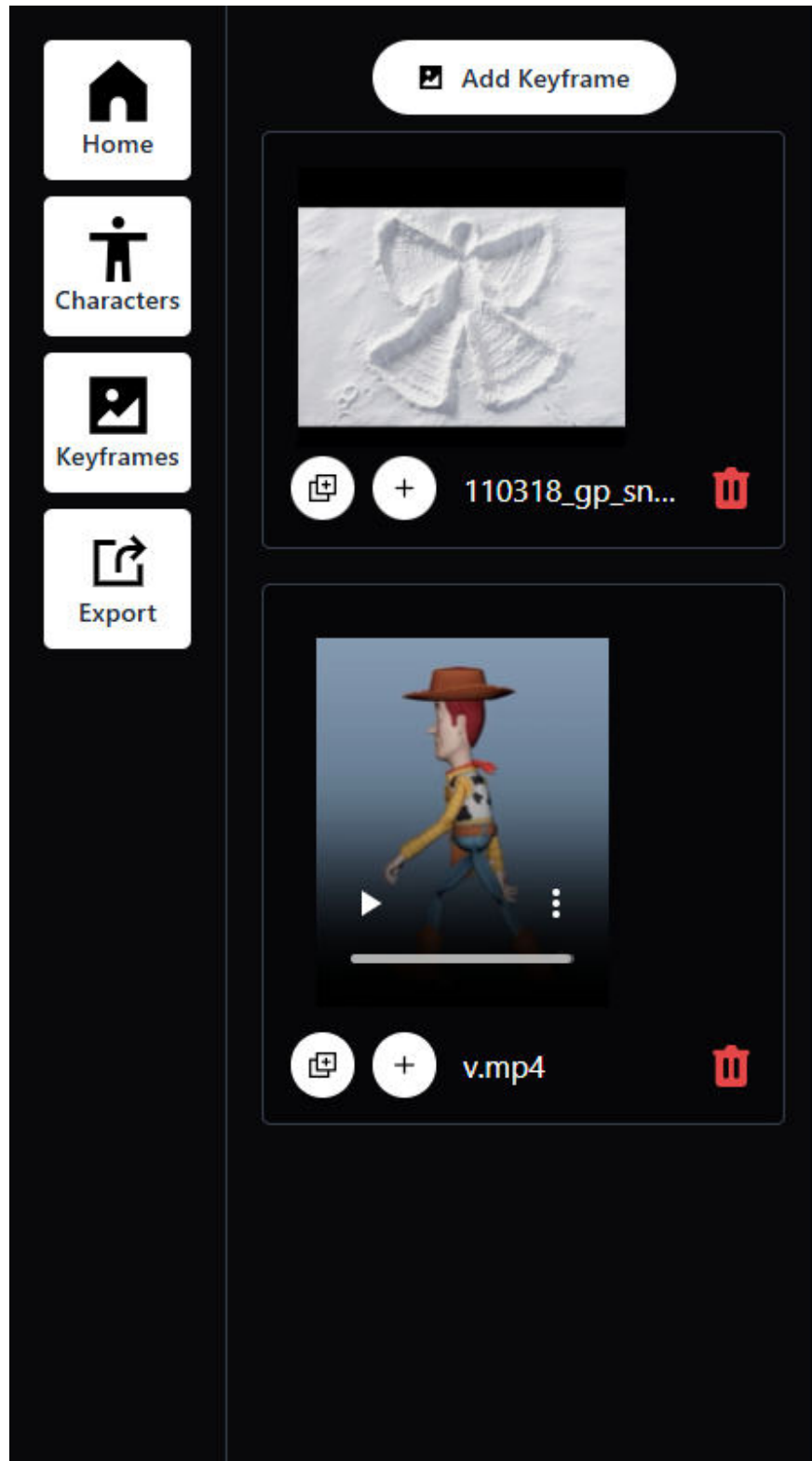


Figure 3.9. Adding images and videos in the Assets Panel

To generate an animation of the character, the user can upload pose reference frames to the character, such as a motion-capture video or sketches of keyframes. This is shown in Figure 3.10 and Figure 3.11. This would run the pose transfer model and output the generated frames onto the timeline.

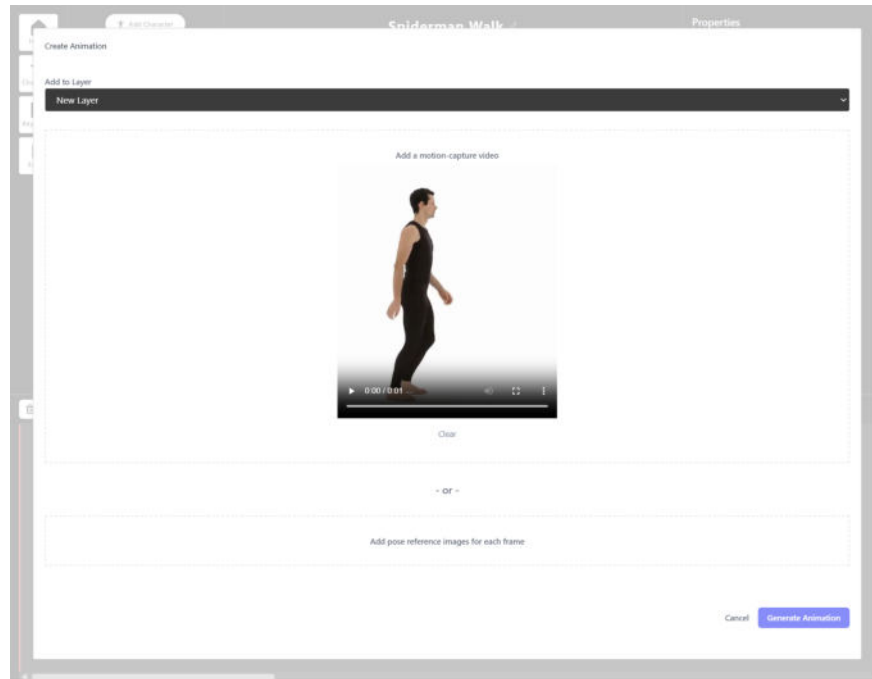


Figure 3.10. Uploading a motion-capture video as pose reference

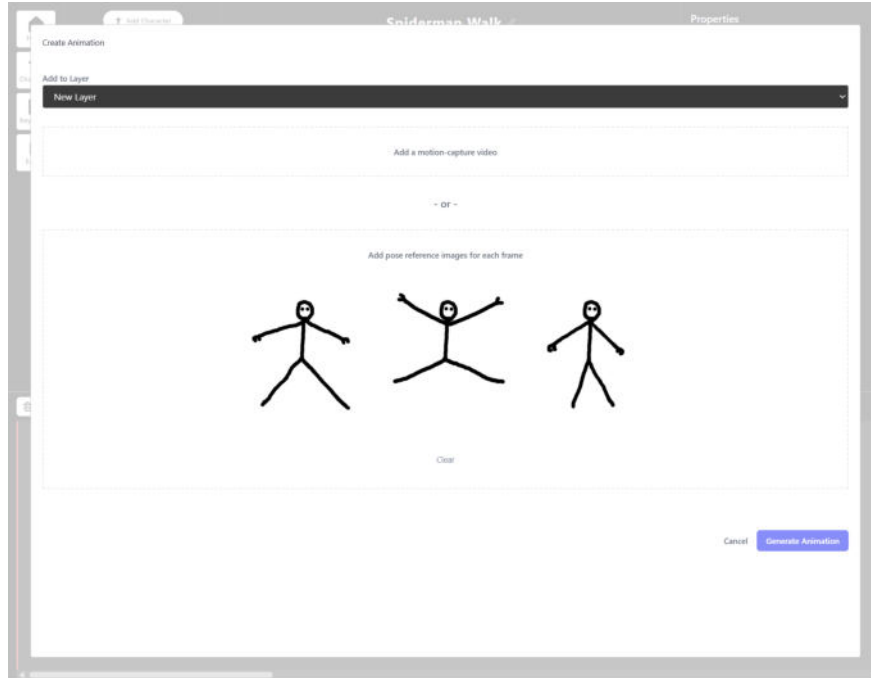


Figure 3.11. Uploading a rough sketches as pose reference

3.5.3 Render Panel

The render panel displays the generated animation. The user can play the animation to check the result and identify frames that they are not satisfied with.

In addition to the generated animation, the user can toggle the display of the detected pose for that frame, the original pose reference frame, and the onion-skinning of previous and subsequent frames. This is shown in Figure 3.12.

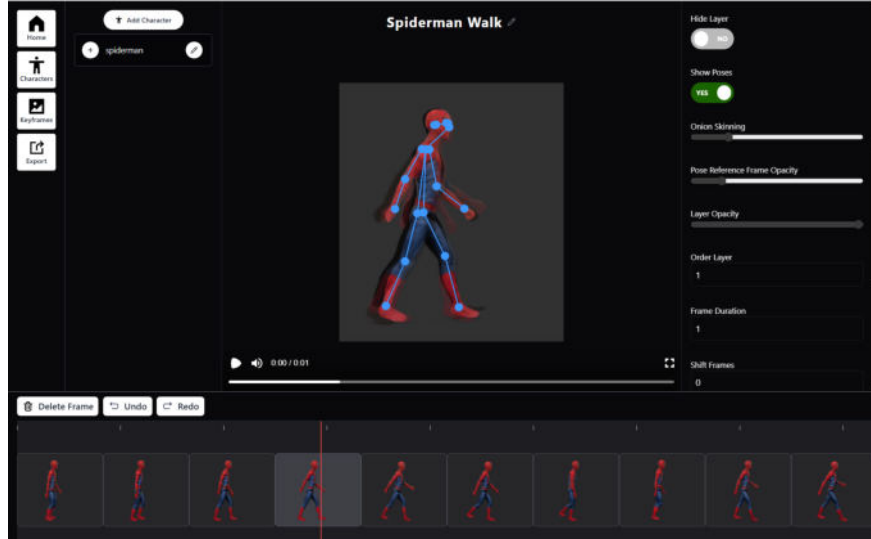


Figure 3.12. The generated frame overlayed with the detected pose, the original pose reference frame, and the onion-skinning of previous and subsequent frames

The pose that is overlayed over the animation is editable, so the user can modify the pose or fix incorrectly detected poses by dragging points on the pose skeleton. Once the user is satisfied with the changes, they can regenerate the frame with the new pose.

With the original pose reference frames overlayed over the animation, the user can compare the reference with the generated result to make corrections.

Finally, onion-skinning would allow the user to compare the current frame with the previous and next frames so changes would fit smoothly with the rest of the animation.

3.5.4 Properties Panel

The properties panel shown in Figure 3.6 provides an interface for the user to edit the generated animation. For example, they can regenerate frames using modified poses or difference character reference images, save and replace specific frames, duplicate and remove ranges of frames, and generate interpolated frames between the current and next frame.

The user can regenerate a frame after modifying the pose. This allows the user to fix incorrectly detected poses or make slight changes with onion-skinning to duplicated frames to generate inbetween frames. Additionally, the user can regenerate with new character

reference images, fixing incorrectly generated frames by providing character reference images that look more similar to the desired result.

By providing the ability to replace specific frames, the user can fix image generation artifacts and distortions. The user can simply download the frames, open it in an image editor to fix the inaccuracies, and upload the image to replace the frame. This gives the user full control over the output to make changes directly to the frames in the animation if other methods fail.

Finally, the interface provides a feature that allows users to interpolate between two frames. This would use the Frame Interpolation for Large Motion (FILM) model [38], which takes two frames as input, and outputs a user-specified number of inbetween frames. With automatic inbetweening, the user could quickly and efficiently create animations with the pose-to-pose approach, without needing to manually create each frame. In addition, this would ensure smooth transition between poses and facial expressions.

4. RESULTS

The performance of our system is evaluated qualitatively using a few representative examples, and quantitatively using image and video quality metrics.

From the metrics, our system performs significantly better than the out-of-the-box PCDMs pose transfer model [29] when animating figurines with a wide variety of physical characteristics, art styles, and complex motions.

The results from the representative examples demonstrate that, with finetuning, our system can generate animations for figurines that have different body types and physical characteristics, even abstract or stylized figurines, and the results can improve if the user inputs more character reference images. It can also animate complex motions, such as ballet. However, it struggles when the figurine has large additional features or if the pose reference video isn't clean enough for pose detection to work correctly.

4.1 Representative Examples

These examples were selected to demonstrate the generalizability of our system across different physical characteristics of the input figurine and complexity of the desired movements to animate.

4.1.1 Simple cases

To evaluate basic functionality, the input for the initial case consists of photos of a simple figurine that looks like a realistic human, shown in Figure 4.1, and a simple video of a man walking as the motion reference video, shown in Figure 4.2.



Figure 4.1. Input images of a realistic human figurine



Figure 4.2. Frames from the input motion reference video of a man walking

Since the existing model was trained on photos of real humans, the baseline results shown in Figure 4.3, which uses the PCDMs model [29] out-of-the-box, look decent even without finetuning. The output looks somewhat similar to the input figurine and the motion looks accurate. However, there are some distortions and artifacts and the physical characteristics are not consistent between frames.



Figure 4.3. Baseline results without finetuning

The results after finetuning in Figure 4.4 look better. Our system was able to transfer the style and texture of the figurine into the generated animation. There are fewer distortions and the appearance of the figurine remains consistent between frames.



Figure 4.4. Results with finetuning

This shows that our system works well for simple cases and finetuning noticeably improves the results.

With a less realistic humanoid figurine as input, specifically a figurine of Spiderman seen in Figure 4.5, the out-of-the-box model cannot generate an accurate animation without finetuning. This is because the figurine looks very different from the photos of real humans on which the model was pretrained. From the results in Figure 4.6, the out-of-the-box model

failed to generate an animation that looks similar to the figurine. There is a lot of distortion and the appearance is inconsistent between the frames.



Figure 4.5. Input images of a Spiderman figurine



Figure 4.6. Baseline results without finetuning

The results look much better after finetuning the model using the two photos of the figurine in Figure 4.5. The output shown in Figure 4.7 looks similar to the input figurine, there are no distortions or artifacts, and the details remain mostly consistent between frames. One noticeable problem is the inaccuracy in the motion of the arms and legs. This could be improved with a larger number of character reference images for finetuning.



Figure 4.7. Results with finetuning

In Figure 4.8, using three photos of the figurine instead of two, the swinging of the arms and legs looks more accurate. This shows that the user could improve the result by regenerating the animation using more character reference images if they're not satisfied with the previous result.



Figure 4.8. Results using three input images instead of two

Comparing the results from the baseline out-of-the-box model (Figures 4.3 and 4.6) to the results from our system (Figures 4.4 and 4.7), it is clear that the finetuning step in our pipeline greatly improves the quality of the generated animation.

4.1.2 Complexity of physical characteristics

With a Hulk figurine (Figure 4.9), which has a much larger body, the differences in body type can be transferred to the generated animation. In Figure 4.10, the output mostly maintains the bulky figure of the original figurine. The only drawback is that the face and the texture lack detail. This could be improved with more character reference images, finetuned for more iterations.



Figure 4.9. Input images of a Hulk figurine



Figure 4.10. Results for different body types

A Spiderman figurine holding a hammer (Figure 4.11) is used as input to generate the animation shown in Figure 4.12. The hammer does not stay in the figurines hand consistently and instead appears and disappears between frames. This is because our system was unable

to recognize that the hammer was a part of the figurine. Our system cannot handle large items added to the figurine, so the user will need to animate those items separately.



Figure 4.11. Input images of a figurine with an additional feature



Figure 4.12. Results for a figurine with an additional feature

An artist mannequin was used as input to generate the animation shown in Figure 4.13. Despite looking very abstract and unrealistic without a face or clothing, the system can still detect all of the body parts correctly and generate the animation in Figure 4.14, demonstrating that the system works well for figurines with fewer features.



Figure 4.13. Input images of an abstract figurine

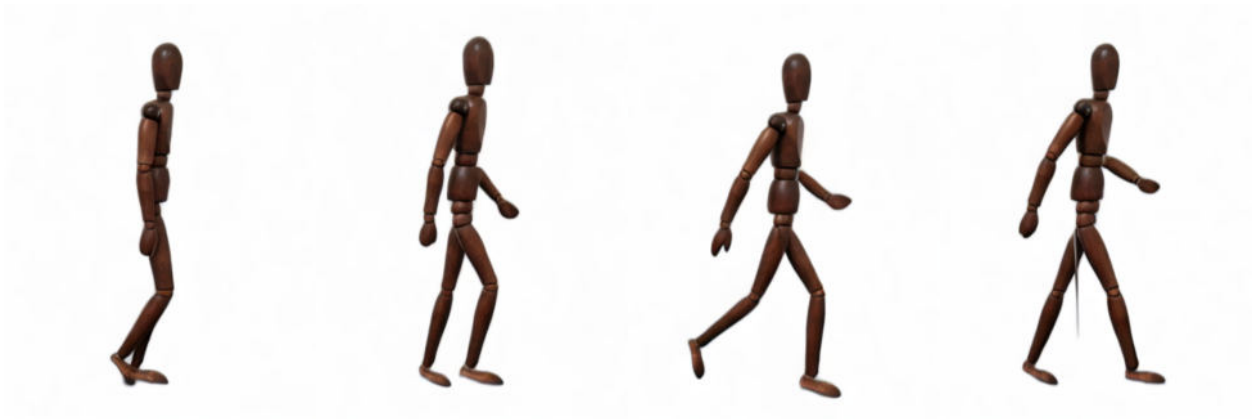


Figure 4.14. Results for an abstract figurine

These examples show that the system is generalizable across a diverse variety of figurines. The cases where the system fails are when the figurine does not look human-like enough or if the additional features are too large.

4.1.3 Complexity of motion

In this case, a video of a ballet performance (Figure 4.15) was used as the motion reference video. Unlike the video of a man walking in the previous cases, the movements are larger and more unusual, and the background was not removed. Additionally, nine photos of the Spiderman figurine were used as input, posed to resemble the key poses of the pose reference video (Figure 4.16).



Figure 4.15. Frames from a video of a ballet performance to input as the motion reference video



Figure 4.16. Four of the nine input images of a figurine in different poses

The result in Figure 4.17 looks very accurate, showing that the system works well with videos that contain motions that are more complex than the video of a man walking. It can capture the details of the figurine and generate a smooth animation. There are a few inaccurate frames due to incorrectly detected poses. This could be resolved using a cleaner pose reference video or user intervention to manually edit the poses in the few frames where pose detection failed.



Figure 4.17. Results for complex motion (ballet)

Thus, our system works well for videos of a person performing complex movements as long as the background is clean enough to detect the poses accurately.

4.2 Quantitative Evaluation

To obtain quantitative results measuring the accuracy of the generated animations, an evaluation dataset was created containing diverse inputs and expected outputs, and a set of metrics were selected based on what was used to evaluate similar systems.

4.2.1 Evaluation Data

The dataset for the quantitative evaluation consists of 52 animations of humanoid 3D models. The poses and physical characteristics are diverse to evaluate the generalizability of our system. It is available at: <https://huggingface.co/datasets/acmyu/KeyframesAI-eval>.

Existing pose transfer models used datasets that consist of photos or videos of real humans, such as the DeepFashion dataset [58], to evaluate their performance. Since our system is used for figurines that may not look like realistic humans, a new dataset needed to be created because an extensive dataset containing only humanoid figurines does not exist yet.

This evaluation dataset consists of 3D character animations collected from ArtStation (see Table A.3 for the attributions), specifically videos of humanoid characters walking, dancing, or performing various motions. This 3D animated character dataset is applicable

to stop-motion animations of figurines since photos of figurines look similar to rendered 3D models.

Each item in the dataset consists of the original animation frames, detected poses for each frame, and between 3 to 10 manually selected keyframes.

When evaluating our system using this dataset, the manually selected keyframes from each video are inputted as the character reference images, and the detected poses from the entire video are used as the pose reference frames. The original frames from the entire video would be used as the expected output to compare with the generated output.

For example, the dataset contains an animation of the walk cycle of a 3D model of the Toy Story character, Woody, shown in Figure 4.18. During evaluation, the detected poses from this animation are inputted to our system as the pose reference frames, and the original frames from this animation are used as the ground-truth data.



Figure 4.18. Evaluation data example: Walk cycle animation

A small number of keyframes extracted from this video, such as the ones in Figure 4.19, are used as character reference images.



Figure 4.19. Evaluation data example: Input images

Our system would attempt to generate the same animation as the original video and compare the generated frames with the frames in the original video. The aim is to increase the similarity between the generated animation and the original video.

To demonstrate that our system is generalizable enough to work on any figurine and any pose, the dataset consists of characters with diverse body shapes, costumes, and art styles, along with animations of the characters performing more complex movements with different camera angles. Figure 4.20 displays a few examples of characters in the dataset.

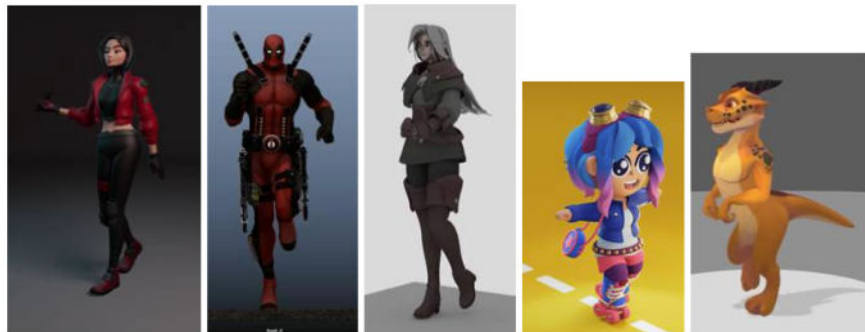


Figure 4.20. More examples from the evaluation data

4.2.2 Evaluation Metrics

For a generated animation to be considered high-quality, the criteria includes the cleanliness of each frame (in other words, fewer distortions), the similarity between the generated and ground truth frames, the accuracy of the figurine’s appearance in the generated frames, and the consistency between frames. There are existing metrics that can quantitatively assess each of these criteria.

The existing image and video quality metrics to measure these criteria include SSIM [59], PSNR [59], LPIPS [60], and FVD [61], which compare the generated frames with the ground-truth frames. These metrics are frequently used to evaluate existing pose transfer models such as PCDMs [29] and Animate-Anyone 2 [57].

- SSIM (Structural Similarity Index) compares two images based on luminance, contrast, and structure. This measures the criterion of how similar the generated frames are to the ground truth.

- PSNR (Peak Signal-to-Noise Ratio) is the difference between the original and the compressed and reconstructed images to measure image fidelity, such as noise and distortion. This determines the image quality of each frame.

- LPIPS (Learned Perceptual Image Patch Similarity) uses features from neural networks, such as AlexNet, to compare semantic content and textures. This measures the accuracy of the figurine’s appearance.

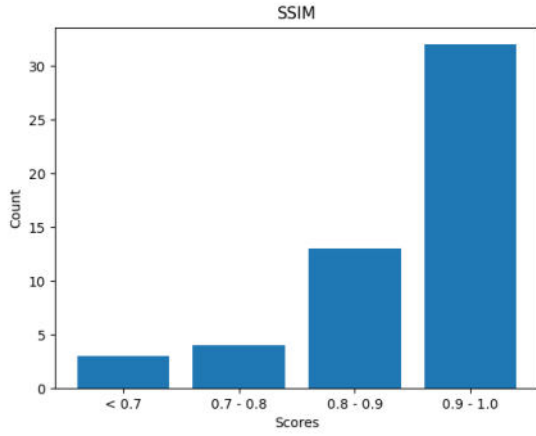
- FVD (Fréchet Video Distance) uses the entire video instead of individual frames. It compares the features between the ground truth and generated videos to measure how realistic each frame looks and the consistency between frames.

The baseline metrics are calculated for images generated using the out-of-the-box PCDMs model [29] without finetuning. This determines how much the steps in our pipeline improve the generated animation compared to using the existing model directly.

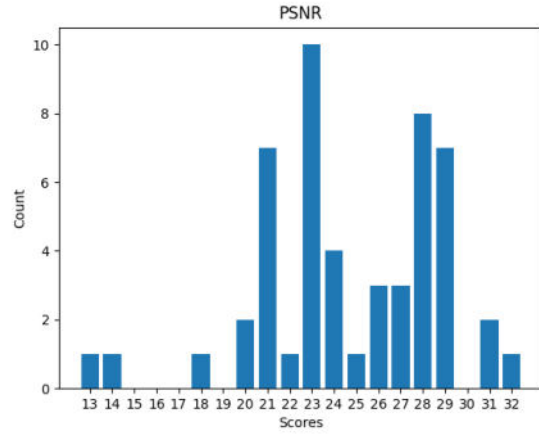
4.2.3 Quantitative Results

Table 4.1. Results

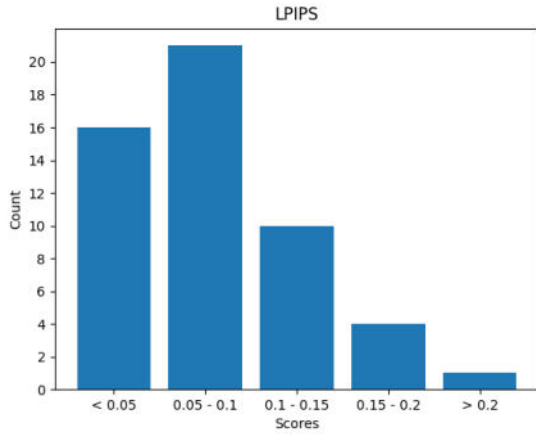
	SSIM	PSNR	LPIPS	FVD
Animate-Anyone 2	0.931	38.49	0.044	81.6
CFLD	0.7478	17.64	0.1819	-
PCDMs	0.7601	-	0.1475	-
Baseline (out-of-the-box)	0.7944	19.55	0.2534	219.2
Our system	0.8821	25.23	0.0829	108.1



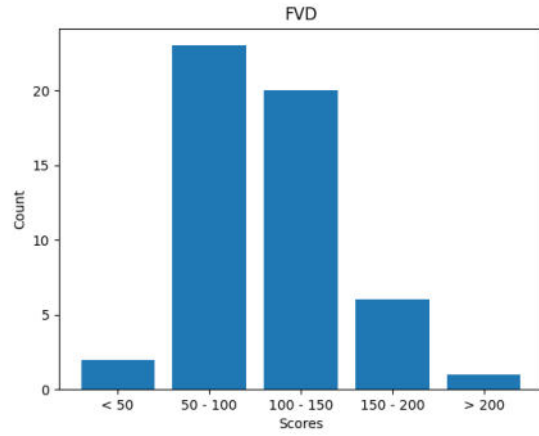
(a) SSIM



(b) PSNR



(c) LPIPS



(d) FVD

Figure 4.21. Distribution of evaluation metrics scores

The results in Table 4.1 compare the metrics for our system with the baseline out-of-the-box model and other existing pose transfer models. The results are obtained by running our system on the evaluation data set that contains 52 3D animations of characters with various appearances.

The metrics for our system are significantly better than the baseline, meaning that our system greatly improves the quality of the generated animation. In Figure ??, the subset of results from our system also look much closer to the ground truth compared to the results from the baseline model.

From the breakdown in Table A.1, which shows the metrics for a subset of 10 characters from the evaluation dataset, our system is most effective at animating characters with a more realistic human-like silhouette. The character "sidewalk" (Figure ??), which is an abstract model with realistic proportions, has the highest scores in all metrics. On the other hand, the characters "ramona" and "renee" (Figure ??) have the lowest scores, since they are stylized to look less like a realistic human.

The metrics for the existing pose transfer and animation models, PCDMs [29], CFLD [30], and Animate-Anyone 2 [57], are obtained from their respective papers. These metrics show that the results for our system are significantly better than PCDMs and CFLD, and approach the quality of the state-of-the-art model, Animate-Anyone 2.

In Figure 4.21, which shows the distribution of the evaluation metrics scores, most SSIM scores are in the range of 0.9 to 1.0, most PSNR scores are in the range of 20 to 29, most LPIPS scores are less than 0.1, and most FVD scores are between 50 and 150. This means that outside of a few animations in the tail of the distribution with unfavorable scores, the system can achieve performance close to the state-of-the-art Animate-Anyone 2 model when generating a majority of the animations.

4.2.4 Human-in-the-Loop Improvements

The system provides tools for the user to edit the generated animation to correct inaccuracies. Some inaccurately generated frames with low scores from the quantitative evaluation were selected to demonstrate how the manual editing tools of the system can fix inaccuracies and improve the quality of the results.

For example, by manually editing an incorrectly detected pose, the user can increase the evaluation metric scores of the generated animation as shown in Figure 4.22.

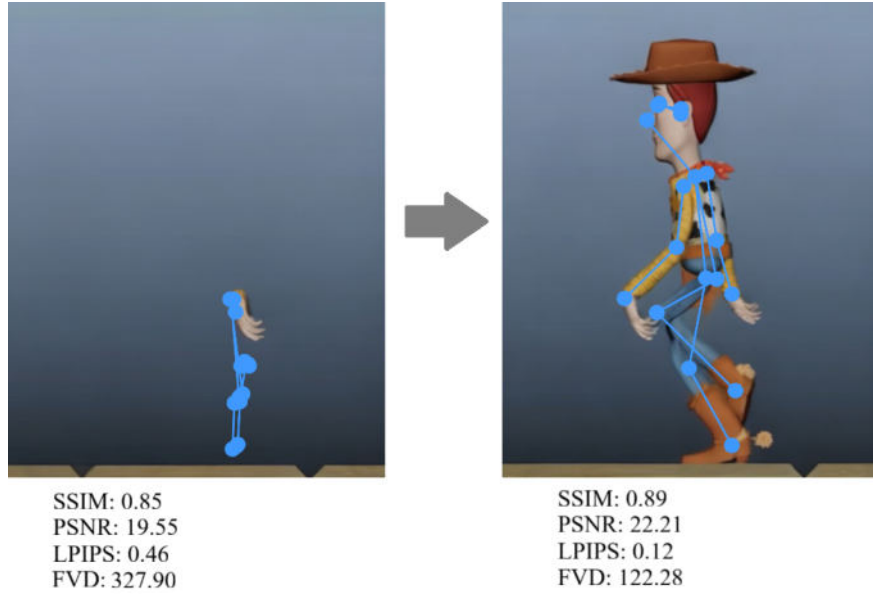


Figure 4.22. Metrics improvements after fix the pose

In Figure 4.23, the inaccurate result can be corrected by changing the character reference images to ones that look more similar to the desired output.

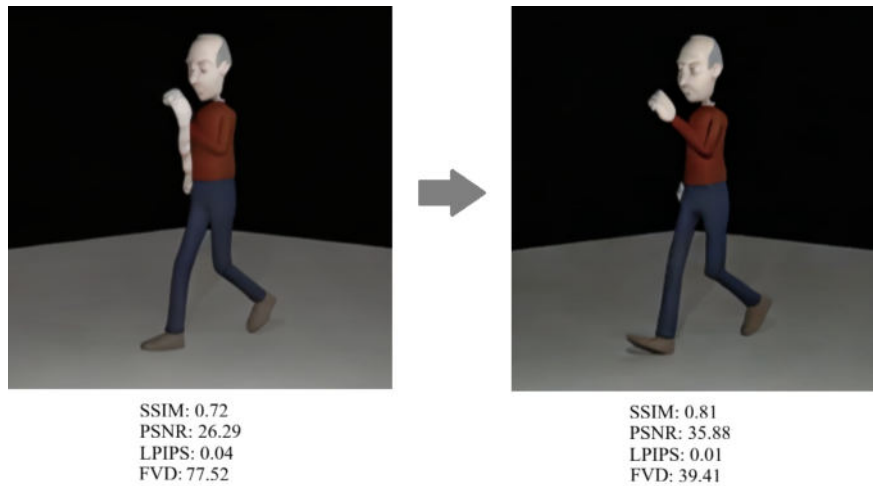


Figure 4.23. Metrics improvements after adding one more character reference image

Finally in Figure 4.24, the result could be improved by downloading frames that contain artifacts or distortions and manually correcting them in an image editor software.

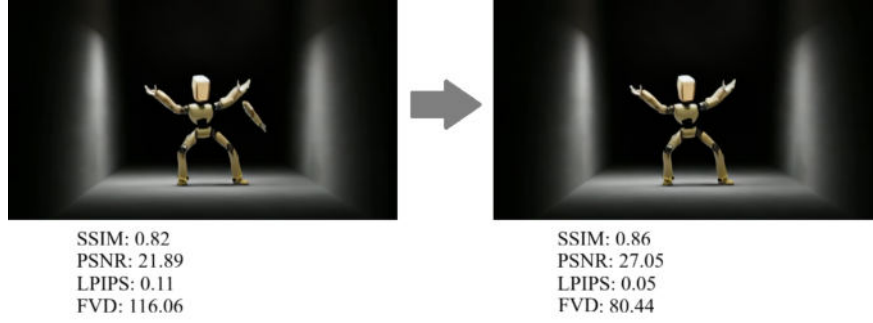


Figure 4.24. Metrics improvements after editing the image

If the user manually edits every incorrectly generated frame, it would significantly improve the evaluation metrics, bringing them closer to or even surpassing the state-of-the-art Animate-Anyone 2 model [57].

4.3 Key findings

From the quantitative and qualitative evaluations, our system can generate animations with complex motions for a wide variety of figurines across many different art styles, physical characteristics, and poses.

The main causes of failure cases are incorrectly detected poses in the pose reference frames or character reference images, additional features on the figurine that are too large, or insufficient character reference images in the required key poses. The user can remedy these failures by manually editing the incorrect poses, animating the additional features separately, and adding more character reference images.

The findings of this evaluation show the importance of having an interactive system that facilitates a human-in-the-loop workflow between the user and the AI models.

5. USE CASES

These use cases were selected to demonstrate how the system allows users to create stop-motion animations through the pose-to-pose or motion-capture approaches, rather than being limited to the straight-ahead approach. Finally, the user can use the system to edit an existing animation to modify the character’s appearance or pose.

5.1 Creating Animations with Pose-to-Pose

The pose-to-pose approach involves keyframes and inbetweening [16]. First, the animator needs to create the keyframes of the animation containing the main poses, mapping out the overall motion. Then they would interpolate between these frames to create the intermediate frames to smooth the motion.

This use case demonstrates how to use pose-to-pose techniques in our system to create a stop-motion animation of a Spiderman figurine making a snowangel.

The animator would first upload the character reference images to finetune the pose transfer model for a specific figurine. These images should be posed similarly to the keyframes of the desired animation. In Figure 5.1, a few photos of a Spiderman figurine were uploaded.



Figure 5.1. Character reference images for the snowangel pose-to-pose use case

Once the finetuning is complete, the animator can generate the keyframes of the animation by uploading the pose reference frames. These can be rough sketches, even stick figures, of the character's pose for each keyframe. In Figure 5.2, three stick figures in different poses were uploaded to generate three keyframes.

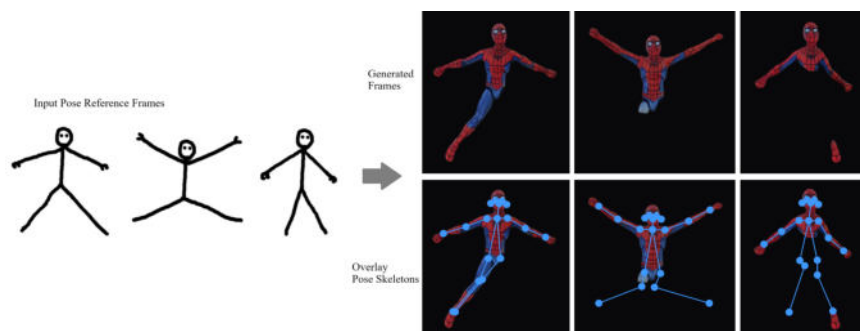


Figure 5.2. Generating keyframes from uploaded sketches

Often, the keyframes generated from rough sketches have incorrectly detected poses. The animator can manually edit the poses while comparing the generated frame with the pose reference frame, as demonstrated in Figure 5.3. When they are satisfied with the pose modifications, they can regenerate the frame using the updated pose.

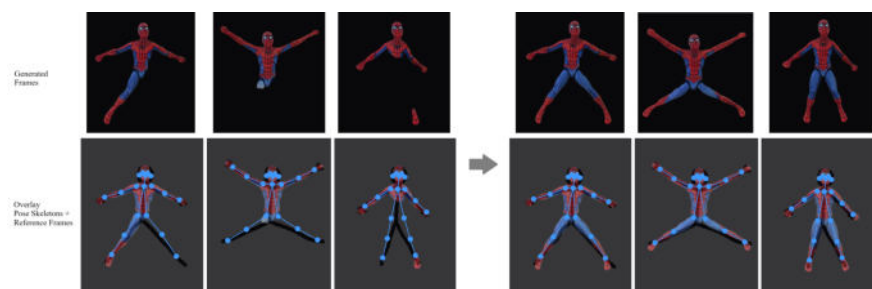


Figure 5.3. Editing incorrectly detected poses and regenerating the keyframes

There are two ways to interpolate between two keyframes. The animator can create the intermediate frames by duplicating the keyframes and manually adjusting the poses, or they can run the frame interpolation model to automatically generate the intermediate frames. The first method is suitable for complex motion that involve 3D rotations or small movements, while the second method works better when the transitions to get from one keyframe to the next are simple and straightforward. Figure 5.4 compares the two methods.

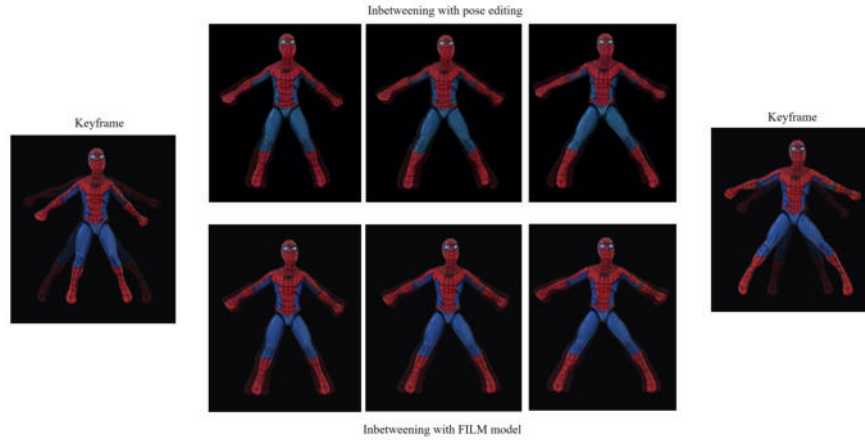


Figure 5.4. Comparing two ways to generate intermediate frames from keyframes

For the first interpolation method, the onion-skinning feature allows the animator to compare the current frame with the previous and next keyframes so they can adjust the pose to maintain smooth motion.

For the second interpolation method, the animator can specify how many intermediate frames to generate, controlling the speed at which the pose changes from one keyframe to the next.

Since the pose transfer model removes the background from the figurine, the animator can upload an image as the background and add it to the animation as a separate layer as seen in Figure 5.5.

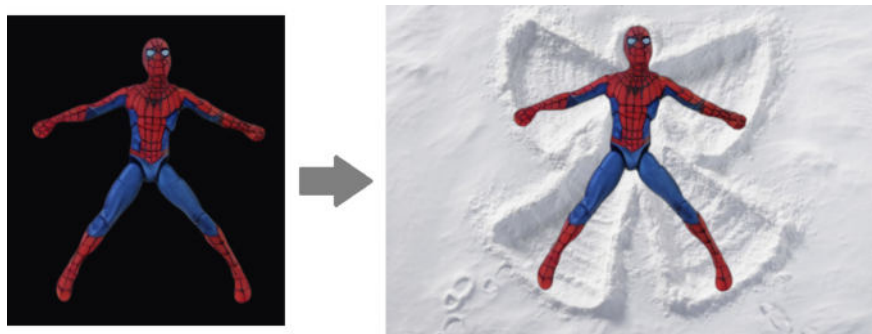


Figure 5.5. Generating keyframes from uploaded sketches

Finally, the animator can playback the animation and if they are satisfied, they can export it as a video file. The final results are shown in Figure 5.6 and Figure 5.7.

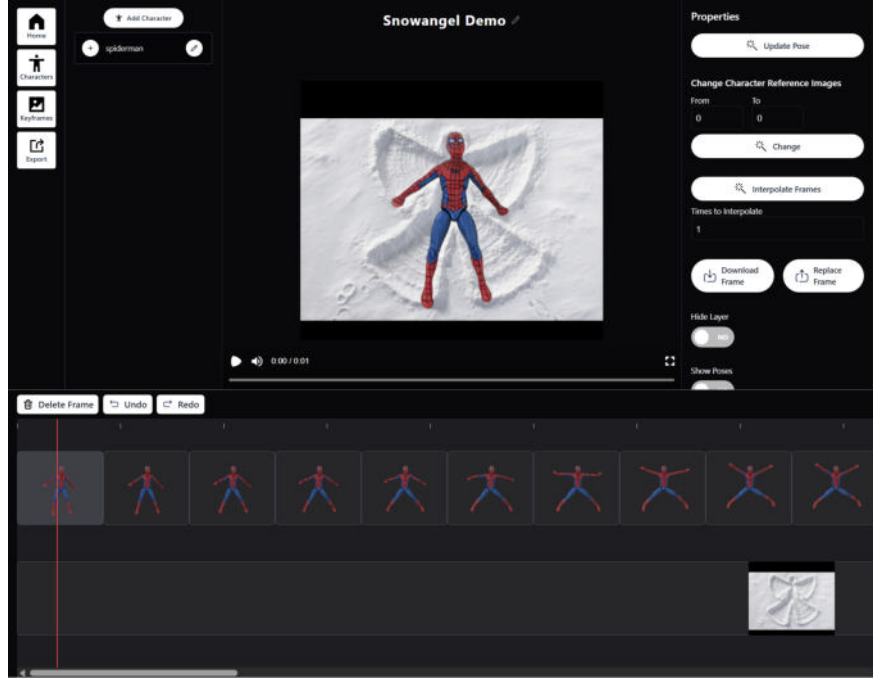


Figure 5.6. The final results of the snowangel pose-to-pose use case



Figure 5.7. Frames from the completed animation in the snowangel pose-to-pose use case

5.2 Creating Animations with Motion-Capture

To create animations using motion-capture, animators need to use a video of a real person moving and transfer the movements to the character. Traditionally the motion-capture process requires special equipment. Our system uses a pose detection model [27] to extract poses from ordinary videos, so the "motion-capture" video can simply be any video of a person. For example, to create an animation of figurines dancing, the animator could find a video of a ballet performance, or use a camera to record a video of themselves doing a simple dance.

Similar to the pose-to-pose process, the animator needs to upload character reference images for each figurine they want to animate. In this case, the images of a Spiderman figurine

and a Mary-Jane figurine, shown in Figure 5.8 and Figure 5.9, are uploaded separately to finetune a pose transfer model for each character. These images should be in similar poses as the main poses of the "motion-capture" videos.



Figure 5.8. Character reference images of the Spiderman figurine



Figure 5.9. Character reference images of the Mary-Jane figurine

For each character, the animator can upload a "motion-capture" video to use as the pose reference frames. For example, they can upload a video of a ballet performance shown in Figure 5.10 for the Spiderman character to create an animation of the Spiderman figurine dancing ballet. Similarly, they can upload a recorded video of themselves dancing, such as the one shown in Figure 5.11, to the Mary-Jane character to create an animation of the Mary-Jane figurine dancing the same way. Each character would be in its own layer. The animator can also upload a background image as a layer.

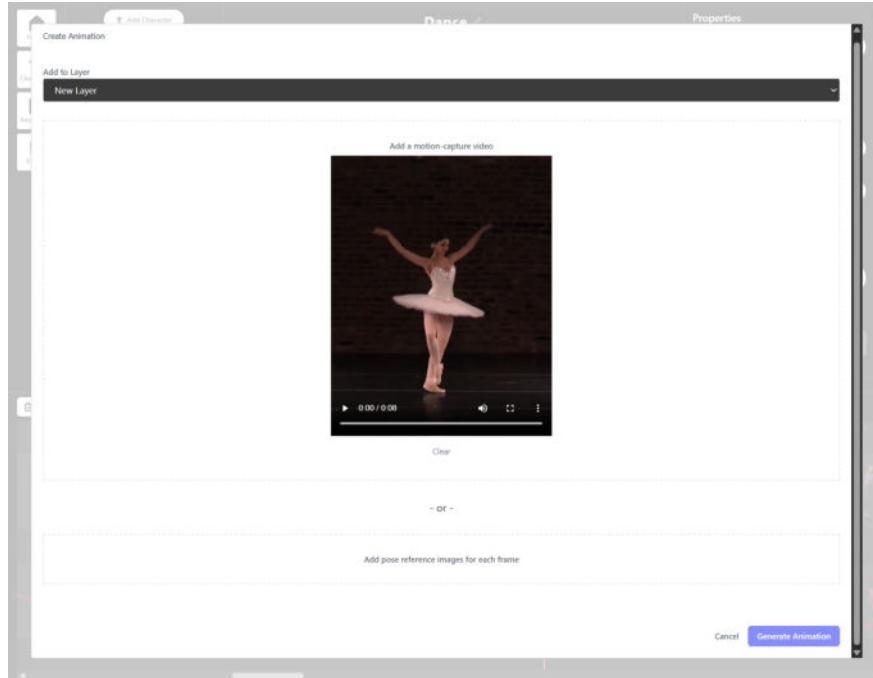


Figure 5.10. Uploading a pose reference video of a ballet performance

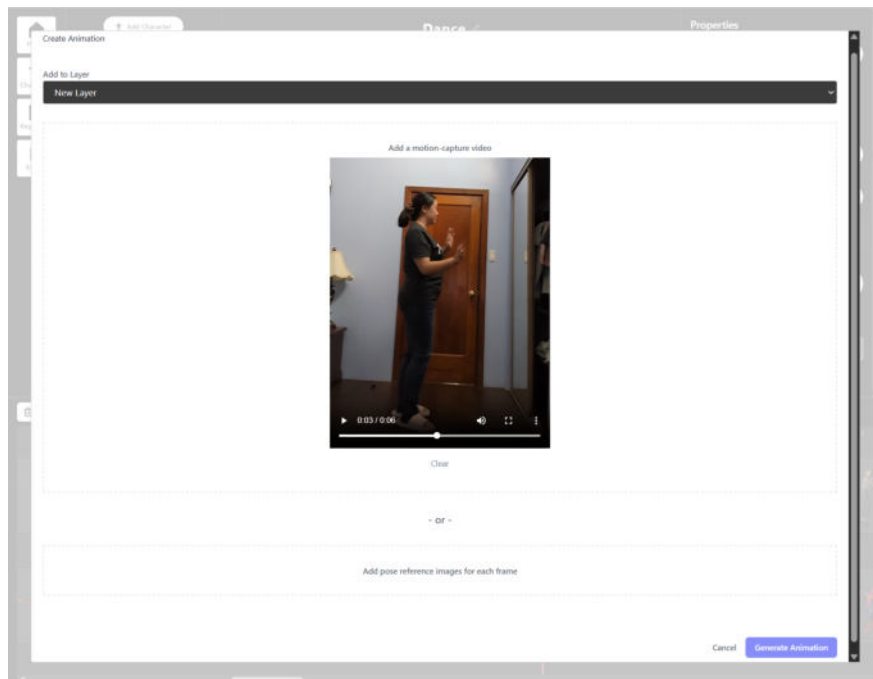


Figure 5.11. Uploading a pose reference video that the user recorded of themselves dancing

The animator can playback the animation to look for inaccurate frames. The system provides tools to make corrections.

In Figure 5.12 the figurine was facing forward instead of backwards so the animator can change the character reference images of this frame to only include an image of the figurine facing backwards. This would regenerate the specific frame to correct the inaccuracy.



Figure 5.12. Changing the character reference images to fix an incorrectly generated frame

In Figure 5.13, the generated frame contained an artifact. The animator can fix this by downloading the frame and editing it in an image editor to erase the artifact before uploading it to replace the original frame with the edited one.

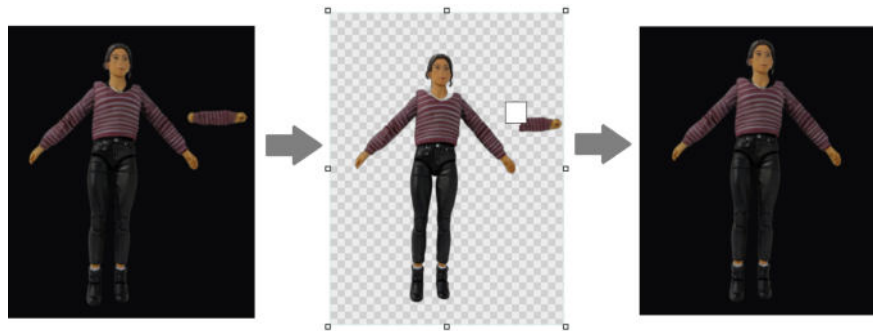


Figure 5.13. Using an image editor to erase an artifact from a generated frame and replacing the frame in the animation

In Figure 5.14, the detected pose was inaccurate. The animator can compare the generated frame with the pose reference frame and edit the pose to match the reference. After regenerating the frame with the new pose, the figurine would be posed correctly.



Figure 5.14. Editing an incorrectly detected pose to match the pose reference frame and regenerating the frame with the corrected pose

Finally, once all of the inaccurate frames have been corrected, the animator can export the animation. The final results are shown in Figure 5.15.

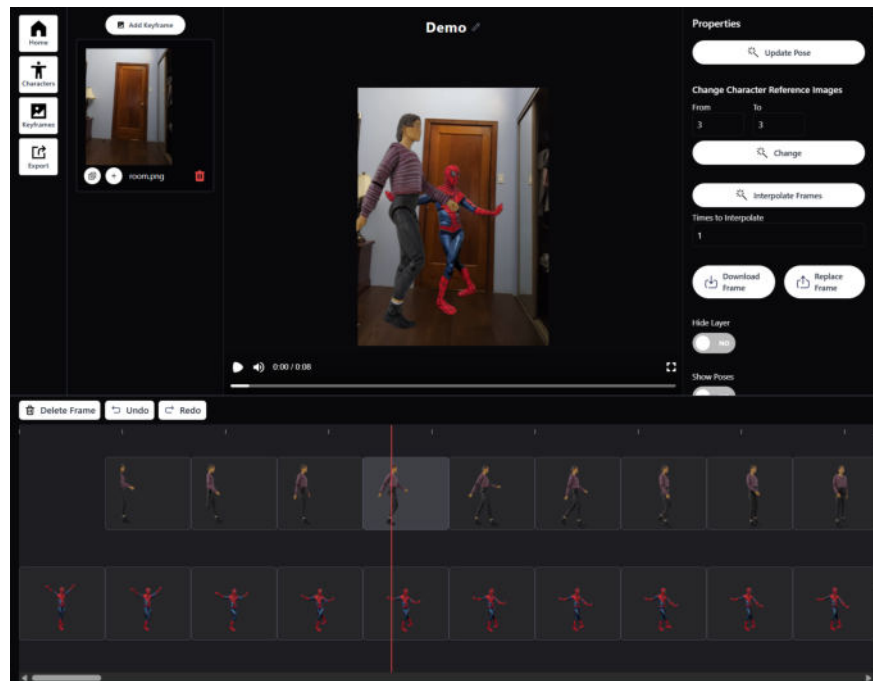


Figure 5.15. Final results of the motion-capture use case of two figurines dancing

5.3 Editing Existing Animations

The system can also be used to make changes to the pose or appearance of characters in an existing stop-motion animation. The user would not need additional images of the character since the pose transfer model would be finetuned on the frames in the existing animation.

First, they can upload the animation and add it as a layer to the timeline as shown in Figure 5.16. They can then make changes to the frames in the animation. For example, they can modify a character’s pose and replace that frame with generated frame of the new pose as seen in Figure 5.17.

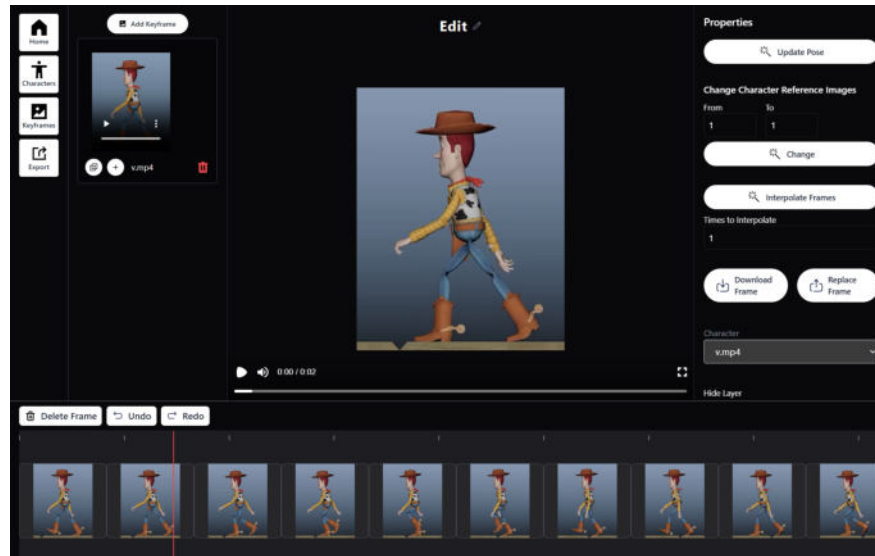


Figure 5.16. Uploading an existing animation

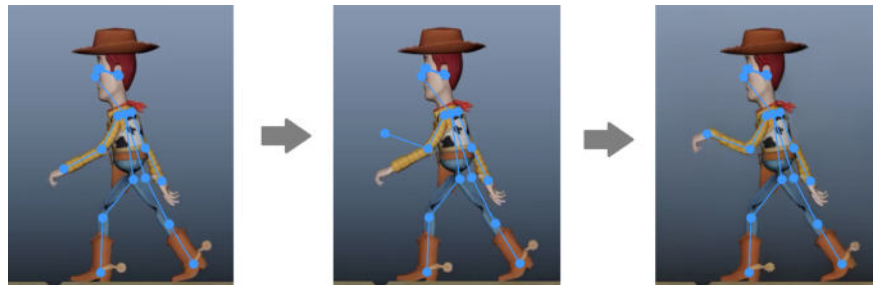


Figure 5.17. Editing a pose in an existing animation

Additionally, the user can change the character’s appearance as shown in Figure 5.18, where they changed character reference images to change the color of the character’s hair. They can edit existing animations with the same methods demonstrated in the previous use cases.



Figure 5.18. Editing the appearance of a character in an existing animation

In the past, to fix or tweak stop-motion animations, the animator would need to reshoot the entire scene [12]. By extending the system to work with existing animations, it becomes possible to edit the stop-motion animations in post-production, decreasing the repetitive workload.

6. LIMITATIONS AND FUTURE WORK

Although the system is designed to allow users to generate stop-motion animations using only character reference images and pose reference frames, there are many limitations that restrict the types of animation that could be generated. For example, currently pose transfer only works for humanoid figurines, it is difficult to animate multi-character interactions, and the system does not handle facial features and expressions.

6.1 Non-Humanoid Characters

Currently, the pose detection model [27] only detects human poses. Namely, the figurine must have a head, two arms, and two legs. This would not work for four-legged animals such as cats, dogs, and horses, or other creatures that have a wide variety of anatomical traits, such as fish, birds, or insects.

Animal pose detection models exist, such as SLEAP [62] and DeepLabCut [63], that can work for multiple types of animals, mostly quadrupeds [64], as shown in Figure 6.1.



Figure 6.1. Example output from DeepLabCut [63], an animal pose detection model (image source: <https://deeplabcut.medium.com/>)

In the future, it would be possible to train the pose transfer model on animal poses rather than human poses to allow it to handle non-humanoid characters, as long as the pose detection model for that type of animal exists.

6.2 Character Interactions

The system could currently animate scenes with multiple characters by generating animations of single characters separately on different layers. This would work if the characters were located in different parts of the scene, such as on opposite sides or one behind the other. It would be difficult to animate characters that touch or intertwine with each other, for example, locked in a fight or hugging.

Currently, to animate such a scene, the user would need to download each frame and manually edit them in an image editor software, erasing parts where the characters overlap.

The currently used pose detection model, DWPose [27], and other models, such as Lcr-net++ [65], can detect multiple poses from images that contain more than one person, as shown in Figure 6.2.

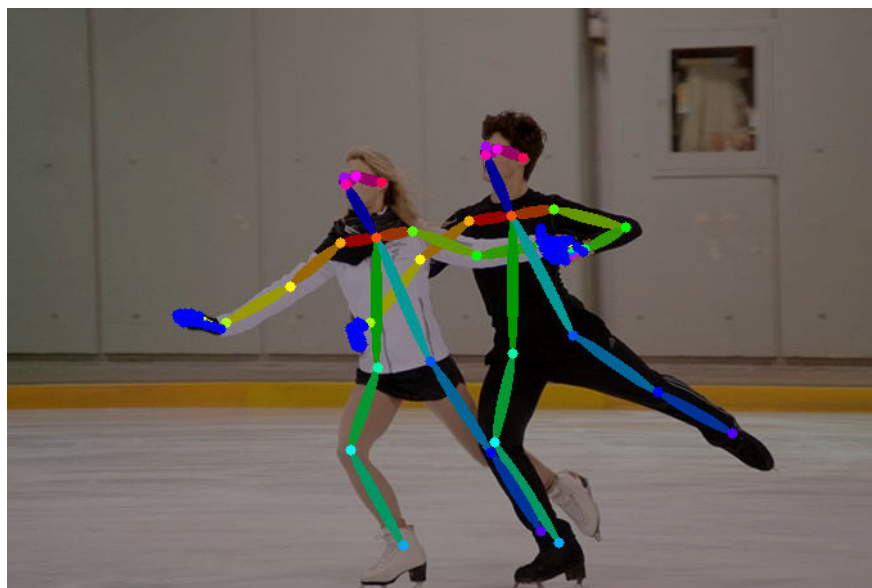


Figure 6.2. Multi-pose detection with DWPose [27] (original image source: <https://torontoobserver.ca/2013/02/14/worlds-next-stop-for-ice-dance-pair-gilles-poirier/>)

In the future, it might be possible to train the pose transfer model [29] on images with multiple poses. This should allow it to generate animations with multiple characters on the same layer, making it easier to animate scenes where characters interact.

6.3 Facial Features and Expressions

The appearance of the character can be controlled by the character reference images. However, the faces of the characters in the generated animation do not fully match the figurine, especially for figurines that look like realistic humans.

For example, in Figure 6.3, the generated animation of a figurine of Mary-Jane from Spiderman does not have the same face as the actual figurine. This is likely because the pose transfer model is pretrained on photos of real humans, so despite finetuning the model on images of the figurine, the generated output is still affected by the faces the model was pretrained on.



Figure 6.3. Comparing faces of the original character reference image and the generated frame for realistic-looking characters

In Figure 6.4, the generated animation and the actual figurine have similar faces. Unlike the previous example, the figurine of Spiderman looks much more different from the photos the model was pretrained on, so the model is able to better focus on the character reference

images separate from the pretraining data when generating the output. This shows that the facial features look more accurate if the figurine is less realistic-looking.

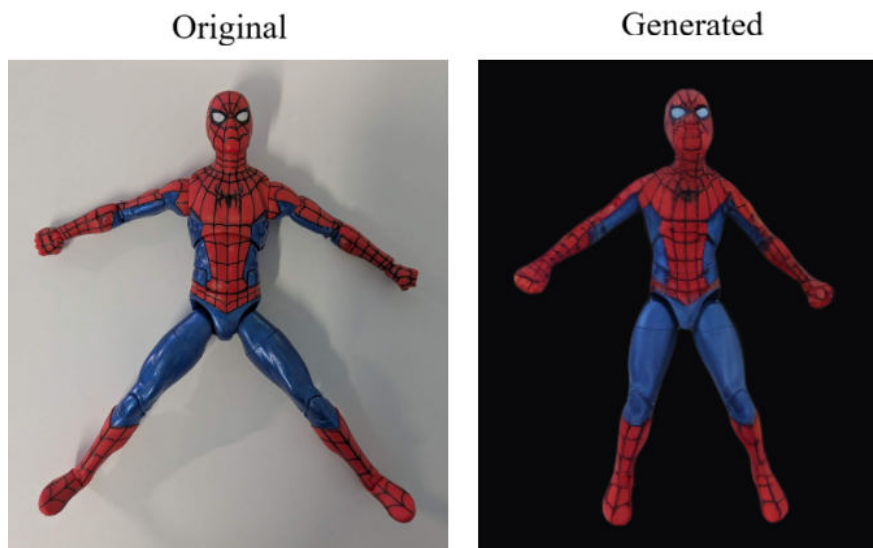


Figure 6.4. Comparing faces of the original character reference image and the generated frame for unrealistic-looking characters

To mitigate this problem, it might be possible to pretrain the pose transfer model on a different dataset. For example, instead of photos of real humans, it could be pretrained on photos of artist mannequins that do not have faces. That way, the character reference images would look more different from the pretraining data so model could prevent the faces it was pretrained on from affecting the generated output.

Additionally, the pose transfer model [29] only transfers the body poses and does not handle facial expressions. While the body would match the poses of the pose reference frames, the facial expressions would still look the same as the expressions from the training data.

With a system that could transfer facial expressions, the animator could use motion-capture techniques to control the character's expressions and frame interpolation models to transition between different expressions, whereas traditionally, animators would need to craft or 3D print many versions of their animation puppets to ensure a smooth transition. For example, "For *Coraline*, 28 different replicas of her face were made with varying expressions, even more, 42 wigs were made for the puppet" [12]. If it could be possible to control and

interpolate the puppet’s facial expressions through software, that would greatly reduce the workload.

As future work, it would be possible to use a face transfer model on top of the pose transfer model to handle the facial expressions. There are existing models for facial expression transfer [66][67][68], such as the one shown in Figure 6.5, but these are trained on photos of real people, such as the CelebFaces Attributes (CelebA) dataset [69]. It may be possible to finetune it on images of the figurines, similar to the pose transfer model [29], for it to work for characters in the animation.

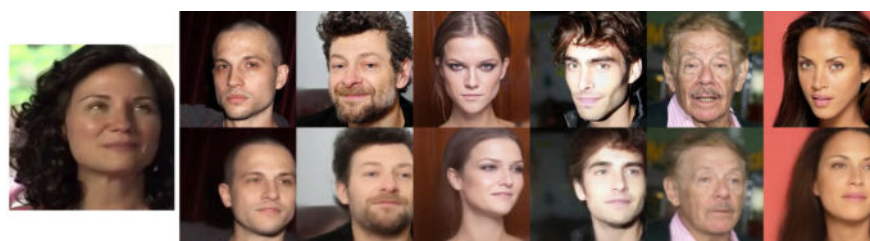


Figure 6.5. Example results from FSRT [66], a facial expression transfer model (image source: <https://github.com/andrerchow/fsrt>)

7. CONCLUSION

In conclusion, this system is a novel approach for generating stop-motion animations using a pose transfer model [29], and the results demonstrate clear improvements over existing out-of-the-box models. Unlike existing approaches that do not generalize well for figurines that do not look like realistic humans, this approach can generate high-quality animations for a wide variety of figurines.

By giving users the ability to create, edit, and control stop-motion animations through pose-to-pose or motion-capture approaches, this system broadens the techniques available for creating stop-motion animations that were previously limited to only straight-ahead techniques [13]. This system would be a useful tool for animators to edit their work and automate the more labor-intensive parts of the process, significantly reducing production barriers and making stop-motion animation accessible to a broader range of creators.

REFERENCES

- [1] N. Pettigrew, “The stop-motion filmography,” *A critical guide to 297 features using puppet animation*, vol. 1, p. 10, 1999.
- [2] G. Hoban and W. Nielsen, “Creating a narrated stop-motion animation to explain science: The affordances of “slowmation” for generating discussion,” *Teaching and Teacher Education*, vol. 42, pp. 68–78, 2014.
- [3] J. Wishart, “Exploring how creating stop-motion animations supports student teachers in learning to teach science,” *Journal of Research on Technology in Education*, vol. 49, no. 1-2, pp. 88–101, 2017.
- [4] S. Z. Maaruf, A. Mohd Nazri, K. Supramaniam, and A. Ahamed Kamal, “The design and development of stop motion animation as a pedagogical tool for teaching and learning science,” *Malaysian Journal of Sustainable Environment (MySE)*, vol. 9, no. 2, pp. 195–214, 2022.
- [5] B. L. Kamp and C. C. Deaton, “Move, stop, learn: Illustrating mitosis through stop-motion animation,” *Science Activities*, vol. 50, no. 4, pp. 146–153, 2013.
- [6] R. Zarin, K. Lindbergh, and D. Fallman, “Stop motion animation as a tool for sketching in architecture,” 2012.
- [7] V. Shtets and O. Melnyk, “Communicative potential of stop-motion animation in the practice of modern design,” 2024.
- [8] J. M. Blair, “Animated autoethnographies: Stop motion animation as a tool for self-inquiry and personal evaluation,” *Art Education*, vol. 67, no. 2, pp. 6–13, 2014.
- [9] A. Bhartiya, *Stop motion. a study on the most usefull technique of experimental animation*, 2015.
- [10] T. Gasek, *Frame-by-frame stop motion: The guide to non-puppet photographic animation techniques*. CRC Press, 2017.
- [11] B. J. Purves, *Stop-motion Animation: Frame by Frame Film-making with Puppets and Models*. A&C Black, 2014.
- [12] J. Hardy, “The fundamentals of claymation films,” 2024.
- [13] T. P. Thesen, “Reviewing and updating the 12 principles of animation,” *Animation*, vol. 15, no. 3, pp. 276–296, 2020.
- [14] K.-L. Chuang, “Dynamic frame rate: A study on viewer perception of changes in frame rate within an animated movie sequence,” in *ACM SIGGRAPH 2016 Posters*, 2016, pp. 1–1.

- [15] S. Emerson, “Visual effects at laika, a crossroads of art and technology,” in *ACM SIGGRAPH 2015 Talks*, 2015, pp. 1–1.
- [16] F. Thomas, “The illusion of life,” 1995.
- [17] J. S. Joon, “Reviewing principles and elements of animation for motion capture-based walk, run and jump,” in *2010 Seventh International Conference on Computer Graphics, Imaging and Visualization*, IEEE, 2010, pp. 55–59.
- [18] D. Apriliyanti, J. N. Fadila, F. Nugroho, *et al.*, “3d animation design" science, lanterns to heaven" using the pose-to-pose method,” *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 8, no. 1, pp. 46–55, 2022.
- [19] E. Khapugin and A. Grishanin, “Physics-based character animation with cascadeur,” in *ACM SIGGRAPH 2019 Studio*, 2019, pp. 1–2.
- [20] L. Leite, “Ai mocap for artists,”
- [21] D. Vronay and S. Wang, “Designing a compelling user interface for morphing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2004, pp. 143–149.
- [22] A. Watkins, *Getting Started in 3D with Maya: Create a Project from Start to Finish—Model, Texture, Rig, Animate, and Render in Maya*. Routledge, 2012.
- [23] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast, “Dynamic response for motion capture animation,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 697–701, 2005.
- [24] A. Blattmann *et al.*, “Align your latents: High-resolution video synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 563–22 575.
- [25] P. Esser, J. Chiu, P. Atighehchian, J. Granskog, and A. Germanidis, “Structure and content-guided video synthesis with diffusion models,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7346–7356.
- [26] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “Openpose: Realtime multi-person 2d pose estimation using part affinity fields,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019.
- [27] Z. Yang, A. Zeng, C. Yuan, and Y. Li, “Effective whole-body pose estimation with two-stages distillation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4210–4220.

- [28] M. Field, Z. Pan, D. Stirling, and F. Naghdy, “Human motion capture sensors and analysis in robotics,” *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 163–171, 2011.
- [29] F. Shen, H. Ye, J. Zhang, C. Wang, X. Han, and W. Yang, “Advancing pose-guided image synthesis with progressive conditional diffusion models,” *arXiv preprint arXiv:2310.06313*, 2023.
- [30] Y. Lu, M. Zhang, A. J. Ma, X. Xie, and J. Lai, “Coarse-to-fine latent diffusion for pose-guided person image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6420–6429.
- [31] R. Abdrashitov, A. Jacobson, and K. Singh, “A system for efficient 3d printed stop-motion face animation,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 1, pp. 1–11, 2019.
- [32] V. Maselli *et al.*, “The evolution of stop-motion animation technique through 120 years of technological innovations,” *International Journal of Literature and Arts*, vol. 6, no. 3, pp. 54–62, 2018.
- [33] A. Ciucanu, N. Bhandari, X. Wu, S. Ravikumar, Y.-L. Yang, and D. Cosker, “E-stopmotion: Digitizing stop motion for enhanced animation and games,” in *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2018, pp. 1–11.
- [34] Y. Liu *et al.*, “Sora: A review on background, technology, limitations, and opportunities of large vision models,” *arXiv preprint arXiv:2402.17177*, 2024.
- [35] A. Blattmann *et al.*, “Stable video diffusion: Scaling latent video diffusion models to large datasets,” *arXiv preprint arXiv:2311.15127*, 2023.
- [36] Y. Guo *et al.*, “Animatediff: Animate your personalized text-to-image diffusion models without specific tuning,” *arXiv preprint arXiv:2307.04725*, 2023.
- [37] R. Burgert *et al.*, “Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise,” *arXiv preprint arXiv:2501.08331*, 2025.
- [38] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless, “Film: Frame interpolation for large motion,” in *European Conference on Computer Vision*, Springer, 2022, pp. 250–266.
- [39] C. Wang and P. Golland, “Interpolating between images with diffusion models,” 2023.
- [40] H. J. Smith, Q. Zheng, Y. Li, S. Jain, and J. K. Hodgins, “A method for animating children’s drawings of the human figure,” *ACM Transactions on Graphics*, vol. 42, no. 3, pp. 1–15, 2023.

- [41] L. Khachatryan *et al.*, “Text2video-zero: Text-to-image diffusion models are zero-shot video generators,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 954–15 964.
- [42] Y. Ren, G. Li, S. Liu, and T. H. Li, “Deep spatial transformation for pose-guided person image generation and animation,” *IEEE Transactions on Image Processing*, vol. 29, pp. 8622–8635, 2020.
- [43] W.-Y. Yu, L.-M. Po, R. C. Cheung, Y. Zhao, Y. Xue, and K. Li, “Bidirectionally deformable motion modulation for video-based human pose transfer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7502–7512.
- [44] P. Zhang, L. Yang, J.-H. Lai, and X. Xie, “Exploring dual-task correlation for pose guided person image generation,” in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2022, pp. 7713–7722.
- [45] T. Wang *et al.*, “Disco: Disentangled control for realistic human dance generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9326–9336.
- [46] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, “Pose guided person image generation,” *Advances in neural information processing systems*, vol. 30, 2017.
- [47] H.-I. Ho, L. Xue, J. Song, and O. Hilliges, “Learning locally editable virtual humans,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 024–21 035.
- [48] J. Karras, A. Holynski, T.-C. Wang, and I. Kemelmacher-Shlizerman, “Dreampose: Fashion image-to-video synthesis via stable diffusion,” in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 2023, pp. 22 623–22 633.
- [49] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “First order motion model for image animation,” *Advances in neural information processing systems*, vol. 32, 2019.
- [50] A. Siarohin, O. J. Woodford, J. Ren, M. Chai, and S. Tulyakov, “Motion representations for articulated animation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 653–13 662.
- [51] J. Zhao and H. Zhang, “Thin-plate spline motion model for image animation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3657–3666.
- [52] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “Animating arbitrary objects via deep motion transfer,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2377–2386.

- [53] L. Liu *et al.*, “Neural rendering and reenactment of human actor videos,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–14, 2019.
- [54] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, “Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 500–22 510.
- [55] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, “Everybody dance now,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5933–5942.
- [56] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, “Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5904–5913.
- [57] L. Hu *et al.*, “Animate anyone 2: High-fidelity character image animation with environment affordance,” *arXiv preprint arXiv:2502.06145*, 2025.
- [58] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.
- [59] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th international conference on pattern recognition*, IEEE, 2010, pp. 2366–2369.
- [60] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [61] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, “Towards accurate generative models of video: A new metric & challenges,” *arXiv preprint arXiv:1812.01717*, 2018.
- [62] T. D. Pereira *et al.*, “Sleap: A deep learning system for multi-animal pose tracking,” *Nature methods*, vol. 19, no. 4, pp. 486–495, 2022.
- [63] J. Lauer *et al.*, “Multi-animal pose estimation, identification and tracking with deeplabcut,” *Nature Methods*, vol. 19, no. 4, pp. 496–504, 2022.
- [64] L. Jiang, C. Lee, D. Teotia, and S. Ostadabbas, “Animal pose estimation: A closer look at the state-of-the-art, existing gaps and opportunities,” *Computer Vision and Image Understanding*, vol. 222, p. 103 483, 2022.

- [65] G. Rogez, P. Weinzaepfel, and C. Schmid, “Lcr-net++: Multi-person 2d and 3d pose detection in natural images,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1146–1161, 2019.
- [66] A. Rochow, M. Schwarz, and S. Behnke, “Fsrt: Facial scene representation transformer for face reenactment from factorized appearance head-pose and facial expression features,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 7716–7726.
- [67] F. Qiao, N. Yao, Z. Jiao, Z. Li, H. Chen, and H. Wang, “Geometry-contrastive gan for facial expression transfer,” *arXiv preprint arXiv:1802.01822*, 2018.
- [68] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, “Real-time expression transfer for facial reenactment,” *ACM Trans. Graph.*, vol. 34, no. 6, pp. 183–1, 2015.
- [69] Z. Liu, P. Luo, X. Wang, and X. Tang, “Large-scale celebfaces attributes (celeba) dataset,” *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.

A. APPENDIX A

Table A.1. Results Breakdown for 10 Animations in the Evaluation Dataset









Character		SSIM	PSNR	LPIPS	FVD
sidewalk	Baseline	0.8774	21.22	0.1753	104.9
	Our system	0.9425	29.31	0.0587	42.4
aaa	Baseline	0.8579	19.08	0.2180	166.1
	Our system	0.8912	23.21	0.1013	128.4
azri	Baseline	0.7655	12.04	0.3510	290.9
	Our system	0.9059	22.26	0.0691	87.6
deadpool	Baseline	0.7146	17.07	0.2726	171.8
	Our system	0.8463	23.79	0.0799	80.1
frankgirl	Baseline	0.8952	21.02	0.1971	196.6
	Our system	0.9162	24.53	0.0827	144.4
kobold	Baseline	0.8856	17.75	0.3780	275.9
	Our system	0.9531	24.28	0.0792	126.9
ramona	Baseline	0.7955	13.75	0.4491	220.4
	Our system	0.8675	18.40	0.1043	62.6
renee	Baseline	0.7514	17.39	0.3036	253.2
	Our system	0.8149	20.47	0.1675	150.1
walk	Baseline	0.7431	16.12	0.2534	240.0
	Our system	0.8605	21.36	0.0889	83.0
woody	Baseline	0.8090	20.44	0.1825	170.2
	Our system	0.8739	24.38	0.1098	113.6

Table A.2. Comparison of results for 10 characters in the evaluation dataset

	Baseline	Our System	Ground Truth
sidewalk			
aaa			
azri			

continued on next page

Table A.2. *continued*

deadpool			
frankgirl			
kobold			
ramona			

continued on next page

Table A.2. *continued*





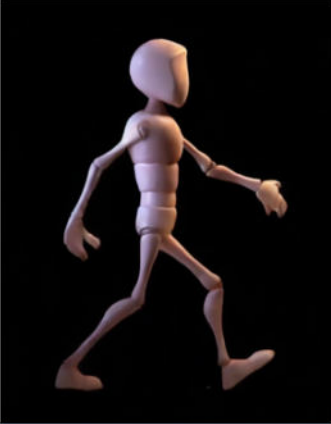




renee			
walk			
woody			

Table A.3. Sources

Character	URL
sidewalk	https://www.artstation.com/artwork/WXK6KX
aaa	https://www.artstation.com/artwork/q9eN0n
azri	https://www.artstation.com/artwork/BXnz86
deadpool	https://www.artstation.com/artwork/1xL3l8
frankgirl	https://www.artstation.com/artwork/39oVXB
kobold	https://www.artstation.com/artwork/qQ3eA2
ramona	https://www.artstation.com/artwork/Vgn80P
renee	https://www.artstation.com/artwork/wmq3O
walk	https://www.artstation.com/artwork/nELwgK
woody	https://www.artstation.com/artwork/aoybOX
camina	https://www.artstation.com/artwork/ea95lX
walkcycleport	https://www.artstation.com/artwork/VJBQ0n
spidergwen	https://www.artstation.com/artwork/Ryld3m
dancerenderport	https://www.artstation.com/artwork/elQ1qX
humans	https://www.artstation.com/artwork/39bm0v
sk	https://www.artstation.com/artwork/L3YDo5
bikerclothes	https://www.artstation.com/artwork/0n5kRY
forward	https://www.artstation.com/artwork/PXlwao
engarde	https://www.artstation.com/artwork/ykG3L8
hgervais	https://www.artstation.com/artwork/xDQX1W
sneak	https://www.artstation.com/artwork/QX69YZ
szucs	https://www.artstation.com/artwork/xYo6P4
catwalk	https://www.artstation.com/artwork/YG45wK
little	https://www.artstation.com/artwork/ZGBKYw
jealous	https://www.artstation.com/artwork/39on12
portoliodance	https://www.artstation.com/artwork/ID2kDG
breakdance	https://www.artstation.com/artwork/9Ew1Jq
dancingbaby	https://www.artstation.com/artwork/kDgy40
tpose	https://www.artstation.com/artwork/Le04Rr
max-fresh	https://www.artstation.com/artwork/2xRAkx
silva	https://www.artstation.com/artwork/RKaKGX
kleiner	https://www.artstation.com/artwork/RKO6rA
danceanim	https://www.artstation.com/artwork/Nqe81b
icecream	https://www.artstation.com/artwork/BXE8XA
blocking	https://www.artstation.com/artwork/DvX2K0
final-render	https://www.artstation.com/artwork/Ry1mRy
sophiedance	https://www.artstation.com/artwork/03ZX8Y
deadpoll	https://www.artstation.com/artwork/x3Yn32
x	https://www.artstation.com/artwork/03VPaY
artstation	https://www.artstation.com/artwork/BkaJ0z

continued on next page

Table A.3. *continued*

Character	URL
holt	https://www.artstation.com/artwork/8w6Y4R
oldman	https://www.artstation.com/artwork/yJWLaQ
nadya	https://www.artstation.com/artwork/RyJVav
dance	https://www.artstation.com/artwork/qe4grz
avatarfirenation	https://www.artstation.com/artwork/o2QYwJ
fernandes	https://www.artstation.com/artwork/g0brle
simplifiedance	https://www.artstation.com/artwork/OGNqJe