This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

StyleAvatar: Stylizing Animatable Head Avatars

Juan C. Pérez[1,2] Thu Nguyen-Phuoc[1] Chen Cao[1] Artsiom Sanakoyeu[1]
Tomas Simon[1] Pablo Arbeláez[3] Bernard Ghanem[2] Ali Thabet[1] Albert Pumarola[1]

[1]Meta [2]KAUST [3]Universidad de los Andes

Figure 1. ***StyleAvatar* for stylizing animatable head avatars.** We present StyleAvatar, an easy-to-use method allowing casual users to stylize their personalized head avatars via images or text. Our method offers control over the stylization strength, *i.e*. fidelity to the original avatar's identity, while preserving the avatar's fundamental animation capabilities. Furthermore, our method achieves consistent stylization effects across individuals and styles, preserves compelling photo-realism even in extreme views and intense facial expressions, and provides disentangled control of texture and geometry. We demonstrate the compatibility of StyleAvatar with existing technologies by deploying it into an AR/VR headset.

Abstract

AR/VR applications promise to provide people with a genuine feeling of mutual presence when communicating via their personalized avatars. While realistic avatars are essential in various social settings, the vast possibilities of a virtual world can also generate interest in using stylized avatars for other purposes. We introduce **StyleAvatar**, *the first method for semantic stylization of animatable head avatars. StyleAvatar directly stylizes the avatar representation, rather than stylizing its renders. Specifically, given a model generating the avatar, StyleAvatar first disentangles geometry and texture manipulations, and then stylizes the avatar by fine-tuning a subset of the model's weights. Our method has multiple virtues, including the ability to describe styles using images or text, preserving the avatar's animatable capacity, providing control over identity preservation, and disentangling texture and geometry modifications. Experiments have shown that our approach consistently works across skin tones, challenging hair styles, extreme views, and diverse facial expressions.*[1]

[1]Work performed as part of Juan's internship at Meta.

1. Introduction

Augmented and virtual reality (AR/VR) technologies hold immense promise in providing a sense of telepresence and realism across distances. With the ability to generate accurate and expressive face avatars, users can precisely replicate their expressions, thus creating a sense of comfort and realism in their interactions with other parties. The realism of avatars is crucial, as it ensures a precise representation of facial intricacies like facial hair, scars, and tattoos.

While photorealistic avatars are essential for formal settings such as family reunions and work meetings, there are situations where people may prefer to present themselves differently. The limitless possibilities of a virtual world should enable individuals to express themselves via their avatars in a wide range of ways, from stylized versions (*e.g*. other textures) to completely different faces (*e.g*. becoming a dragon). Providing this flexibility fosters a more diverse range of appealing experiences, which is vital for platforms aiming to offer meaningful AR/VR experiences.

To enable casual users to customize their appearance without requiring technical proficiency, platforms should

provide tools that empower users to easily do so. This comfort is necessary to ensure fair access to everyone. While we have witnessed significant momentum in photorealistic and animatable avatars [3, 9, 27], there is still a lack of automated capabilities to easily stylize these representations.

In this paper, we fill this gap by introducing **StyleAvatar**, the first method for stylizing animatable head avatars. By directly stylizing the avatar representation, our method provides a high-resolution stylization that preserves the avatar's animatable capabilities, thus providing consistent appearance across facial expressions and views. Stylization can be guided by images or text, and fidelity to the person's identity can be easily tuned, allowing for soft and intense stylizations. Moreover, geometry and texture manipulations are disentangled. The method demonstrates consistent stylization capacity across styles and identities, as well as challenging hair types and diverse skin tones. Please refer to Figure 1 for an illustration of the capabilities of StyleAvatar, and its deployment on an AR/VR headset.

We summarize our contributions as follows: **(i)** We present StyleAvatar, the first method for semantic stylization of photorealistic 3D head avatars. By directly operating on the avatar representation, our method provides high-resolution stylizations across diverse skin tones and challenging hair types. **(ii)** Our pipeline preserves the avatars' animatable capabilities, providing consistent stylized appearance across extreme views and facial expressions. **(iii)** Our method provides disentangled control of geometry and texture. Our experiments validate these properties across styles and identities, demonstrating StyleAvatar's effectiveness in stylizing, while preserving the avatar's fundamental driving capabilities for AR/VR applications.

2. Related Work

In this section, we cover stylization methods in 2D and 3D. We briefly list here avatar-generation methods [3, 9, 26, 27], but will not cover them in-depth since our work focuses on stylizing, rather than generating, head avatars.

Stylizing in 2D. The seminal work of Gatys *et al*. [11] manipulated images to follow artistic styles, and spurred interest in leveraging deep learning for stylization. This interest increased further with the surfacing of GANs [12]. In particular, the advent of StyleGAN [21] stimulated a plethora of methods that exploited its generative capabilities for stylizing images [17, 29, 45]. Recently, Toonify [39] demonstrated generation of novel styles by interpolating between different StyleGAN models, and was later extended to videos [46]. AgileGAN [41] performs portrait stylization by inverting to GAN latent space and performing operations there. A different group of methods leverage CLIP's semantic association of images and text [40] to derive and edit style [4]. StyleCLIP [38] edits images by finding directions in StyleGAN's latent space via CLIP. StyleGAN-NADA [10] further introduced a directional CLIP loss, and used it to fine-tune StyleGAN for novel domains. Our method, StyleAvatar, uses a CLIP-guided loss to stylize the avatar. In contrast to the methods mentioned above, StyleAvatar directly stylizes the 3D representation (via texture and geometry manipulations) rather than stylizing 2D images.

Stylizing in 3D. The field of generation and stylization of 3D objects and scenes is rapidly growing. Classical methods used traditional shape representations like point clouds and meshes [6, 8, 13, 15, 20, 30, 48]. Recent approaches now take advantage of neural rendering pipelines to synthesize realistic 3D objects and scenes [5, 22, 32, 49]. Dream Fields [19] leverages NeRFs [31] to generate 3D objects from text, achieving impressive results. StylizedNeRF [18] also uses NeRFs for implicit 3D understanding, while creating stylized scene renderings. A similar work, CIPS-3D [50], uses NeRFs to design a 3D-aware GAN generator. CLIP-Mesh [24] uses an input text prompt to optimize the vertices of a control shape and create novel 3D objects. AvatarCLIP [16] generates full-body human avatars, introducing both appearance and movement into the avatar. Focusing on style transfer in 3D, SNeRF [36] generates artistically-stylized novel views of scenes by updating a NeRF based on image statistics [11]. The methods mentioned in this section provide strong pipelines to generate and stylize 3D content. However, in contrast to our approach, none have leveraged semantic understanding for stylizing a drivable 3D asset such as an avatar. Namely, StyleAvatar aims at semantically-grounded stylization, and achieves consistent styles across views and expressions. Concurrent to our work, Instruct-NeRF2NeRF [14] uses a diffusion model to edit a NeRF by modifying its training dataset according to a language instruction.

3. Method: StyleAvatar

Personalized avatars can be represented by a model that maps the appearance of a person, described by their texture and geometry, to a volumetric representation. This representation allows for rendering the avatar from different perspectives. To account for facial expressions, the model can further be conditioned on features that encode expressions.

We propose StyleAvatar, a novel method for stylizing a person's avatar using a style described by images or text. StyleAvatar achieves this by modifying the person's model to disentangle the texture and geometry of the avatar, and fine-tuning a subset of the model's weights. An overview of the method is provided in Figure 2. Next, we describe the model's architecture, our procedure for disentangling texture and geometry, and our optimization objective.

3.1. Preliminaries: avatar architecture

We use the Instant Avatar architecture introduced in [3], the current state-of-the-art for realistic and personalized avatars. Essentially, this architecture is an encoder-decoder model that leverages rendering based on Mixture of Volu-

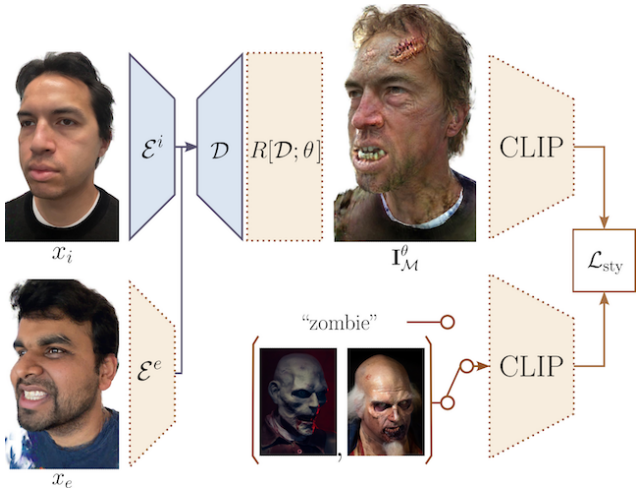

Figure 2. **StyleAvatar overview.** StyleAvatar stylizes an avatar by fine-tuning a disentangled avatar architecture. We first disentangle geometry and texture modifications with the procedure described in Section 3.2, and then perform optimization. For a given identity and expression, we compute a CLIP loss between renders of the avatar, and CLIP embeddings of the target style. We use this loss to fine-tune the identity encoder, $\mathcal{E}^{\text{i}}$, and the decoder, $\mathcal{D}$, that generate the avatar. Blue indicates the modules we fine-tune.

metric Primitives [26] for photo-realistic results. The identity of the produced avatar is defined by the model's specific weights, that is, each person has their own set of weights.

The inputs to this model are an identity—the person's appearance—denoted by x_{i}, and an expression, denoted by x_{e}. Both x_{i} and x_{e} are pairs of a position map (representing a mesh) and a texture (representing color appearance). These identity-expression inputs are mapped to an avatar of the given identity exhibiting the specified expression. Avatars should desirably disentangle the factors of identity and expression. To achieve this objective, these factors are separately processed by two independent encoders (correspondingly called $\mathcal{E}^{\text{i}}$ and $\mathcal{E}^{\text{e}}$), whose outputs are processed by decoder $\mathcal{D}$ to generate the avatar. Formally, the renderable outputs of the model are $\mathcal{M}(x_{\text{i}}, x_{\text{e}}) = \mathcal{D}\left(\mathcal{E}^{\text{i}}(x_{\text{i}}), \mathcal{E}^{\text{e}}(x_{\text{e}})\right)$. By defining θ as the camera parameters and $R[\,\cdot\,;\,\theta]$ as the rendering operator conditioned on θ, a rendered image of the avatar with model $\mathcal{M}$ is

$$\mathbf{I}^\theta_\mathcal{M} = R\left[\mathcal{M}(x_{\text{i}}, x_{\text{e}}); \theta\right] = R\left[\mathcal{D}\left(\mathcal{E}^{\text{i}}(x_{\text{i}}), \mathcal{E}^{\text{e}}(x_{\text{e}})\right); \theta\right]. \tag{1}$$

In this framework, manipulating an avatar's appearance amounts to manipulating the architecture's components that are associated with the avatar's identity. StyleAvatar thus stylizes an avatar by manipulating its identity. Specifically, StyleAvatar fine-tunes a subset of $\mathcal{M}$'s modules, corresponding to both $\mathcal{E}^{\text{i}}$ and $\mathcal{D}$, while leaving $\mathcal{E}^{\text{e}}$ fixed.

3.2. Disentangling texture and geometry

In traditional graphics, face identity is described by two primary factors: texture and geometry. Our intended identity transformation thus reduces, essentially, to manipulating these two factors. While texture and geometry are typically treated separately in graphics pipelines, they are intertwined in the method we target, as the Instant Avatar architecture combines these inputs at several stages in the forward pass to generate realistic avatars. Hence, modifying $\mathcal{E}^{\text{i}}$ and $\mathcal{D}$ results in entangled edits to the avatar's texture and geometry. That is, despite the impressive photo-realistic avatars produced by the Instant Avatar architecture, its inner workings prevent disentangled editing of texture and geometry, making our task challenging.

StyleAvatar circumvents this limitation by disentangling these factors in the architecture. First, we note that both the identity encoder $\mathcal{E}^{\text{i}}$ and the decoder $\mathcal{D}$ are internally divided into geometry and texture branches, *i.e.* $\mathcal{E}^{\text{i}} = \{\mathcal{E}^{\text{i}}_{\text{geo}}, \mathcal{E}^{\text{i}}_{\text{tex}}\}$ and $\mathcal{D} = \{\mathcal{D}_{\text{geo}}, \mathcal{D}_{\text{tex}}\}$. These branches inherit the names of the inputs received by their encoders: $\mathcal{E}^{\text{i}}_{\text{geo}}$ receives a position map (representing a mesh), while $\mathcal{E}^{\text{i}}_{\text{tex}}$ receives a texture map. Figure 3 (left) illustrates this internal division.

Under this configuration, the branch processing geometry is composed of $\{\mathcal{E}^{\text{i}}_{\text{geo}}, \mathcal{D}_{\text{geo}}\}$, while the one processing texture is composed of $\{\mathcal{E}^{\text{i}}_{\text{tex}}, \mathcal{D}_{\text{tex}}\}$. The configuration of this architecture, while promising for our purposes, connects encoders with decoders via a set of skip connections S. In the original implementation of Cao *et al.* [3], S connects the forward passes of the geometry and texture branches, as illustrated in Figure 3 (top-right), via a bias layer B. This implementation thus suffers from geometry-texture entanglement. We fix this entanglement by introducing simple modifications on S.

Please refer to Figure 3 (bottom-right) for an illustration of our architecture modifications. Specifically, we note that, inside S, the bias layer B receives geometry and texture features as input (b_{geo} and b_{tex}), and feeds its output to *both* $\mathcal{D}_{\text{geo}}$ and $\mathcal{D}_{\text{tex}}$. To prevent this connection during fine-tuning, we perform two steps. First, we freeze the initial values for b_{geo} and b_{tex} (denoted by $\hat{b}_{\text{geo}}$ and $\hat{b}_{\text{tex}}$ in Figure 3 (bottom-right)). Second, we split B into two layers, B_{geo} and B_{tex}, whose corresponding inputs, computed in the first step, are kept fixed during optimization.

With these modifications, we can independently edit geometry and texture by simply fine-tuning the corresponding encoder-decoder weights. That is, geometry can be controlled by modifying $\{\mathcal{E}^{\text{i}}_{\text{geo}}, \mathcal{D}_{\text{geo}}\}$, while texture can be controlled by modifying $\{\mathcal{E}^{\text{i}}_{\text{tex}}, \mathcal{D}_{\text{tex}}\}$.

3.3. CLIP-guided stylization

To stylize the animatable head avatars, StyleAvatar utilizes a CLIP direction loss [10]. This loss function is designed to fine-tune a StyleGAN generator from a source to a target domain by "sliding" the generator away from the original generator along a specific direction in CLIP's space. The fine-tuning process involves using a frozen copy

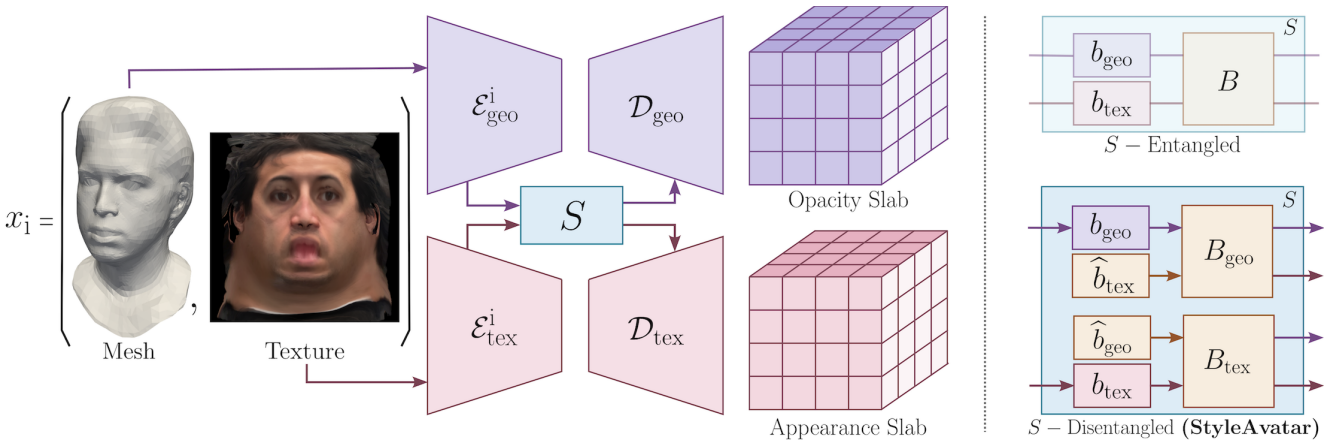

Figure 3. **StyleAvatar disentangles geometry and texture in the Instant Avatar architecture.** Left: In this architecture, identity x_i is represented as a mesh and a texture. The mesh is processed by the "geometry" branch ($\mathcal{E}^i_{geo}$ and $\mathcal{D}_{geo}$), and the texture is processed by the "texture" branch ($\mathcal{E}^i_{tex}$ and $\mathcal{D}_{tex}$). This architecture connects encoders and decoders with skip connections S. Right, top: The original implementation of Cao *et al*. ("S–Entangled") uses **entangled skip connections** S connecting the forward passes of both branches, and thus suffers from geometry-texture entanglement. Specifically, geometry outputs (*i.e*. the opacity slab) are affected by the texture branch; conversely, appearance outputs (*i.e*. the appearance slab) are affected by the geometry branch. Right, bottom: StyleAvatar uses **disentangled skip connections** S, effectively splitting the architecture into two *separate* encoder-decoder models. Notice our modification allows using pre-trained weights.

of the original model during optimization. The optimized model is encouraged to generate images that differ from those generated by the frozen model *only* along a specified target direction in CLIP space. This objective is achieved by enforcing the optimized images to follow a direction that is *parallel* to the given target direction.

Here, we leverage this loss to fine-tune the disentangled architecture. We use CLIP to process texts and renders of the avatar, and keep a frozen copy $\mathcal{M}^*$ of the avatar model throughout optimization. The target CLIP-space direction, $\mathbf{d}_{tgt} = \mathbf{e}_{tgt} - \mathbf{e}_{src}$, is dictated by the embeddings $\mathbf{e}_{src}$ and $\mathbf{e}_{tgt}$ describing the source and target styles. The optimized direction is computed between CLIP embeddings of renders of the avatar being stylized ($\mathbf{I}_{\mathcal{M}^*}$) and the original avatar ($\mathbf{I}_{\mathcal{M}}$). Formally, the stylization loss we optimize is

$$\mathcal{L}_{sty}(\mathcal{M}^*, \mathcal{M}) = D_{cos}\left(f\left(\mathbf{I}^{\theta}_{\mathcal{M}^*}\right) - f\left(\mathbf{I}^{\theta}_{\mathcal{M}}\right), \mathbf{d}_{tgt}\right), \quad (2)$$

where, D_{cos} is the cosine distance, $\mathcal{M}^*$ and $\mathcal{M}$ denote the stylized and frozen avatar models, respectively, and $f(\cdot)$ is the CLIP image encoder. Similar to StyleCLIP [38], guiding stylization based on text or images is reduced to the manner in which $\mathbf{e}_{tgt}$ and $\mathbf{e}_{src}$ are computed. That is, for text guidance, $\mathbf{e}_{tgt}$ and $\mathbf{e}_{src}$ are CLIP text embeddings with template augmentations [38,40]. On the other hand, for image guidance, $\mathbf{e}_{tgt}$ is the target style images' embedding, and $\mathbf{e}_{src}$ is an embedding of a render of the original avatar.

3.4. Regularization

StyleAvatar uses two regularizers, one to control identity preservation and another to control asymmetrical artifacts in the avatar. Next, we describe these regularizers in detail.

Identity preservation. Ensuring the preservation of identity is crucial when stylizing personalized avatars. Off-the-shelf avatars may suffice for users who do not require identity preservation, but it is necessary to preserve key facial features to enable identification of the avatar's owner. To achieve this purpose, StyleAvatar incorporates a regularizer that controls the preservation of key facial features during the stylization process. Note that a naïve regularizer based on face recognition models may be unsuitable for stylized faces. Therefore, we adopt the image-structure regularizer proposed by Bar-Tal *et al*. [2]. This regularizer preserves the spatial layout, shape, and perceived semantics between two images, serving as a proxy for facial features and structure. Specifically, the regularizer operates on the self-similarity matrices of an image's features, which correspond to the tokens extracted from the image by CLIP. Namely, by defining $f^i(\cdot)$ as the i^{th} CLIP token, the entries of the self-similarity matrix are given by

$$\mathbf{S}\left(\mathbf{I}\right)_{i,j} = 1 - D_{cos}\left(f^i\left(\mathbf{I}\right), f^j\left(\mathbf{I}\right)\right).$$

The regularizer is then defined as the Frobenius norm between these self-similarity matrices. Formally, the loss is:

$$\mathcal{L}_{id}(\mathcal{M}^*, \mathcal{M}) = \|\mathbf{S}\left(\mathbf{I}^{\theta}_{\mathcal{M}^*}\right) - \mathbf{S}\left(\mathbf{I}^{\theta}_{\mathcal{M}}\right)\|_F.$$

Symmetry. Some stylizations introduce undesirable asymmetrical artifacts in the avatars, as shown Figure 7 in Section 4. While human perception tolerates (and can even find desirable) some degree of asymmetry in the face, asymmetric artifacts *between* the eyes are particularly disturbing to humans [43]. To address this issue, StyleAvatar leverages a regularizer that specifically targets asymmetrical artifacts

Figure 4. **StyleAvatar stylization across identities and styles.** Here, each row is an identity, and each column is a style. Note how StyleAvatar preserves key facial features of each subject, while introducing believable changes to the avatars' appearance. Furthermore, StyleAvatar provides consistent stylizations across subjects, and manipulates both texture and geometry, *e.g.* the "Boterismo" style correctly introduces the painter's traditional color palettes and exaggerated face sizes.

around the avatar's eyes. The regularizer is formulated as an SSIM [44] loss comparing the two eyes from a frontal view. By defining Θ as the camera parameters corresponding to a frontal view of the avatar, our loss is defined as:

$$\mathcal{L}_{\text{sym}}(\mathcal{M}^*) = \text{SSIM}\left(\text{eye}_{\text{L}}\left(\mathbf{I}^{\Theta}_{\mathcal{M}^*}\right), \text{eye}_{\text{R}}\left(\mathbf{I}^{\Theta}_{\mathcal{M}^*}\right)\right),$$

where the extraction of the eye region is enabled by our direct control over the avatar and its renders. As such, this process is independent of additional face-parsing pipelines.

3.5. Loss

The overall objective for stylization is defined as:

$$\arg\min_{\mathcal{M}^*} \mathop{\mathbb{E}}_{\theta, x_{\text{e}}} \left[\, \mathcal{L}_{\text{sty}} + \lambda_{\text{id}}\, \mathcal{L}_{\text{id}} + \lambda_{\text{sym}}\, \mathcal{L}_{\text{sym}} \right], \qquad (3)$$

where the expected value is taken over camera parameters θ and facial expressions x_{e} from Equation (1). We experimentally set the regularizers to $\lambda_{\text{id}} = 10$ and $\lambda_{\text{sym}} = 12$.

4. Experiments

Next, we present an extensive evaluation of the architectural modifications and losses introduced in StyleAvatar. We evaluate our method on the data captured in [3].

4.1. Implementation details

We fine-tune with Adam [25] with a learning rate of 10^{-3}. We run 400 optimization steps, which require around 30 minutes on an NVIDIA V100 GPU.

Batches consist of renders of the original and stylized avatars. We augment data for generating avatar renders and computing the losses. That is, we implement Equation (3) by randomizing the camera's azimuth and the avatar's expression. For computing the style loss from Equation (2), we use image augmentations proposed in [2].

4.2. Main results

Appearance. We showcase the results of our proposed StyleAvatar method for various styles and identities in Figure 4. The results demonstrate that StyleAvatar is capable of achieving high fidelity to the original avatar while introducing physically-plausible stylizations of the avatar's appearance. Furthermore, note how our stylization is consistent across identities, while also leveraging each person's facial features to fit the desired style. For instance, note how the "manga" style slightly modifies eyebrow shape, and hair/skin tone to achieve realistic stylization, while preserv-

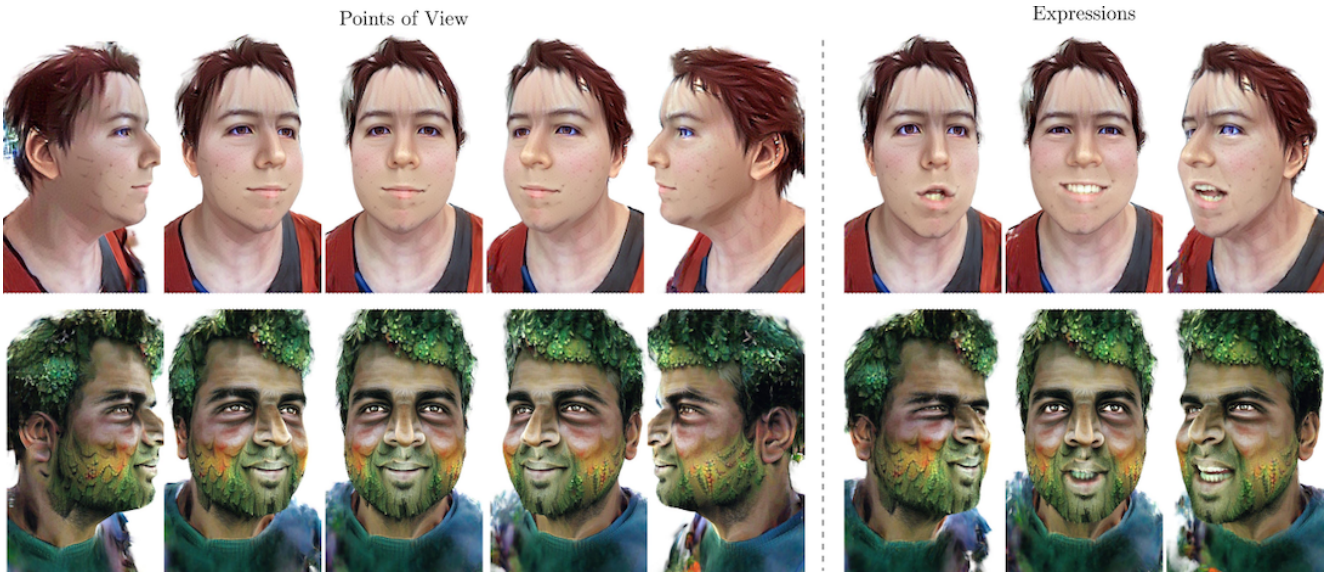

Figure 5. **Appearance consistency across viewpoints and facial expressions.** StyleAvatar provides consistent appearance when the avatar is *(i)* viewed from multiple (and even extreme) points of view, and *(ii)* exhibiting (intense) facial expressions.

ing the individual's most salient facial features.

Multi-view and expression consistency. We visualize the stylized avatar under various camera poses and facial expressions in Figure 5. Our results demonstrate that StyleAvatar's stylization is consistent across different views and expressions, which is fundamental for the avatar's usefulness in AR/VR applications. Without this consistency, the avatar would appear artificial and erratic, potentially leading to uncanny valley effects [33, 34]. Please refer to the **Supplementary Materials** for video samples showcasing our method's consistency across angles and expressions.

4.3. Analysis

Identity preservation. StyleAvatar allows for controlling the balance between identity preservation and stylization strength via the identity regularizer λ_{id}. Figure 6 shows the effect of varying this regularizer. We observed that small regularizer values lead to significant changes in appearance that hinder the recognition of the avatar's owner, which is consistent with findings in psychology [42]. Conversely, large regularizer values strongly preserve facial features while decreasing the degree of stylization. Offering users the ability to control this aspect of stylization is crucial. However, after experimentation, we found that $\lambda_{\text{id}} = 10$ offers a reasonable trade-off between identity preservation and stylization strength, and therefore selected this as the default value for StyleAvatar.

Eye symmetry. The symmetry regularizer λ_{sym}, introduced in Section 3.4, addresses the unpleasant artifacts that can appear around the avatars' eyes in certain styles. The presence of these asymmetrical artifacts is sporadic and can occur without the proposed regularization, as illustrated in the left column of Figure 7. Our experiments indicate that incorporating λ_{sym} effectively resolves these artifacts. To maintain a balance between the removal of these artifacts and oversmoothing effects in the stylization result, we set the default value for λ_{sym} to 12 in StyleAvatar.

Visualizing geometry modifications. The volumetric nature of the avatar lends itself to visualizing the geometric changes induced by StyleAvatar. To this end, we color the stylized geometry according to the output of a Non-Rigid Iterative Closest Point (NR-ICP) [1] algorithm. Specifically, we assign colors to vertices in the stylized mesh based on their distance to the original mesh, as determined by the NR-ICP algorithm. To generate the meshes, we use Poisson surface reconstruction [7, 23] on point clouds obtained from renders of the avatar with a neutral expression.

We use StyleAvatar to stylize the avatar for *"obese"* and *"skinny"* text queries, and visualize the resulting geometry deformations in Figure 8. These stylizations introduce sizable appearance changes. However, upon comparing the stylized appearance with the geometric visualization, we observe a bias towards introducing changes in radiance (*i.e.*, the adversarial-like noise in the obese-style forehead) over changes in geometry. Although the avatar appears to have undergone significant geometric modifications, most of the modifications that optimize our proposed losses are those in radiance. Thus, we find that StyleAvatar displays a preference for modifying appearance via texture changes. This finding aligns with previous observations in neural rendering [37, 47], whereby appearance modifications can be achieved by changing texture while mostly disregarding disregarding geometry. Furthermore, this result aligns with findings in cosmetic science, where people can alter their appearance via optical illusions using makeup [28, 35].

Figure 6. **StyleAvatar control of identity preservation—stylization strength.** The rows depict, respectively, "zombie" and "scary clown" style. From left to right, we introduce and progressively increase the identity-preservation regularizer λ_{id}. The column with $\lambda_{id} = 0$ shows how the lack of regularization results in the introduction of dramatic appearance changes that could hinder recognizing the avatar's owner. On the other hand, a large regularizer of $\lambda_{id} = 20$ allows to easily distinguish the person; however, the stylization strength is compromised, and so the stylization may not be easily perceived. We find that $\lambda_{id} = 10$ provides a sensible trade-off between both (undesirable) phenomena.

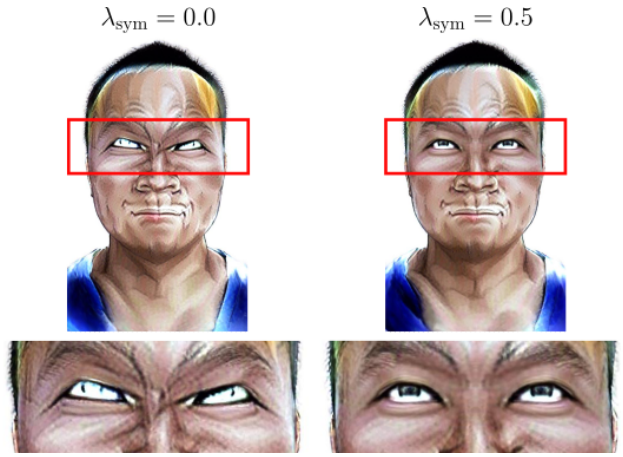

Figure 7. **Eye symmetry loss ablation.** Our symmetry regularizer prevents stylization from sporadically introducing undesirable artifacts in the avatar's eyes. These artifacts strongly affect the avatar's appearance, and have even stronger consequences when animating the avatar.

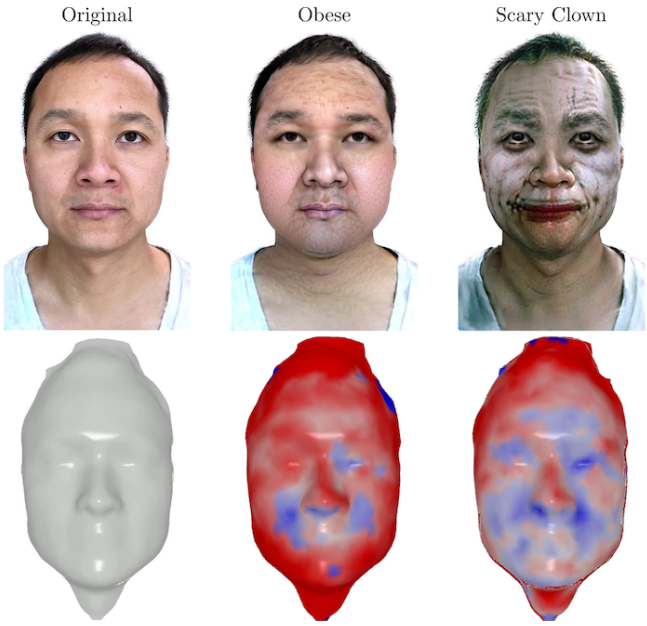

Figure 8. **Visualizing geometric changes.** We color the vertices of the stylized geometry according to their displacement w.r.t. the original geometry. Blue/red colors indicate inwards/outwards displacement. Stylization displays a strong bias for modifying appearance by concentrating modifications to radiance, rather than geometry. Specifically, note how a geometry-inclined objective of *"obese"* is satisfied by introducing large changes in radiance.

4.4. Comparisons

Portrait-based semantic stylization. To the best of our knowledge, no other work has attempted to provide semantic stylization of animatable avatars. To establish a baseline for comparison, we consider portrait stylization, where the avatars are stylized by directly altering their rendered images. Specifically, we compare the performance of StyleAvatar against StyleGAN-NADA [10], which involves inverting the portrait to StyleGAN's latent space, fine-tuning the generator to the target style, and then forwarding the latent

Figure 9. **Comparison against other stylization approaches.** We compare against *(i)* StyleGAN-NADA (columns 5-7), a method for stylizing portrait images, and *(ii) Image style transfer*, Gatys *et al*. (columns 8-10), a method for artistic stylization of images. As expected, there are clear disadvantages to stylizing individual renders of the avatar, either semantically (*i.e*. StyleGAN-NADA), or artistically (*Image style transfer*): mode collapse and inconsistent appearance across views and expressions, among others. Notably, directly operating on the avatar has the advantage of preserving identity, facial pose, facial expression and background.

code through the fine-tuned generator. We present the results of this comparison in columns 5 to 7 of Figure 9. As expected, operating directly on the render has several shortcomings when compared to stylizing the avatar representation. Our findings demonstrate that StyleAvatar outperforms StyleGAN-NADA in preserving the identity, facial pose, and expression of the avatars. These characteristics are crucial for maintaining a coherent animatable avatar.

Non-semantic stylization. We also compare StyleAvatar against a method for artistic (*i.e*. *non-semantic*) stylization based on local image statistics. In particular, we use the seminal work of Gatys *et al*. [11] on artistic style transfer with neural networks. This technique is designed to modify a content image to match the artistic style of another image, such as rendering a landscape in the style of Van Gogh. In the last three columns of Figure 9, we demonstrate the difference between StyleAvatar's semantic stylization and that of [11]. While [11] produces mesmerizing results, its per-image nature hinders coherent avatar stylization across variations in point of view and facial expressions.

5. Conclusions and Limitations

We presented StyleAvatar, a novel method for stylizing animatable head avatars. StyleAvatar disentangles texture and geometry and then conducts CLIP-guided [40] optimization to fit a target style. Our method operates on a model that was pre-trained on a limited set of a few hundred identities [3]. Despite the constraints of the training domain (*i.e*. real images of people), we observe, akin to observations in [10], that CLIP provides useful feedback to shift the model from the original realistic domain to the desired stylized domain. We highlight several key advantages of StyleAvatar, including its ability to accept text and image guidance, its flexibility across styles and identities, its preservation of the avatar's driving capabilities, and its ability to control the fidelity of the stylization to the original identity. These advantages are essential in empowering casual users to create stylized avatars that meet their needs.

Despite StyleAvatar's strengths, we observe some limitations. Firstly, our stylizations are not cartoon-ish (in contrast to [10,39,46]), but rather costume-ish: results look like *plausible* ways in which a person could use heavy make-up. Secondly, the stylization process can occasionally be unstable, and produce undesirable artifacts that are not easily resolved via regularization. Thirdly, while our architecture disentangles texture from geometry, we still find that direct geometry manipulation can be challenging and may result in degenerate solutions with hole-like artifacts [26]. Finally, our method is not scalable to a large number of identities/styles, as it requires the optimization of each identity-style pair separately. We attribute some of these limitations to the difficulty of manipulating implicit representations [26] through CLIP, as observed in [16, 19, 30].

References

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2007. 6

[2] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *European Conference on Computer Vision (ECCV)*, 2022. 4, 5

[3] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)*, 2022. 2, 3, 5, 8

[4] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-based clip-guided essence transfer. *arXiv preprint arXiv: 2110.12427*, 2021. 2

[5] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu. Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene. In *arxiv*, 2022. 2

[6] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022. 2

[7] Dawson-Haggerty et al. trimesh. 6

[8] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. *arXiv preprint arXiv:2204.01943*, 2022. 2

[9] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021. 2

[10] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2022. 2, 3, 7, 8

[11] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 8

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 2

[13] Fangzhou Han, Shuquan Ye, Mingming He, Menglei Chai, and Jing Liao. Exemplar-based 3d portrait stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2021. 2

[14] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[15] Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6198–6208, 2022. 2

[16] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)*, 2022. 2, 8

[17] Jialu Huang, Jing Liao, and Sam Kwong. Unsupervised image-to-image translation via pre-trained stylegan2 network. *IEEE Transactions on Multimedia*, 2021. 2

[18] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[19] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 8

[20] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2human: Text-driven controllable human image generation. *ACM Transactions on Graphics (TOG)*, 2022. 2

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2

[22] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[23] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 6

[24] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. Clip-mesh: Generating textured meshes from text using pretrained image-text models. December 2022. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 2021. 2, 3, 8

[27] Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[28] Soyogu Matsushita, Kazunori Morikawa, and Haruna Yamanami. Measurement of eye size illusion caused by eyeliner, mascara, and eye shadow. *Journal of cosmetic science*, 66(3):161–174, 2015. 6

[29] Yifang Men, Yuan Yao, Miaomiao Cui, Zhouhui Lian, Xuansong Xie, and Xian-Sheng Hua. Unpaired cartoon image synthesis via gated cycle mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[30] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021. 2, 8

[31] 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. *Communications of the ACM*, 2021. 2

[32] Shailesh Mishra and Jonathan Granskog. Clip-based neural neighbor style transfer for 3d assets, 2022. 2

[33] Masahiro Mori. Bukimi no tani [the uncanny valley]. *Energy*, 1970. 6

[34] Masahiro Mori, Karl F MacDorman, and Norri Kageki. The uncanny valley [from the field]. *IEEE Robotics & automation magazine*, 2012. 6

[35] Kazunori Morikawa. Geometric illusions in the human face and body. *The oxford compendium of visual illusions*, pages 252–257, 2017. 6

[36] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *ACM Transactions on Graphics (TOG)*, 2022. 2

[37] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6

[38] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 4

[39] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *Machine Learning for Creativity and Design Workshop at NeurIPS 2020.*, 2020. 2, 8

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2, 4, 8

[41] Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. Agilegan: Stylizing portraits by inversion-consistent transfer learning. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2021. 2

[42] Keiko Tagai, Hitomi Ohtaka, and Hiroshi Nittono. Faces with light makeup are better recognized than faces with heavy makeup. *Frontiers in psychology*, 7:226, 2016. 6

[43] Tim T. Wang, Louis Wessels, Gazi Hussain, and Steve Merten. Discriminative Thresholds in Facial Asymmetry: A Review of the Literature. *Aesthetic Surgery Journal*, 2017. 4

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004. 5

[45] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *CVPR*, 2022. 2

[46] Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Vtoonify: Controllable high-resolution portrait video style transfer. *ACM Transactions on Graphics (TOG)*, 2022. 2, 8

[47] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 6

[48] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 2

[49] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields, 2022. 2

[50] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. 2021. 2