Talking Head Anime 4: Distillation for Real-Time Performance

Pramook Khungurn
pixiv Inc.
Tokyo, Japan
pong@pixiv.co.jp

Abstract

We study the problem of creating a character model that can be controlled in real time from a single image of an anime character. A solution would greatly reduce the cost of creating avatars, computer games, and other interactive applications.

Talking Head Anime 3 (THA3) is an open source project that attempts to directly address the problem [40]. It takes as input (1) an image of an anime character's upper body and (2) a 45-dimensional pose vector and outputs a new image of the same character taking the specified pose. The range of possible movements is expressive enough for personal avatars and certain types of game characters.

THA3's main limitation is its speed. It can achieve interactive frame rates (≈ 20 FPS) only if it is run on a very powerful GPU (Nvidia Titan RTX or better). Based on the insight that avatars and game characters do not need to change their appearance every so often, we propose a technique to distill the system into a small student neural network (< 2 MB) specific to a particular character. The student model can generate 512×512 animation frames in real time (≥ 30 FPS) using consumer gaming GPUs while preserving the image quality of the teacher model. For the first time, our technique makes the whole system practical for real-time applications.

1. Introduction

We are interested in animating a single image of an anime character through specifying explicit pose parameters, as if controlling a rigged 3D model. We are motivated by the recent popularity of *virtual YouTubers* (VTubers): anime characters which are animated in real time with the help of recent computer graphics technologies [52]. Typically, VTubers models are layered images (aka 2.5D models) [48] created by tools such as Live2D [49], E-mote [53], and Spine [23]. Such a model can be costly to create, so a solution to our problem would make it much easier to acquire a controllable avatar.

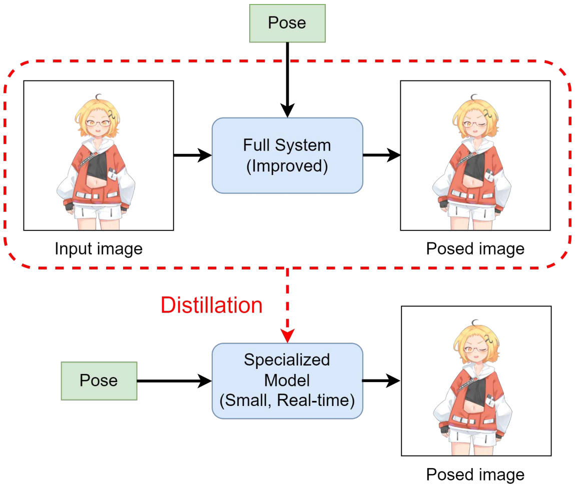

Figure 1. We present a way to distill the full THA3 system [40], which can animate an arbitrary anime character slowly, to a small model ($<$ 2MB) specialized to a single character that can generate animation in real-time.

The problem has received some attention from the research community [41, 102, 103], private companies [1, 34], and individual open-source developers [40, 87]. In particular, Khungurn proposes a neural network system called "Talking Head Anime 3" (THA3) that can generate simple animations of a humanoid anime character, given only a single image of the frontal view of the character's torso [40]. With it, a character can be controlled interactively through 45 parameters, enabling rich facial expressions and rotation of the head and the body by small angles. The afforded movements that are similar to what typical hand-made VTuber models are capable of, and they are generated with no manual modeling. Nevertheless, THA3 is not yet practical for real-time applications. Its main shortcoming is its speed: interactive frame rates can only be achieved when using very powerful GPUs, such as the Nvidia Titan RTX [40].

In this paper, we directly address the speed problem. The idea is to distill [28] the knowledge of the full system (the teacher) into a new neural network (the student) that is small

(< 2 MB) and can generate a 512×512 frame in no more than 30 ms using a consumer gaming GPU. The student, however, can only animate a specific input image. Distillation takes several ten hours, but, once finished, the student can be used as a controllable character model. This capability makes THA3 usable in real-time applications for the first time. While we cannot change character and animate it immediately, the system remains practical because a VTuber or a game character does not change its appearance so often (every second or every minute). Our solution thus trades offline preprocessing time for fast online performance in a way that suits the application at hand.

The architecture for the student network is based on the SInusoidal REpresentation Network (SIREN) [79], which we extend to make it faster and better at preserving details of the input image. In particular, we make it generate images in a multi-resolution fashion: the first few layers generate a low-resolution feature tensor, which is upscaled and passed to later layers. In this way, the network does not have to process 512×512 tensors at all layers, greatly speeding up inference. The SIREN has a pathway that generates output pixels directly, but the network has not enough capacity to generate all details present in the input image. To solve this problem, we have the SIREN also generate an appearance flow [108], which is used to warp the input image. The result is then alpha blended with the directly generated pixels. We propose a three-phase training process for the proposed model, and we verified that all of the phases are necessary to achieve better image quality.

As a minor contribution, we also address an image quality problem of THA3. When occluded parts of the character are rotated and become visible, they often appear blurry. Moreover, THA3 has a tendency to remove thin structures, such as hair stands, after head rotation. We updated the architectures of THA3's subnetworks that rotate body parts from an encoder-decoder network and a vanilla U-Net as described in the original paper [70] to a variant of U-Net with attention [88], now widely used in diffusion probabilistic models [20, 29]. The new architecture improves image quality according to three commonly used metrics, reduces blurring in disoccluded[1] areas, and preserves thin structures better. The system becomes slower due to larger and more complicated networks, but distilling it yields better student models.

2. Related Works

2.1. Implicit Neural Representation

The student network is a **neural implicit representation** (INR), a neural network used to approximate signals rather than functions that transform them. INRs often incorporate positional encoding [84] or have unconventional activation functions [72, 79] or network structures [78]. Researchers have applied INRs to signals such as images [13, 82], 3D surfaces [58, 64], and volume density coupled with radiance [59]. INRs can be used to build generative models of high-resolution 3D signals, which were previously hard to achieve [10, 11, 14, 74, 83].

While INRs can be used to directly represent articulated characters [18, 66, 100, 110], we follow Bemana *et al*. [6] and view our signal as a parameterized collection of images rather than a deformable 3D shape. As a result, our student model employs image processing tools such as warping and interpolation.

2.2. Parameter-Based Posing of a Single Image

We want to create simple animations from a single image of a humanoid character. The input is an image of a subject (the **target** image), and we need to modify it so that the subject is posed according to some specification. Based on how the pose is specified, the problem can be classified as **parameter-based posing** (explicitly by a **pose vector**), **motion transfer** (implicitly via an image or video of another subject), or **visual dubbing** (inferred from a spoken voice record). Our system solves parameter-based posing. As a result, we shall review works that solve the same problem and exclude those that take videos or multiple images as input [21, 26, 30, 31, 33, 75–77, 92, 94, 96, 98, 101, 106]. However, we will compare our system against two of these systems [26,33] to illustrate the advantages of our approach.

To our knowledge, there are three approaches to the problem at hand.

Direct modeling. We can create a controllable model of the subject's geometry from the target image. The common approach is to fit a **3D morphable model** (3DMM) [8, 9, 46, 51, 62, 65, 107] to it. While earlier works are limited in controllability and only suitable for image manipulation [7, 9, 24], recent works provide much more control [17, 25, 27, 32, 44, 47]. A drawback of this approach is that parametric models often do not cover all visible parts. For example, models specialized to the face might ignore the hair [44, 47], the neck [32], or both [25].

While there is much research on modeling from human photos, much less attention has been paid to other image domains. Saragih *et al*. construct controllable 3D models of non-human faces, but they can only animate masks [73]. Jin creates E-mode models from single anime-style images [36]. Chen *et al*. study 3D reconstruction from a single anime character's image [12] where the reconstruction can be later animated with the help of off-the-shelf components [39, 97].

[1] "Disocclude" means "to cause to be no longer occluded." As far as we know, the word does not appear in standard dictionaries but has been used in a number of computer vision papers [63, 99].

Generative modeling in the latent space. Another approach is to train a generative model that maps a **latent code** to an image, engineering it so that the output is controllable through a pose vector. At test time, we first fit a latent code to the target image.[2] Animation frames can then be generated by fixing the latent code and varying the pose vector. Tewari *et al*. train a network to alter latent codes of a StyleGAN [37, 38] according to 3DMM parameters [85] and later propose a specialized algorithm to fit latent codes to portraits [86]. Using different methods, Kowalski *et al*. [43] and Deng *et al*. [19] train GANs whose latent codes have parts that are explicitly controllable. Recent works extend EG3D [11], a 3D-aware GAN, so that the facial expression can be controlled [54, 83, 95]

Image translation. Alternatively, we can view parameter-based posing as a special case of **image translation**: transforming an image into another according to some criteria. Isola *et al*. [35] present a general framework based on conditional generative adversarial networks (cGANs) [60], which is extended in various aspects by subsequent research [15, 16, 109]. Recently, researchers have also started exploring using diffusion models for the task [45, 71, 93].

Pumarola *et al*. create a network that modifies human facial features given an *Action Units* (AUs) encoding of a facial expression [67]. Ververas and Zafeiriou do the same but use blendshape weights instead of the AUs [89]. Ren *et al*.'s PIRenderer handles not only facial expression but also head rotation [68]. Zhang *et al*.'s SadTalker [105] can control a face image through 3DMM parameters by mapping them to facial landmark positions, which are then fed to Wang *et al*.'s face-vid2vid model [90] to move the input image. Nagano *et al*. design a conditional GAN that outputs a realistic facial texture, taking as input the target image and renderings of a template mesh whose expression can be freely controlled [61].

Several works exclusively target faces of anime characters, such as those by Zhang *et al*. [102, 103]. Kim *et al*. created a dataset that can be used to train parameter-based posers, such as PIRenderer, so that they work on anime faces [41]. Unlike these works, ours deals with the whole torso.

3. Baseline and Its Improvement

THA3 as a whole is an image translator. It takes as inputs (1) a 512×512 image of the "half-body shot" of a humanoid anime character and (2) a 45-dimensional pose vector. It then outputs a new image of the same character, now posed accordingly. The 45 parameters allow a character to not only express various emotions but also move its head and body like a typical professionally-created VTuber model. Among the parameters, 39 control facial expression, and 6 control rotation of the face and the torso.

The system has 5 neural networks that can be divided into two modules. Three networks form the **face morpher**, whose duty is to alter the character's facial expression. We will not modify this module, but we will distill it into a smaller network in Section 4. The remaining two networks are called the **half-resolution rotator** and the **editor**. Together, they form a module called the **body rotator**, whose duty is to rotate the head and the torso according to the 6 non-facial-expression parameters. The half-resolution rotator operates on a half-resolution (256×256) image obtained by downscaling the output of the face morpher. Its output is then upscaled to 512×512 and then passed to the editor, which in turn generates the final output that is returned to the user.

The two networks share the same overall structure. Each contains a backbone convolutional neural network (CNN): the half-resolution rotator uses an encoder-decoder network, and the editor uses a U-Net. Each backbone network outputs a feature tensor that has the same resolution as the input image. The feature tensor can then be used to perform three image processing operations.

Warping. The feature tensor is transformed into an *appearance flow*, a map that tells, for each pixel in the output, which pixel in the input should data be copied from [108]. It is applied to the input image to get a warped version.

Direct generation. The feature tensor is transformed into pixel values directly. Because this operation is not limited by what is visible in the input image, it yields more plausible disoccluded parts but cannot preserve all the details in the visible parts.

Blending. The feature tensor is transformed into an alpha map, which can then be used to blend the results of other steps together.

Outputs of the two networks are generated using some combinations of the above operations.

3.1. Improved Architectures

When THA3 rotates the body in such a way that disoccluded parts become visible, these parts can appear blurry. Moreover, if rotated parts are thin, THA3 tends to remove them altogether. As mentioned earlier, we propose to distill THA3 to a student network. Clearly, the student would inherit the teacher's behavior including all of its image quality problems. It is thus advisable to improve THA3 before distilling it to get better student models. We do so by modifying the networks in the body rotator module without significantly changing their functions.

We changed all backbone networks to U-Nets with attention. The architecture is now widely used in diffusion

[2] Optionally, the generative model can be fine-tuned to match the input image better [69].

models [20, 29] and proves to be excellent at image generation. We slightly altered the interfaces of the networks. There is no change to the half-resolution's interface: it still outputs the posed imaged and the appearance flow used to generate the former. On the other hand, the editor take both outputs, scaled up to 512 × 512, instead of just the appearance flow like the THA3's editor. We also changed how the networks internally handle its inputs and outputs. Details of these changes can be found in the supplementary material.

To trained the changed networks, we used datasets created from approximately 8,000 controllable 3D anime character models we individually collected from the Internet. Each example in the datasets contains three items: (1) an image of a character in a "rest" post, (2) a pose vector, and (3) another image of the same character after being posed according to the pose vector. The training dataset contains 500,000 examples, while the test dataset contains 10,000. The two datasets do not share 3D models, ensuring clean separation between training and test data. Please refer to the write-up of the THA3 project for how to prepare the datasets [40]. Other details on the training process, such as loss functions, optimizers, and batch sizes, can be found in the supplementary material.

4. Distillation

As we shall see in Section 5, THA3 cannot achieve interactive frame rates without a powerful GPU. Moreover, the improved networks in Section 3.1 trade inference speed for image quality. Our task is thus to improve image generation speed so that real-time performance is achieved on less powerful hardware.

We observe that the system is overly capable. At any time, we can change the input image, and the change would be reflected on the output immediately. Nevertheless, in computer games and streaming, a character does not change its appearance every second or every minute. By creating a model that is specialized to a particular input image, we may obtain a faster model. To do so, we rely on **knowledge distillation**, the practice of training a smaller model (the student) to mimic the behavior of a larger model (the teacher = the full system) [28].

Our student models are coordinate-based networks [84]. By construction, they allow generating any specific subimage at a cost proportional to the subimage's size. Moreover, unlike CNN-based image generators, subimage generation can be done without having to generate the whole image. This feature is beneficial for game characters and real-time streaming because, in some cases, the user might want to depict only the head instead of the whole torso. We use the SInusoidal REpresentation Network (SIREN) [79] architecture because we found that it produced smooth images that fit well with the anime style. On the other hand, a competing approach [84] tends to produce grainy artifacts [79].

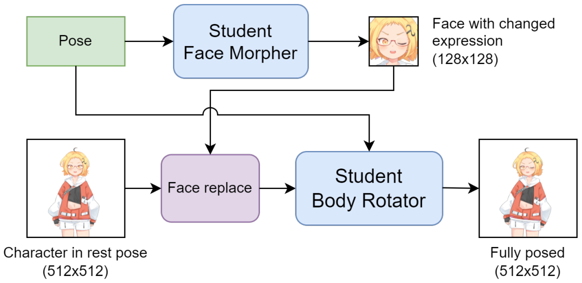

Figure 2. Overall architecture of the student model.

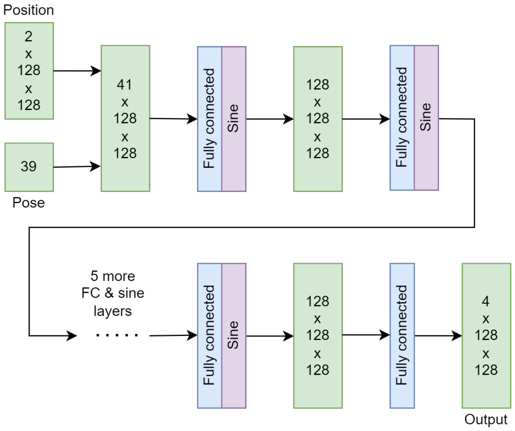

Figure 3. Architecture of the student face morpher.

4.1. Student Architecture

The student model contains two modules, **the face morpher** and the **body rotator**, with the same functionalities as those of the full system. An overview of the student's architecture is shown in Figure 2.

The student face morpher (Figure 3) is a SIREN with 9 fully connected layers, and each hidden layer has 128 neurons. It is trained to generate a 128 × 128 area of the input image that contains movable facial organs (eyebrows, eyes, mouth, and jaw). It receives a pixel position (2 dimensions) and a facial pose (39 dimensions), and it produces an RGBA pixel (4 dimensions). Its size is only 475 KB.

The student body rotator (Figure 4) needs to generate 512 × 512 images in real time. A vanilla SIREN would be too slow because it has to operate on tensors of that size at all of its layers. To improve speed, we introduce three substeps where the network would operate on tensors with spatial resolution of 128 × 128 first, then 256 × 256, and lastly 512 × 512. Each substep has 3 fully connected layers, except for the last one which has 4, resulting in a network with 10 such layers. After this, the network uses the image formation process employed by the teacher's body rotator to generate the final output. In particular, it is trained to generate (1) an appearance flow, (2) an RGBA image, and

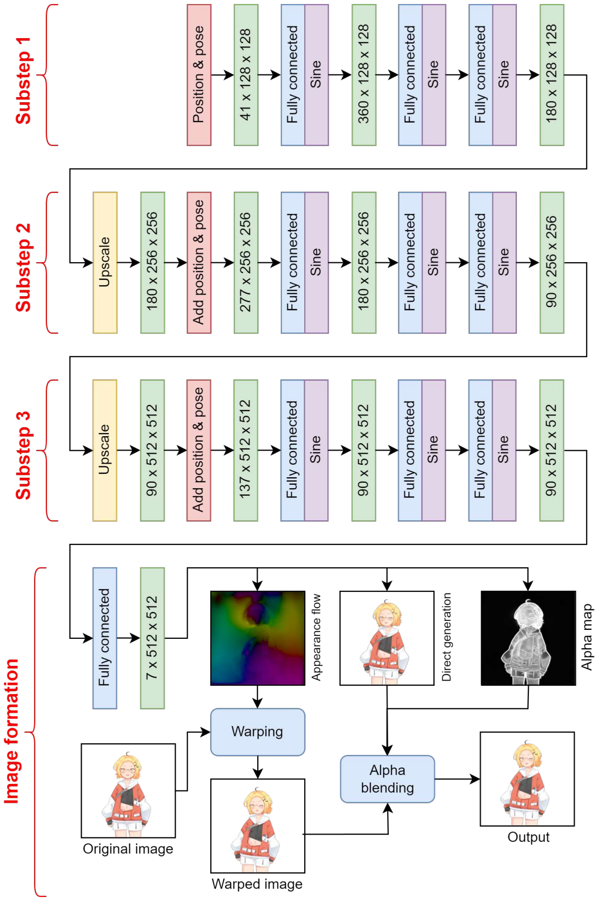

Figure 4. Architecture of the student body rotator.

(3) an alpha map. The output image is formed by first using the appearance flow to warp the input character image and then alpha blending the warped image with the directly generated RGBA image. The model size is about 1.3 MB.

4.2. Student Training

Face morpher. The student face morpher is trained to minimize the L1 differences between its outputs and those generated by the teacher face morpher. The loss function has two terms. The first is the L1 difference between the whole outputs, and the second is the L1 difference between areas that contain movable facial parts. We weigh the second term 20 times more than the first because the movable parts are small compared to the whole face. The precise definition of the loss is given in the supplementary material.

At training time, the character image is fixed, and the pose vectors are sampled uniformly from the training dataset of the teacher. Training lasts 2 epochs (1M examples) with batch size of 8. We use the Adam optimizer [42] with $(\beta_1, \beta_2) = (0.9, 0.999)$. The learning rate starts at 10^{-4} and decays to 3.33×10^{-5}, 10^{-5} and then 3.33×10^{-6} after 200K, 500K, and 800K training examples, respectively. Training takes about an hour and a half on a computer with four V100 GPUs.

Phase	# Examples	λ_{flow}	λ_{warp}	λ_{dir}	λ_{fin}
#1	$\leq$ 400K	0.50	0.25	2.00	0.25
#2	$\leq$ 800K	5.00	2.50	1.00	1.00
#3	$\leq$ 1.5M	1.00	1.00	1.00	10.00

Table 1. Training phases of the student body rotator.

Body rotator. Recall that the body rotator uses the same image formation process as the teacher body rotator. The outputs of the last fully-connected layer are (1) an appearance flow I_{flow}, (2) an RGBA image I_{dir}, and (3) an alpha map I_α. The flow I_{flow} is used to generate a warped image I_{warp} from the input character image. Then, I_{warp} and I_{dir} are alpha blended according to I_α to generate the final output image I_{fin}. The teacher also generates these data as well. We distinguish between those generated by the student with the superscript "S" (e.g., I^S_{flow}, I^S_{warp}) and those generated by the teacher with the superscript "T" (e.g., I^T_{dir}, I^T_{fin}).

The student body rotator is trained to minimize a loss, each of whose term involves one of the data above:

$$\mathcal{L}_{\mathrm{br}} = \lambda_{\mathrm{flow}}\mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{warp}}\mathcal{L}_{\mathrm{warp}} + \lambda_{\mathrm{dir}}\mathcal{L}_{\mathrm{dir}} + \lambda_{\mathrm{fin}}\mathcal{L}_{\mathrm{fin}},$$

where $\mathcal{L}_\square = \|I^S_\square - I^T_\square\|_1$, and $\square$ can be replaced with the suffixes in the above equation. The λs are weights that change throughout the training process, which is divided into three phases as shown in Table 1. We can see that the the first phase focuses on training the direct generation, the second the warping, and the third the final output.

For training, we sample pose vectors from the training dataset of the full system, use the Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, and set the batch size to 8. Training now lasts for 3 epochs (1.5M examples), which is about 10 hours on a computer with four V100 GPUs. Learning rate starts at 10^{-4} and decays to 3×10^{-5}, 10^{-5}, and 3×10^{-6} after we have shown 200K, 600K, and 1.3M training examples, respectively.

5. Results

5.1. Performance of Improved Baseline

In Section 3.1, we change THA3's body rotator. We show in this subsection the results of the change.

Image quality. We compare the new body rotator against the THA3 baseline. We quantitatively evaluate the networks by comparing the images they generate against the ground-truth images in the test dataset, using three metrics for image similarity: (a) peak signal-to-noise ratio (PSNR), (b) structural similarity (SSIM) [91], and (c) Learned Perceptual Image Patch Similarity (LPIPS) [104]. Table 2

Network	PSNR (↑)	SSIM (↑)	LPIPS (↓)
THA3	22.369330	0.909369	0.048016
Section 3.1	22.962184	0.919532	0.033566

Table 2. Quantitative comparison of body rotator models' performance.

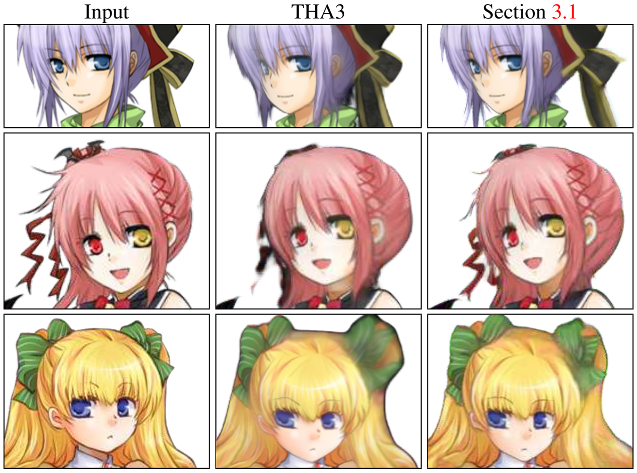

Figure 5. Qualitative comparison between images generated by body rotator models. The artworks were created by Mikatsuki Arpeggio [2–5].

shows the averages of the metrics over the 10,000 examples of the test dataset. The new architecture improves all the metrics. The LPIPS, in particular, sees an improvement of approximately 30% over THA3.

For qualitative comparison, we applied the networks to three hand-drawn characters, and we show the results in Figure 5. The characters' faces and bodies are rotated to the left of the viewer with the largest possible angles. For the 1st and 2nd characters, the THA3 rotator cannot produce sharp left silhouettes, and the ribbons worn by the 2nd character are close to being completely erased. On the other hand, the new architecture generates much sharper silhouettes and preserve the ribbons better. For the 3rd character, the THA3 rotator generates blurry hair and ribbons on the right side, while ours generates sharper results.

Model size and speed. Table 3 compares the size and speed of THA3 and our proposal. The new editor network is 4 times larger than the THA3 one, but it does not significantly increase the size of the whole system because there are four other networks that are already as large as it is. We assessed the system's speed by measuring the time it takes to fully process one input image and one pose. We performed experiments on three different computers. **Computer A** has two Nvidia RTX A6000 GPUs and represents a computer used for machine learning research. **Computer B** has an Nvidia Titan RTX GPU and represents a high-end gaming PC. **Computer C** has an Nvidia GeForce GTX 1080 Ti GPU and represents a typical, yet outdated gaming PC. Other details of the computers, such as their CPUs and memory, can be found in the supplementary material. We can see that our proposed architecture, while yielding higher image quality, are about 3 to 4 times slower than THA3. However, because of better image quality, it serves as a better teacher.

System	Size (MB)		
	HRR*	Editor	Total**
THA3	128	33	517
Section 3.1	136	137	627

System	Time needed to generate a frame (ms)		
	Computer A (RTX A6000)	Computer B (Titan RTX)	Computer C (GTX 1080 Ti)
THA3	35.899	41.409	64.607
Section 3.1	125.843	116.763	159.647

Table 3. Size and speed comparison between the THA3 system and our proposed one. The times needed to generate a frame are averages of 1,000 measurements. (*) HRR stands for "half-resolution rotator." (**) Complete THA systems have three other networks. This column contains the sizes of all the networks combined.

Character	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Top	36.156	0.9914	0.0061
Middle	36.048	0.9883	0.0066
Bottom	34.543	0.9863	0.0087

Table 4. Average PSNR, SSIM, and LPIPS between images generated by student models trained to animate the characters from Figure 5 and those generated by the teacher models.

5.2. Performance of Student Models

We will evaluate multiple student modes in this section. For each such a model, we use it to pose characters according to 1,000 fixed poses taken from the test dataset in Section 3.1. For each posed image, we compute the PSNR, SSIM, and LPIPS with respect to the corresponding image generated by the teacher model (also described in Section 3.1). We record the average of the 1,000 metric values.

Comparison against the teacher. We trained a student models for each of the three characters in Figure 5, and we report the models' metrics in Table 4. The SSIMs and LPIPSs are close to their best possible values (1 and 0, respectively), and the PSNRs range from 34 dB to around 36 dB, meaning that the average error is about 2% of the maximum pixel value. It is hard to spot large differences between the outputs, but a student model might ignore extremely fine details, such as the black dot that represents the nose, seen in Figure 6.

A student model is around 8 times faster than the teacher

System	Time needed to generate a frame (ms)		
	Comp A (RTX A6000)	Comp B (Titan RTX)	Comp C (GTX 1080 Ti)
THA3	35.899	41.409	64.607
Section 3.1	125.840	116.760	159.640
Student model	12.523	15.098	22.091

Table 5. Comparison between average time required to generate a frame of animation by the THA3 system, the teacher model (Section 3.1), and the student model.

Architecture	PSNR	Time per a frame (ms)		
		Computer A	Computer B	Computer C
Vanilla SIREN	38.259	21.319	33.086	54.937
Section 4.1 w/o multi-res	38.923	24.337	34.883	57.394
Section 4.1	38.881	12.523	15.098	22.091

Table 6. An ablation study on the architecture of the student model.

Model	Training phases			PSNR (↑)	SSIM (↑)	LPIPS (↓)
	#1	#2	#3			
A			✓	29.308	0.9739	0.0164
B		✓		29.118	0.9729	0.0167
C		✓	✓	29.484	0.9746	0.0160
D	✓			38.026	0.9946	0.0054
E	✓		✓	38.668	0.9954	0.0036
F	✓	✓		37.399	0.9943	0.0060
G	✓	✓	✓	38.881	0.9956	0.0033

Table 7. Quantitative comparison between student models trained with and without specific training phases.

model and 3 times faster than THA3, as can be seen in Table 5. It can now achieve real-time animation (≥ 30 FPS) on Computer C, which has an outdated gaming GPU.

Ablation study on student network architecture. We compare our architecture against (a) a vanilla SIREN that generates the output image directly and (b) our proposed architecture where the body rotator is modified so that it always operates at the 512 × 512 resolution. We trained the three architectures to animate a specific character image [81], and we evaluated them with the average PSNR metric. We also measured the average time it took to generate a frame. Results are available in Table 6 and Figure 6.

In Table 6, the PSNR values are close to one another. However, Figure 6 reveals that the vanilla SIREN architecture is qualitatively much worse than the other two because it cannot reproduce fine face details, such as the eyebrows, the mouth shapes, and the highlights on the pupils. This shows that the more complicated image formation steps are required to preserve them. The architecture without multi-resolution SIREN is slightly more accurate than our architecture, but it is hard to identify differences between their images in Figure 6. Hence, multi-resolution SIREN retains much of the accuracy of full-resolution SIREN while being about two times faster.

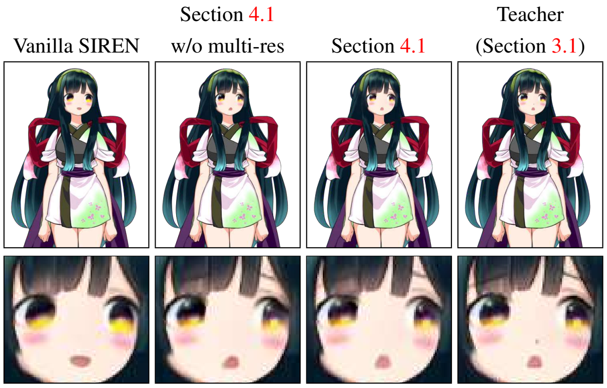

Figure 6. Qualitative comparison between images generated by the teacher and three student architectures. The character is © Touhoku Zunko · Zundamon Project [80].

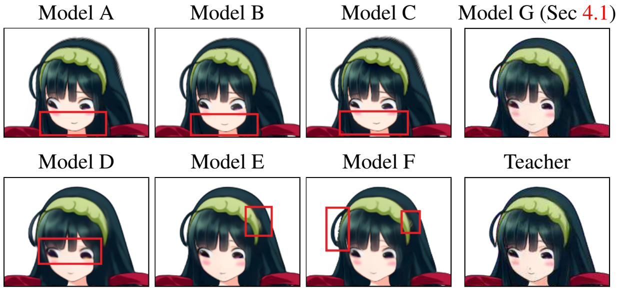

Figure 7. Qualitative comparison between outputs of models in Table 7. Problematic areas are highlight with red rectangles.

Ablation study on student training process. The training process has 3 training phases. To show their necessity, we trained student models on the character image in the last study, ablating the training phases while keeping the rest of the settings the same. We report the metrics in Table 7. Employing all phases yields the best scores. Omitting Phase #1 results in significantly worse image quality. This manifests qualitatively as noticeable differences in the shape of the rotated faces in Figure 7. Models that were trained with Phase #1, on the other hand, approximate the teacher's outputs well, achieving PSNR scores of around 38. When one or more of the other phases is missing, there are visible degradations. Model D does not reproduce the highlights on the pupils. Model E and Model F have artifacts around the headband. Moreover, Model F also yields jagged edges on one side of the head. Model G, which experienced all training phases, achieves the best scores and produces the least amount of artifacts.

5.3. Comparison Against Other Systems

We compare our teacher and student models to THA3 and two SOTA systems for image animation: AnimateAny-

Method	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Full image (512 × 512)			
THA3	25.344	0.9094	0.0422
Teacher model (Section 3.1)	25.650	0.9143	0.0294
Student model	25.754	0.9154	0.0330
AnimateAnyone	19.935	0.7955	0.1441
Face crop (192 × 192)			
THA3	19.990	0.7133	0.1061
Teacher model (Section 3.1)	20.488	0.7361	0.0741
Student model	20.698	0.7432	0.0835
AnimateAnyone	15.672	0.5096	0.1837
LivePortrait (actor video driven)	15.074	0.4796	0.2552

Table 8. Quantitative comparison of our systems against the THA3 baseline and other SOTA systems.

one [33] and LivePortrait [26]. We use an unofficial open-source implementation [50] for the former[3] and the official one for the latter.

For quantitative comparison, we recorded an actor singing a song, resulting in a video that lasts 25 seconds and contains 759 frames. We converted the actor's movement to a sequence of pose vectors with an off-the-shelf motion capture software [22]. We then use the pose vectors to animate three 3D models [55–57] and used the resulting video frames as ground truths. We also rendered the models in rest positions and used the renderings as target images. We drove THA3, the teacher model (Section 3.1), and the student models with the extracted pose vectors. We annotated the 3D models with face keypoints, rendered videos depicting them and the models' skeletons, and used the results to drive AnimateAnyone. Lastly, we drove LivePortrait with the actor video.[4]

In Table 8, we report the PSNR, SSIM, and LPIPS metrics, computed against the ground-truth videos and averaged over the three 3D models. For each metric, there are two numbers. One was computed using the full 512 × 512 images, and the other using the 192 × 192 crop around the face. Because LivePortrait can only animate faces, it does not have numbers computed with full images. We can see that the teacher model (Section 3.1) has the best LPIPS scores, and we were surprised that the student models achieved slightly better PSNR and SSIM even though they were trained to mimic the teacher. The THA systems (including THA3) achieve much better scores than other baselines, showing the advantages of systems specialized to anime characters.

For qualitative comparison, we used the systems to animate three hand-drawn characters. The animations are available in the supplementary material, and we show frames of a character in Figure 8. We can clearly see how our systems and THA3 perform much better than other systems. AnimateAnyone can neither properly rotate the head nor move the eyes and the mouth, and LivePortrait's outputs break down once the actor starts moving his head more vigorously.

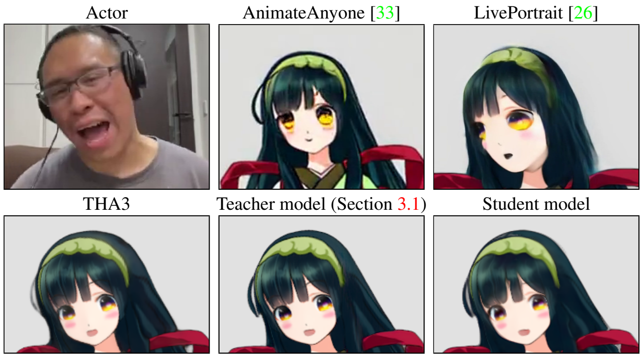
Actor
AnimateAnyone [33]
LivePortrait [26]
THA3
Teacher model (Section 3.1)
Student model

Figure 8. Qualitative comparison of our systems against THA3 baseline and other SOTA systems.

5.4. Miscellaneous Results

Student models are so fast and lightweight that they can be executed inside a web browser and still generate animations in real time. In the supplementary material, we include two demo web applications. One can pose characters through UI widgets. The other makes characters imitate the user's movement as captured by a web camera.

6. Conclusion

We improved the THA3 system, speeding it up so that it can generate animation in real time on a common gaming GPU. The improvement makes the system practical as a streaming tool for the first time. The main insight is that we can use a more expensive architecture (U-Net with attention) to get better image quality and then distill the improved model to small and fast students. Our technical contribution includes an effective architecture for the student model and an algorithm to train it.

There are still several limitations to our work. While the student model can run in real-time on a computer with a dedicated gaming GPU, it still cannot do the same on devices such as tablets or mobile phones. We also think that image quality can be improved further, and we would like to expand the possible movements the system can generate. We hope to address these problems in future work.

We also recognize that our system may be used to generate harmful and deceptive contents. In particular, it can be used to impersonate existing VTubers. Further research, such as a watermarking system, is needed to distinguish the outputs of our system from other modes of animation generation.

[3] There has been no official code/model release at the time of writing.

[4] We also tried driving LivePortrait with the rendered ground-truth videos, but the results were worse because LivePortrait could not pick up movements of facial organs from faces of 3D models.

References

[1] Algoage Inc. DeepAnime: Automatically turning illustration to anime. https://lp.deepanime.com/, 2022. Accessed: 2023-08-07. 1

[2] Mikatsuki Arpeggio. Mikatsuki Arpeggio. http://roughsketch.en-grey.com/, 2023. Accessed: 2023-09-14. 6

[3] Mikatsuki Arpeggio. Mikatsuki Arpeggio, Koakuma Mei. http://roughsketch.en-grey.com/Entry/83/, 2023. Accessed: 2023-09-14. 6

[4] Mikatsuki Arpeggio. Mikatsuki Arpeggio, Marietta. http://roughsketch.en-grey.com/Entry/67/, 2023. Accessed: 2023-09-14. 6

[5] Mikatsuki Arpeggio. Mikatsuki Arpeggio, Taoist Boy. http://roughsketch.en-grey.com/Entry/110/, 2023. Accessed: 2023-09-14. 6

[6] Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. X-fields: Implicit neural view-, light- and time-image interpolation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2020)*, 39(6), 2020. 2

[7] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging Faces in Images. *Computer Graphics Forum*, 2004. 2

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2

[9] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014. 2

[10] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*, 2021. 2

[11] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2, 3

[12] Shuhong Chen, Kevin Zhang, Yichun Shi, Heng Wang, Yiheng Zhu, Guoxian Song, Sizhe An, Janus Kristjansson, Xiao Yang, and Matthias Zwicker. Panic-3d: Stylized single-view 3d reconstruction from portraits of anime characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[13] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 2

[14] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[15] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[16] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[17] Enric Corona, Mihai Zanfir, Thiemo Alldieck, Eduard Gabriel Bazavan, Andrei Zanfir, and Cristian Sminchisescu. Structured 3d features for reconstructing relightable and animatable avatars. In *CVPR*, 2023. 2

[18] Boyang Deng, J. P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII*, page 612–628, Berlin, Heidelberg, 2020. Springer-Verlag. 2

[19] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning, 2020. 3

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 2, 4

[21] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars, 2023. 2

[22] Yasushi Emoto. iFacialMocap. https://www.ifacialmocap.com/, 2024. Accessed: 2024-09-08. 8

[23] Esoteric Software. Spine: 2d animation for games. http://esotericsoftware.com/, 2023. Accessed: 2023-08-07. 1

[24] Ohad Fried, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM Trans. Graph.*, 35(4), July 2016. 2

[25] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3d guided fine-grained face manipulation. In *CVPR*, 2019. 2

[26] Jianzhu Guo, Dingyun Zhang, Xiaoqiang Liu, Zhizhou Zhong, Yuan Zhang, Pengfei Wan, and Di Zhang. Liveportrait: Efficient portrait animation with stitching and retargeting control. *arXiv preprint arXiv:2407.03168*, 2024. 2, 8

[27] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *ICCV*, 2021. 2

[28] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 1, 4

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. 2, 4

[30] Fa-Ting Hong and Dan Xu. Implicit identity representation conditioned memory compensation network for talking head video generation. In *ICCV*, 2023. 2

[31] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[33] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 2023. 2, 8

[34] IRIAM Inc. Character streaming service "iriam," using technology contributed by preferred networks, becomes the first in the world to implement automatic character modeling by ai in smart phones. an illustration can move with rich expression through the power of ai. https://prtimes.jp/main/html/rd/p/000000006.000070082.html, 2021. Accessed: 2023-08-07. 1

[35] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[36] Yanghua Jin. Crypko - a new workflow for anime character creation. https://codh.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=400&item_no=1&page_id=30&block_id=41, 2020. 2

[37] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 3

[38] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2019. 3

[39] Pramook Khungurn. Talking head anime from a single image 2: More expressive. https://web.archive.org/web/20220327163627/https://pkhungurn.github.io/talking-head-anime-2/, 2021. Accessed: 2023-08-04. 2

[40] Pramook Khungurn. Talking head(?) anime from a single image 3: Now the body too. https://web.archive.org/web/20220606125417/https://pkhungurn.github.io/talking-head-anime-3/, 2022. Accessed: 2023-08-04. 1, 4

[41] Kangyeol Kim, Sunghyun Park, Jaeseong Lee, Sunghyo Chung, Junsoo Lee, and Jaegul Choo. Animeceleb: Large-scale animation celebheads dataset for head reenactment. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022. 1, 3

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 5

[43] Marek Kowalski, Stephan J. Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[44] Alexandros Lattas, Stylianos Moschoglou, Stylianos Ploumpis, Baris Gecer, Jiankang Deng, and Stefanos Zafeiriou. Fitme: Deep photorealistic 3d morphable model avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8629–8640, June 2023. 2

[45] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *CVPR*, 2023. 3

[46] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2

[47] Connor Z. Lin, Koki Nagano, Jan Kautz, Eric R. Chan, Umar Iqbal, Leonidas Guibas, Gordon Wetzstein, and Sameh Khamis. Single-shot implicit morphable faces with consistent texture parameterization. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[48] Peter C. Litwinowicz. Inkwell: A 2-d animation system. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, pages 113–122, New York, NY, USA, 1991. ACM. 1

[49] Live2D Inc. What is live2d. https://www.live2d.com/en/about/, 2023. Accessed: 2023-08-07. 1

[50] lixunsong, liangyang mt, kegeyang, npjd, and songtao-liu mt. Moore-AnimateAnyone. https://github.com/MooreThreads/Moore-AnimateAnyone, 2024. Accessed: 2024-09-08. 8

[51] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2

[52] Bryan Lufkin. The virtual vloggers taking over youtube. *BBC Worklife*, Oct 2018. 1

[53] M2 Inc. Character animation tool e-mote. https://emote.mtwo.co.jp/, 2023. Accessed: 2023-08-07. 1

[54] Zhiyuan Ma, Xiangyu Zhu, Guojun Qi, Zhen Lei, and Lei Zhang. Otavatar: One-shot talking face avatar with controllable tri-plane rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[55] Hololive Management. Nekomata okayu (official). https://3d.nicovideo.jp/works/td63648, 2019. Accessed: 2024-09-08. 8

[56] Hololive Management. Yuukoku robel (official). https://3d.nicovideo.jp/works/td83164, 2019. Accessed: 2024-09-08. 8

[57] Hololive Management. Hakui koyori (official). https://3d.nicovideo.jp/works/td84748, 2022. Accessed: 2024-09-08. 8

[58] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space, 2019. 2

[59] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 2

[60] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 3

[61] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. Pagan: Real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6), dec 2018. 3

[62] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, pages 598–613, 2020. 2

[63] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[64] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation, 2019. 2

[65] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[66] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 2

[67] A. Pumarola, A. Agudo, A.M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: One-shot anatomically consistent facial animation. In *International Journal of Computer Vision (IJCV)*, 2019. 3

[68] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H. Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *ICCV*, 2021. 3

[69] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 3

[70] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 2

[71] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery. 3

[72] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *CVPR*, 2023. 2

[73] J. M. Saragih, S. Lucey, and J. F. Cohn. Real-time avatar animation from a single image. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 213–220, 2011. 2

[74] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis, 2021. 2

[75] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[76] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, December 2019. 2

[77] Aliaksandr Siarohin, Oliver Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *CVPR*, 2021. 2

[78] Rajhans Singh, Ankita Shukla, and Pavan Turaga. Polynomial implicit neural representations for large diverse datasets. In *CVPR*, 2023. 2

[79] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2, 4

[80] SSS LLC. Touhoku Zunko · Zundamon PJ Official HP. `https://zunko.jp/`, 2023. Accessed: 2023-10-03. 7

[81] SSS LLC. zzm_a1zunko11.png. `https://zunko.jp/sozai/zunkot_s/zzm_a1zunko11.png`, 2023. Accessed: 2023-10-03. 7

[82] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, jun 2007. 2

[83] Jingxiang Sun, Xuan Wang, Lizhen Wang, Xiaoyu Li, Yong Zhang, Hongwen Zhang, and Yebin Liu. Next3d: Generative neural texture rasterization for 3d-aware head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20991–21002, June 2023. 2, 3

[84] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2, 4

[85] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, june 2020. 3

[86] Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. In *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia)*, volume 39, December 2020. 3

[87] Transpchan. Collaborative neural rendering using anime character sheets. https://github.com/transpchan/transpchan.github.io/blob/57efe17cdce35cf2c49c8d11ebd9bac108d1ac59/live3d/CoNR.pdf, 2022. Accessed: 2023-08-07. 1

[88] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[89] Evangelos Ververas and S. Zafeiriou. Slidergan: Synthesizing expressive face images by sliding 3d blendshape parameters. *International Journal of Computer Vision*, pages 1–22, 2020. 3

[90] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[91] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 5

[92] Huawei Wei, Zejun Yang, and Zhisheng Wang. Aniportrait: Audio-driven synthesis of photorealistic portrait animations, 2024. 2

[93] Chen Henry Wu and Fernando De la Torre. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance, 2022. 3

[94] You Xie, Hongyi Xu, Guoxian Song, Chao Wang, Yichun Shi, and Linjie Luo. X-portrait: Expressive portrait animation with hierarchical motion attention. In *arXiv preprint arXiv:2403.15931*, 2024. 2

[95] Hongyi Xu, Guoxian Song, Zihang Jiang, Jianfeng Zhang, Yichun Shi, Jing Liu, Wanchun Ma, Jiashi Feng, and Linjie Luo. Omniavatar: Geometry-guided controllable 3d head synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[96] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *CVPR*, 2024. 2

[97] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *ACM Trans. on Graphics*, 39, 2020. 2

[98] Shurong Yang, Huadong Li, Juhao Wu, Minhao Jing, Linze Li, Renhe Ji, Jiajun Liang, and Haoqiang Fan. Megactor: Harness the power of raw video for vivid portrait animation, 2024. 2

[99] Yanchao Yang, Ganesh Sundaramoorthi, and Stefano Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4408–4416, 2015. 2

[100] T Yenamandra, A Tewari, F Bernard, HP Seidel, M Elgharib, D Cremers, and C Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[101] Bohan Zeng, Xuhui Liu, Sicheng Gao, Boyu Liu, Hong Li, Jianzhuang Liu, and Baochang Zhang. Face animation with an attribute-guided diffusion model, 2023. 2

[102] Jiale Zhang, Chengxin Liu, Ke Xian, and Zhiguo Cao. Hierarchical feature warping and blending for talking head animation. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2024. 1, 3

[103] Jiale Zhang, Ke Xian, Chengxin Liu, Yinpeng Chen, Zhiguo Cao, and Weicai Zhong. Cptnet: Cascade pose transform network for single image talking head animation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020. 1, 3

[104] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 5

[105] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. Sadtalker: Learning realistic 3d motion coefficients for stylized audio-driven single image talking face animation. In *CVPR*, 2023. 3

[106] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation, 2022. 2

[107] Mingwu Zheng, Hongyu Yang, Di Huang, and Liming Chen. Imface: A nonlinear 3d morphable face model with implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20343–20352, 2022. 2

[108] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow, 2017. 2, 3

[109] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[110] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. In *European Conference on Computer Vision*, 2022. 2