Ultra-High Resolution Facial Texture Reconstruction from a Single Image

Hongxiang Huang
State Key Lab of CAD&CG
Zhejiang University, P.R.China
crisprhhx@outlook.com

Guoyuan An
KAIST
Daejeon, Republic of Korea
anguoyuan111@gmail.com

Jingzhen Lan
State Key Lab of CAD&CG
Zhejiang University, P.R.China
lanjz@zju.edu.cn

Lingfei Wang
Zhejiang Lab
Hangzhou, P.R.China
leonardo22@163.com

Rui Wang
State Key Lab of CAD&CG
Zhejiang University, P.R.China
ruiwang@zju.edu.com

Yuchi Huo
State Key Lab of CAD&CG
Zhejiang University, P.R.China
huo.yuchi.sc@gmail.com

Abstract

Advances in mobile cameras have made it easier to capture ultra-high resolution (UHR) portraits, however, existing face reconstruction methods lack specific adaptations for UHR input (e.g., 4096×4096), leading to under-use of high-frequency details that are crucial for achieving photorealistic rendering.

Our method supports 4096×4096 UHR input and utilizes a divide-and-conquer approach for end-to-end 4K Albedo, Micronormal, and Specular texture reconstruction at the original resolution. We employ a two-stage strategy to capture both global distributions and local high-frequency details, effectively mitigating mosaic and seam artifacts common in patch-based predictions. Additionally, we innovatively apply hash encoding on facial UV coordinates to boost the model's ability to learn regional high-frequency feature distributions.

Our method is easy to integrate with state-of-the-art facial geometry reconstruction pipelines, and will significantly improves the texture reconstruction quality, facilitating artistic's creation workflow.

Keywords: Texture Reconstruction, Face Reconstruction, Ultra-High Resolution, Single Image Synthesis

1. Introduction

Avatars play a pivotal role in virtual worlds, with broad applications across virtual reality, gaming, and multimedia platforms [14]. Realistic personalized avatars bridge the gap between physical and digital reality [14]. Achieving lifelike facial renderings requires capturing fine details of the human face, such as pores, redness, wrinkles, and freckles, which are essential for realism [22]. Ultra-high-resolution (UHR) images (e.g., 4096×4096) provide significantly more detail compared to lower-resolution images (e.g., 512×512), capturing more nuanced facial features. While acquiring such UHR textures was once prohibitively expensive, advancements in mobile camera technology [6] now allow the capture of UHR portraits. However, existing methods [14, 34, 15] are typically constrained by memory and computational limitations, restricting input image resolution to lower sizes (e.g., 256×256 or 512×512) [19]. This downsampling results in substantial information loss, impairing the capture of high-frequency local details and degrading texture quality.

While some existing methods aim to generate high-resolution physically-based rendering (PBR) textures, they primarily only use low-resolution inputs and apply super-resolution techniques to produce high-resolution outputs. For example, MoSAR [14] and DreamFace [55] utilize specialized super-resolution networks, such as ESRGAN [51] and Real-ESRGAN [50], to upscale the initial raw outputs (512×512). Although these methods leverage priors from ultra-high-resolution datasets in their super-resolution modules, the fidelity of these outputs is still inherently limited by the resolution constraints of the model's input and output. Consequently, these methods struggle to accurately capture fine texture details directly from UHR inputs, resulting in potential inaccuracies in the generated textures.

One reason existing methods do not directly use UHR inputs is the significant GPU memory cost involved. Even processing 1K resolution is computationally expensive. For example, StyleGANv3 [24] takes over three months to train on an NVIDIA V100 GPU with 16GB of memory at 1K resolution. Scaling to 4K would require 16 times more memory (256GB), making training prohibitively expensive for many research groups.

To fully leverage 4K UHR input images, we adopt a "divide-and-conquer" approach by splitting the 4K input into patches and processing them individually. However, this presents three challenges: 1) Different facial regions

exhibit varying textures. Dividing the images does not guarantee that the corresponding patches from different faces align with the same facial region. 2) Separately generating each patch can not guarantee texture consistency and seamless stitching for the borders. 3) Local texture generation often assumes near-planar geometry, otherwise, global normals may interfere with local texture (e.g. Micronormal) prediction [20]. However, facial geometry is complex and cannot satisfy this constraint. To address these challenges, we introduce a Global Distribution Extraction stage (Stage 1, Sec. 3.3) that uses a downsampled image to determine the global distribution of textures as a guide for the next stage. Additionally, we propose a novel facial UV hash encoding technique to better capture regional information in generated guidance texture by Stage 1, which not only ensures that Stage 1 provides global consistency for Stage 2 but also delivers substantial continuous details, facilitating further refinement in Stage 2. Finally, we incorporate a geometry normal prior to ensure that local texture generation remains independent of global geometry. This allows our method to function effectively even when the original input image is non-planar.

Our final method employs a two-stage pipeline that efficiently captures both low-frequency global distributions and high-frequency local details from the UHR input. It outputs 4K Albedo, Micronormal, and Specular textures without the loss of detail typically caused by downsampling.

More specifically, Stage 1 (Global Distribution Extraction) employs an UNet-like network to generate the facial UV and normal maps from a downsampled 1024×1024 copy of the original input image. The UV map is then used for facial hash encoding, which has proven effective in enhancing the network's ability to capture both regional features and high-frequency details. The input image, along with the UV map, Normal map, and features produced by the hash-encoding, are concatenated and passed into the Global Distribution Extractor. This module outputs 1024×1024 Albedo, Micronormal, and Specular maps, which serve as guides for the subsequent stage, maintaining global consistency when predicting details across different patches.

In Stage 2 (High-Frequency Detail Refinement), we splitting the original input into 8×8 patches, each with a resolution of 512×512. These patches are processed individually to predict detailed PBR texture patches. The patches are then reassembled and post-processed to generate the final output. The guide textures produced by Stage 1 offer a coarse prediction, enabling Stage 2 to focus on refining high-frequency texture details.

By fully utilizing UHR input images, our method generates 4K textures with rich high-definition details, greatly enhancing artists' productivity. Additionally, our approach is completely independent of geometric reconstruction, functioning as a plug-and-play module that can significantly improve texture quality in existing facial geometry reconstruction methods.

In summary, this article makes the following contribution:

- We propose a novel two-stage 4K facial PBR texture reconstruction pipeline that fully leverages 4096×4096 UHR inputs, ensuring consistency across local detail patches via global guidance. Additionally, the geometric normal prior is designed to alleviate the planar assumption for local detail prediction.
- We introduce a novel facial region encoding technique, which enhances the network's ability to adaptively learn regional texture distributions, leading to improved prediction of fine details.
- Experiments demonstrate that our method significantly enhances facial texture reconstruction quality. It operates independently of geometric reconstruction, allowing for easy integration with state-of-the-art facial geometry pipelines. Leveraging advancements in mobile camera technology, our approach opens the door to even higher-quality single-image facial reconstruction in the future.

2. Related Work

2.1. 3D Face Reconstruction

3D face reconstruction, which focuses on recovering 3D facial shapes, expressions, and textures from 2D images, has been extensively explored in the literature [58]. Although accurate geometry estimation can be achieved in controlled environments, such as using multiple cameras in a light stage setup [12], these methods are expensive and time consuming, restricting their use in broader applications. This motivated research on facial rendering asset recovery from a single image, which enables a wider variety of applications.

Monocular reconstruction is typically treated self-supervised by modeling parametric scenes that include geometry, lighting, reflectance, and camera parameters [12]. These approaches can be broadly categorized into those based on parametric geometry models such as 3D Morphable models [9] (3DMM) and those that attempt to recover unconstrained geometry [39, 44, 53], which go beyond 3DMM priors.

Within the 3DMM framework, methods based on linear 3DMMs [10, 11, 13, 15, 17, 33, 43, 47, 57] are restricted by the statistical priors of the model, limiting expressiveness in reconstruction. To overcome these limitations, learning non-linear substrates or texture decoders [8, 41, 49, 48, 57] significantly improves the expressiveness

of 3DMM. Moreover, in some recent work, the differentiable rendering pipeline [27] facilitates many reconstruction methods learning directly from images.

Recently, with the further development of deep learning, many works focusing on implicit face modeling have achieved promising results. Along with the advances in DDPM theory [21], approaches using diffusion-based neural renderers for rendering high-fidelity human faces have also emerged [30]. Notably, with the advent of 3D Gaussian Splatting (3DGS) [28], a variety of methods based on this approach have emerged [38, 52, 45]. In particular, [42] shows highly realistic submillimeter-level relighting results, which have significant potential for future face-to-face interactions with VR or MR. However, these methods typically require a large amount of multiview, well-calibrated datasets and do not directly generate explicit geometry and textures, which may be challenging to integrate into most current rendering pipelines or require additional adaptation efforts.

2.2. High Resolution Facial Texture Generation

High-fidelity asset details are critical for achieving realistic computer rendering results, especially in facial rendering. Humans are particularly sensitive to the appearance of face [35], making the acquisition of high-fidelity rendering assets important to enhancing perceptual realism. As mobile device cameras continue to improve, capturing high-definition (4K) portraits is becoming increasingly accessible. Therefore, leveraging these high-resolution inputs to generate ultra-high-resolution (UHR) facial textures becomes a valuable topic.

Supported by differentiable rendering techniques, some methods utilize self-supervised and weakly supervised learning to reconstruct high-fidelity facial textures [13, 17]. Other approaches are based on texture decoders [46, 18, 32, 48], employing StyleGAN2 [26]to generate high-resolution UV textures, followed by 3D face geometry matching algorithms (e.g., 3DMM base) to find the optimal latent code for reconstruction.

FitMe [32] and Relightify [37]leverage real-world albedo datasets from AvatarMe++ and employ diffusion models [21]and GAN tuning [40]to extract albedo textures from in-the-wild images. Similarly to our goal, UV-IDM [34] focuses on reconstructing facial textures in UV space, and using existing BFM-based 3D face reconstruction methods to get geometry. For input images, UV-IDM maps facial region to the UV space as a condition for a latent diffusion model to complete the texture, while their output resolution is limited to 256x256. Similar to UV-IDM, our tasks of high-quality UHR texture generation and geometry matching are decoupled. We chose HRN [33] as our geometry matching algorithm because it employs a coarse-to-fine multi-stage iterative strategy, which excels at capturing high-frequency geometric details of the face.

MoSAR [14] reconstructs relightable avatars using high-quality light stage data, supporting the generation of 4K textures. However, their original input and output texture resolutions are limited to 512x512, which does not fully utilize the information in 4K input images. Their results are upsampled to 4K using a super-resolution network (ESRGAN), which potentially leads to texture pattern distortion. Additionally, their method relies on expensive high-quality light-stage data.

Recently, generative high-resolution avatars have become a prominent research topic, such as UltraAvatar [56]and DreamFace [55], which support both text and images as input to generate geometry and textures. In particular, DreamFace supports 4K input images and produces up to 4K high-fidelity albedo, micronormal, and specular textures. DreamFace utilizes a two-stage framework, leveraging a diffusion model to learn priors from a large-scale, high-quality dataset. This approach enables the generation of textures that not only ensure overall consistency with the input but also effectively preserve fine details. However, similar to MoSAR, DreamFace's pipeline does not generate 4K textures in an end-to-end manner, but instead by applying super-resolution (Real-ESRGAN [50]) on 512×512 raw output. In contrast, our method adopts a divide-and-conquer approach, directly reconstructing textures from high-resolution inputs in an end-to-end fashion. This strategy maximizes the retention of high-frequency details from the input without the need for super-resolution, resulting in superior preservation of fine details. Our method demonstrates better performance in maintaining subtle features, and we present a detailed comparison with DreamFace's results in our paper.

3. Methods

In this section, we introduce our two-stage approach to reconstruct ultra-high resolution PBR facial textures. As illustrated in Figure 1, our method has two main stages: in **Stage 1**, the target is to obtain the global texture distributions as guidance for next stage. In **Stage 2**, original input is split into 8×8 patches, each with a resolution of 512×512, which are processed individually to predict highly detailed PBR texture patches. These patches are then reassembled and post-processed to produce the final high-resolution output.

3.1. Preliminaries

Facial UV Texture Space. Facial texture distribution is highly region-specific [7], making an accurate positional encoding essential for the network to learn these patterns effectively. Instead of using facial segmentation masks [56] or landmarks [29] , we leverage UV coordinates for more

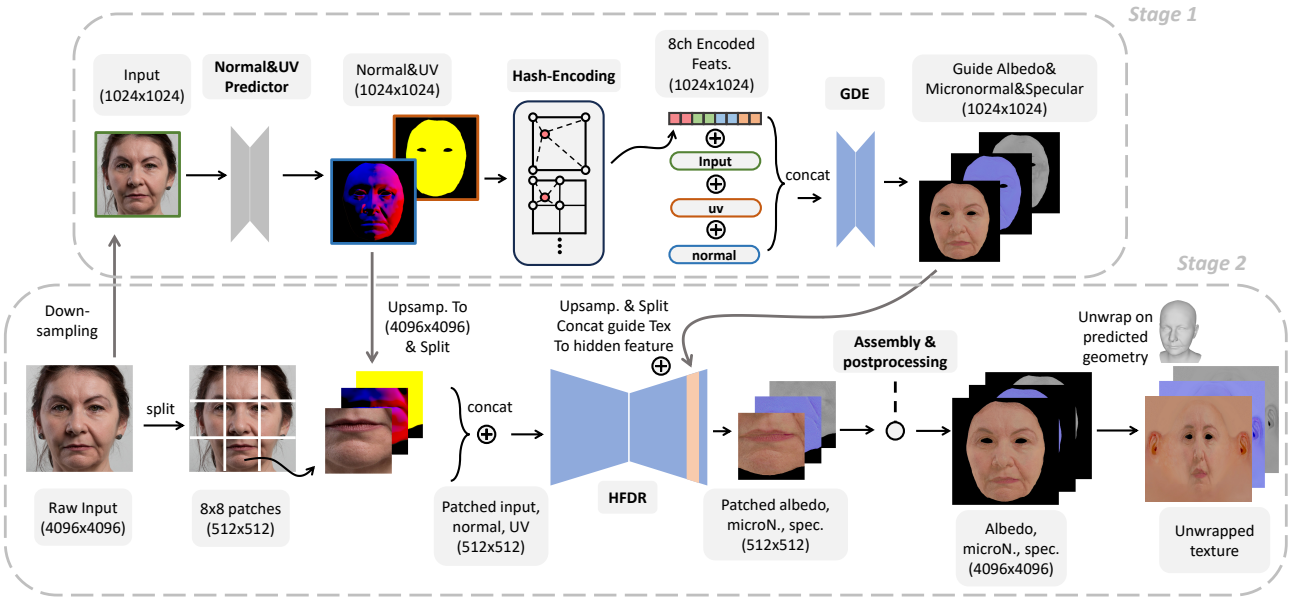

Figure 1. Our method's pipeline consists of two stages. In **Stage 1**, the input is downsampled to 1024x1024 and fed into Normal&UV predictor network to generate facial Normal and UV maps, with the UV map used for hash encoding of the facial region. 8 channels encoded features, combined with the Input, Normal, and UV maps, which are all in 1024*1024, are then processed by the Global Distribution Extractor (GDE) to produce 1024x1024 guide textures. In **Stage 2**, the input is split into 8x8 patches of 512x512. Each patch is concatenated with the corresponding region's cropped and upsampled Normal and UV maps from Stage 1, then passed through the High-Fidelity Detail Refiner (HFDR), producing 512x512 Albedo, Micronormal(microN.), and Specular(spec.) outputs. These patches are then assembled, post-processed, and mapped to the given geometry's UV space to produce the final texture.

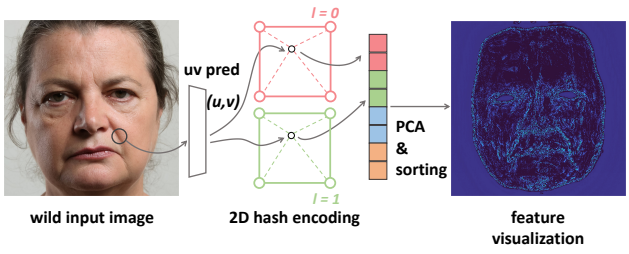

Figure 2. Visualization of the output features of the hash-encoding module. PCA is performed on the output 8-channel features of the hash-encoding module, and the three channels with the highest weights are visualized.

precise localization of facial regions. Specifically, we train a UNet (Normal&UV predictor) to map input RGB images to both UV and Normal maps, formulated as:

$$\mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H\times W\times 5}$$

Here, $\mathbb{R}^{H\times W\times 3}$ represents the RGB input, and $\mathbb{R}^{H\times W\times 5}$ contains the unified UV coordinates (first two channels) and normalized normal maps (last three channels). Normal maps are normalized as:

$$n = \frac{n_{\text{raw}} + 1}{2}$$

Data from [1] and rendered via Blender Cycles [2] are used for training Normal&UV predictor.

Geometry and Texture Mapping. We use HRN [33] as the geometric basis to capture detailed shapes. The geometry, $G(\beta, V_\gamma, M_\sigma)$, is formed using blendshape coefficients β, deformation map V_γ, and displacement map M_σ extracted from input. NvDiffrast [31] is then used to map our output textures from image space to UV space, optimizing by minimizing the difference between the rendered geometry and the input image.

$$\mathcal{L}_{\text{texture}} = \sum_{i,j} \|T_{\text{image}}(i,j) - \mathcal{R}(G(\beta, V_\gamma, M_\sigma), T_{\text{UV}})(i,j)\|_2^2$$

where $\mathcal{R}(G(.), T_{\text{UV}})$ represents the rasterized geometry with the UV texture, and T_{image} is the texture in image space.

3.2. Geometry Normal Prior

In the texture asset reconstruction task, our goal is to reconstruct Albedo, Micronormal, and Specular maps, which are wrapped on the geometric surface. Directly reconstructing local detail from the image space input to the texture space necessitates the assumption that the object itself is planar or near-planar, as described in [20]. However, the lack of prior knowledge about global geometry normals can potentially negatively affect the prediction of the lo-

cal Micronormal, especially for Micronormal prediction on small patches due to a limited receptive field. Without a global normal guidance, the network may incorrectly fuse the global geometry normal with the local Micronormal.

Therefore, we additionally train a **Normal&UV predictor** (module in **Stage 1**) to predict the UV and geometry normal distribution for given input, and utilize the obtained UV and normal predictions as auxiliary inputs for GDE and HFDR. With the assistance of global normal, the network can effectively focus on learning the Micronormal distribution in the texture space, which can be expressed as follows:

$$||f(I_{\text{raw}}) - n_{\text{micro}}||^2 \geq ||f(n_{\text{global}}, I_{\text{raw}}) - n_{\text{micro}}||^2 - \Delta$$

where n_{global} is normal aquired by Normal&UV predictor, and Δ is a constant that indicates the reduction in learning effort for f to converge to n_{micro} when relying solely on the raw image I_{raw}. This demonstrates that the predicted global normal facilitates the model to learn the subtle and high frequency in Micronormal distribution. Meanwhile, additional geometry normal prior also alleviating the pressure of model's relighting process when predicting Albedo and Specular maps. In the comparison shown in Figure 7, it can be observed that our method is less affected by self-shadowing caused by geometry, resulting in cleaner and more refined albedo outputs.

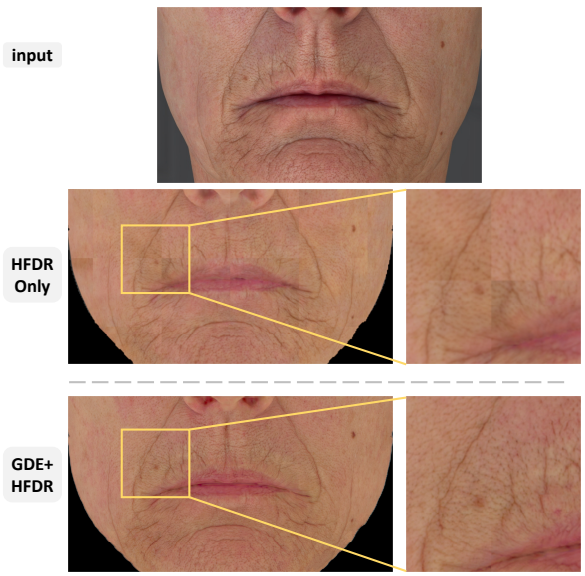

Figure 3. Directly stitching the texture maps from HFDR results in noticeable inconsistencies across patches. In this case, the input image (4096×4096) is divided into 64 non-overlapping 512×512 patches, with each processed independently by HFDR.

3.3. Global Distribution Extraction (Stage 1)

Given the raw input image I_{raw}, in Stage 1, the goal is to obtain the global texture distribution, which serves as guidance for Stage 2. The guide textures generated in Stage 1 provide coarse guidance, allowing Stage 2 to focus the network's capacity on refining high-frequency texture details, and more importantly, maintains global consistency when predicting details across different patches. In this stage, the involved modules include the Normal&UV Predictor (as discussed in Sec 3.2), the Facial Hash Encoding module, and the Global Distribution Extractor.

Facial Hash Encoding.

For facial reconstruction tasks, an effective spatial encoding method is crucial due to the strong correlation between different facial regions and their corresponding texture distributions. Proper encoding enhances the network's ability to efficiently learn the distribution of facial details, particularly in areas like the lips and eye corners, where the texture and color typically differ from other parts of the face.[22]

Previous works [56] typically employ a facial segmentation network to segment the face and use the predicted mask as part of the auxiliary input. Other works leverage facial landmarks [29] for the same purpose. Although these approach is straightforward and easy to implement, leveraging priors learned from large datasets of facial images, the predicted results only provide a very coarse spatial encoding. For example, a part of mask encodes an entire region with a single ID, while landmarks only provide relative positions for eyes, nose, etc., lacking finer granularity, which potentially reduces the network's ability to adaptively capture subtle details. [23]

To further enhance the Global Distribution Extractor (GDE) 's network's ability to adaptively learn the subtle differences in texture distributions across various facial regions, we apply hash encoding [36] to the UV coordinate map of the face predicted by Normal&UV predictor in this Stage. The Normal&UV predictor generates 2D coordinates of the face within a unified UV space, represented as $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2$. These coordinates serve as inputs to the hash encoding, allowing the network to better capture fine-grained texture details across the face.

The hash encoding consists of multi-layers of multiresolution hash table of trainable textures whose values are optimized through stochastic gradient descent. This compact representation helps the network capture complex spatial patterns and fine-grained details in high-resolution data. Specifically, we define the number of learnable hash texture levels as $L = 4$, with $n_F = 2$ features per texture level, yielding a final output of $C = L \times n_F = 8$ channels after hash encoding During training, we observed that an 8-channel hash encoding was sufficient, as increasing the number of channels did not yield significant improvements and instead led to slower training speeds.

Our hash encoding module is jointly trained with the Global Distribution Extractor (GDE) network, allowing the

gradients from GDE to optimize the features in hash encoding module, which can be expressed as:

$$\mathbf{H}^{(k+1)}_{\text{optimized}} = \mathbf{H}^{(k)}_{\text{optimized}} + \nabla \mathcal{L}_{\text{GDE}}(\mathbf{H}^{(k)}_{\text{optimized}})$$

where $\mathbf{H}$ denotes features in hash textures in hash encoding module, and $\nabla \mathcal{L}_{\text{GDE}}(\mathbf{H})$ denotes the gradient of the loss function propagated to the hash-encoded features, which enables the model to learn the prior closely related to the UV distribution of the facial geometry.

By performing principal component analysis (PCA) on the 8-channel features output by the hash-encoding module, we extract the 3 most weighted features and make visualization, please refer to Figure 2. It shows that hash-encoding can learn the feature distribution of different regions of the face. In the Ablation Study, we quantitatively compare the prediction quality of the GDE after removing hash encoding module.

Global Distribution Extractor.

For ultra-high-resolution inputs, such as 4096x4096 textures, directly feeding them into the network is impractical due to the memory size restriction on current GPUs, making training inpractical. As a result, it is a common practice to divide the image into patches for localized predictions. However, limiting the network's input to individual patches results in a significant loss of global contextual information, thereby constraining the receptive field and making it insufficient for capturing broader texture distributions. The absence of constraints on the texture value distribution from neighboring patches causes the average distribution of each patch to fluctuate independently. This lack of coherence often leads to noticeable blocky artifacts and mosaic effects between adjacent patches. Such issues typically arise when the network relies solely on local information, without incorporating global context (see Figure. 3).

To enhance the capability of the High-Frequency Detail Refiner (HFDR) module in capturing long-range structured dependencies, we introduce additional module, Global Distribution Extractor (GDE), which is designed to capture global long-range contextual features. However, as noted in Sec.3.2, directly using RGB images as input for either the GDE or HFDR modules conflicts with the near-planar prior assumption. To address this, we leverage the outputs of the Normal&UV Predictor as auxiliary inputs, providing additional global geometric distrubution and ensuring that both modules can better capture the subtle high-frequency detail context of the face.

Our GDE module consists of a 12-layer UNet-like backbone architecture. The input is downsampled to a resolution of 1024×1024, then, the Normal&UV predictor is used to predict the corresponding UV and normal maps:

$$\{UV_{\text{pred}}, N_{\text{pred}}\} = F_{\text{uv-normal}}(I_{\text{downsampled}})$$

The predicted UV distribution is used in **Facial Hash Encoding** to generate an 8-channel feature map. This, along with the downsampled input, predicted UV, and normal maps, are concatenated as input:

$$\mathbf{X}_{\text{GDE}} = I_{\text{downsampled}} \oplus UV_{\text{pred}} \oplus N_{\text{pred}} \oplus F_{\text{encoded}}$$

This concatenated input is then fed into the GDE network, producing a 7-channel output:

$$\mathbf{G}_{\text{output}} = \text{GDE}(\mathbf{X}_{\text{GDE}})$$

where $\mathbf{G}_{\text{output}} = [\mathbf{G}_{\text{g\_albedo}}, \mathbf{G}_{\text{g\_micronorm}}, \mathbf{G}_{\text{g\_spec}}]$, with $\mathbf{G}_{\text{g\_albedo}} \in \mathbb{R}^3$, $\mathbf{G}_{\text{g\_micronorm}} \in \mathbb{R}^3$, and $\mathbf{G}_{\text{g\_spec}} \in \mathbb{R}^1$. Here, $g\_$ denotes the abbreviation for 'guidance', which will serve as the guiding features for HFDR in subsequent stage.

The optimization objective of the GDE network is a weighted combination of **L1 Loss** and **Laplacian Loss**. We observed that Laplacian loss enhances the network's ability to perceive and learn high-frequency textures more efficiently. This is especially crucial for capturing fine details such as skin pores, wrinkles, and other subtle features, which are likely missed by standard L1 loss alone. The total loss is formulated as follows:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{L1}} + \lambda_2 \mathcal{L}_{\text{Laplace}},$$

where $\lambda_1 = \lambda_2 = 1$. The Laplacian loss, $\mathcal{L}_{\text{Laplace}}$, is computed using a convolution with a Laplacian kernel to perceive sharp spatial discontinuities, encouraging learning in high-frequency distributions. The kernel $\mathbf{K}$ is defined as:

$$\mathbf{K} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Given above kernel, the Laplacian loss is implemented as follows:

$$\mathcal{L}_{\text{Laplace(K)}} = \mathbb{E}\left[|F_{\text{Laplace}}(\mathbf{P}) - F_{\text{Laplace(K)}}(\mathbf{T})|\right],$$

where F_{Laplace} represents the 2D convolution operation using the Laplacian kernel $\mathbf{K}$, $\mathbf{P}$ is the predicted texture, and $\mathbf{T}$ is the target texture. Both loss is computed over all channels.

Notably, GDE module is trained independently rather than in a cascaded manner with the HFDR. We observed that joint training in a cascaded setup results in the GDE module being continuously influenced by the gradients from the HFDR's patch receptive field. This interference hinders the GDE's ability to focus on global context, ultimately decrease its capacity to guide the HFDR effectively.

3.4. High-Frequency Detail Refinement (Stage 2)

To obtain ultra-high-resolution (UHR) textures with rich local details, we need to predict highly detailed PBR textures at the original resolution level, therefore maximize the

utilization of the detail information within the UHR inputs. Patches are processed individually in this stage, taking advantage of the global guidance of Stage 1 to maintain overall consistency.

At this stage, the raw input image is cropped into 8×8 patches, each of size (512x512). The guidance texture from the GDE is split into 64 patches (128x128), which are then upsampled to match the patch resolution (512x512). These upsampled patches are concatenated with the features of last hidden layer of HFDR, guiding its final output. The 64 resulting patches are subsequently assembled to form a complete 4096×4096 output. To further smooth the seams between patches, we apply a Gaussian filter to convolve the adjacent regions between the patches. Finally, the predicted 4096×4096 PBR textures from the original image are unwrapped into the UV texture space of the given geometry using nvdiffrast [31], and the blank areas are inpainted accordingly.

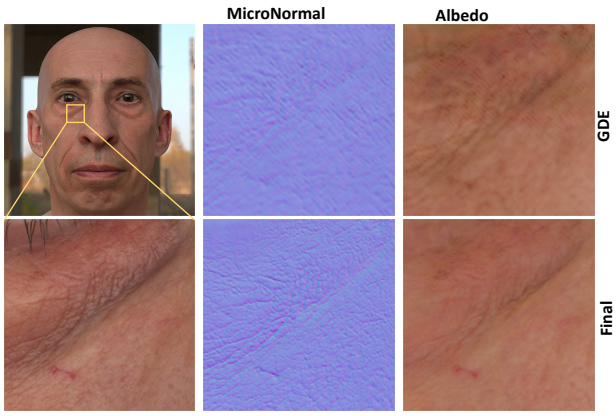

Figure 4. Illustration of the output of GDE, as compared with the final prediction after HFDR (Final). Closeups are shown in this case. Note that fine details in the input are missing in the output of the GDE of Stage 1.

High Frequency Detail Refiner.

Given the guidance texture from the GDE module, the HFDR module can more efficiently focus on capturing high-granularity, high-frequency local features within patches. Similar to GDE, HFDR consists of a UNet-like backbone. The input consists of the original resolution image, which is cropped into 512×512 patches, and concatenated with the upsampled predicted UV and normal patch in the same position. The guidance from GDE is concatenated with the 17th hidden layer features of HFDR, significantly reducing the burden of predicting the overall texture distribution for HFDR. This also introduces long-range contextual information into HFDR, mitigating discontinuities between the distributions of adjacent patches. The loss function of HFDR is similar to that of GDE, consisting of a L_1 loss and a Laplace loss, but the weight ratio between them is set to 1:2, as shown in the equation below:

$$\mathcal{L}_{HFDR} = \mathcal{L}_{L_1} + 2 \cdot \mathcal{L}_{Laplace}$$

This allows HFDR to focus more on learning high-frequency detail patterns.

Figure 5. Comparison before and after seam post-processing. After postprossing, the slight discontinuities between patches become smoother.

Assembling and Postprossing.

Given 64 patches $P_{i,j}$ of size 512×512, where $i, j \in \{0, 1, \ldots, 7\}$, we assemble them according to their original positions to form an initial output of size 4096×4096:

$$\text{Output}(x, y) = P_{\lfloor \frac{x}{512} \rfloor, \lfloor \frac{y}{512} \rfloor}(x \bmod 512, y \bmod 512)$$

With the assistance of the global guidance from the GDE module, the discontinuities between different patches in the output are significantly reduced. However, due to resolution limitations, GDE cannot capture high-frequency fine-grained details. As a result, there may still be noticeable seams where the patterns do not match perfectly at the patch boundaries. To address this issue, we apply anisotropic Gaussian filtering to region of the horizonal and vertical seam, achieving both seam smoothing and the preservation of certain local high-frequency patterns. Specifically, we process the regions adjacent to the seams with a padding size of 8 pixels. For horizontal seams, we apply a Gaussian kernel defined by $x = 2$ and $y = 16$; for vertical seams, we use a convolutional kernel with $x = 16$ and $y = 2$. Both convolutional kernels have a standard deviation of $\sigma = 5$. The results demonstrate that this approach effectively reduces the impact of seam artifacts, as shown in Figure 5. After obtaining the processed output, it needs to be transformed from the image space to the corresponding UV space of the specific geometry. As explained in the Preliminaries 3.1 , we reconstruct the geometry using the method [33], and utilize NvDiffrast [31] to optimize the texture in the UV space. This is achieved by minimizing the loss between the predicted image-space texture and the projected (or rasterized) texture in the UV-space, effectively "unwrapping" the predicted image-space texture onto corresponding UV coordinates. To capture the predicted output with the highest fidelity as possible, we employ a cascaded texture

optimization strategy, with the highest texture resolution set to 8192×8192, then downsampling it to 4096×4096. Finally, we interpolate the resulting UV space texture with a predefined texture. The interpolation weights are determined based on the average color of the facial region.

4. Implementation

4.1. Dataset

Our dataset is sourced from the online asset shop [1], which includes a collection of 82 head assets with albedo, micronormal, and specular texture maps up to 8K. The training and testing sets are split in a 5:1 ratio, with the testing set used to evaluate texture reconstruction quality. For generating synthetic data in 4096×4096, we select 20 different environment maps and rotation settings to light the heads and introduce perturbations to the camera angles to enhance robustness. We render the assets using Blender's Cycles [2] renderer, with the sample count set to 1024 spp. The material is defined using Blender's built-in Principled BSDF shader, specifying the diffusecolor, normal, and specular texture slots. The renderer outputs 4096x4096 radiance images of the front-facing head, along with corresponding Albedo, Micronormal, and Specular maps. The wild images used for testing come from the web, and since this dataset lacks ground-truth textures, it is used to assess the network's generalization capabilities on wild images.

4.2. Training

The modules requiring training consist of the Normal&UV predictor, the hash-encoding module, the Global Distribution Extractor(GDE) module, and the High Frequency Detail Refiner(HFDR) module. All networks are optimized using the Adam [16] optimizer, with an initial learning rate of 0.001, and a learning rate reduction strategy with the ReduceLROnPlateau [5] scheduler, where patience is set to 5. For Normal&UV predictor, we observe convergence after training for 56 epochs. For GDE and HFDR modules, we performed end-to-end training for the hash-encoding module and GDE modules first, then froze their weights and trained the HFDR module separately. The GDE and hash-encoding modules were trained for 600 epochs, taking approximately 2 days on an Nvidia RTX 4090; the HFDR module was trained for 40 epochs, taking about 5 days on the same device setting.

5. Experiments

To verify the effectiveness and efficiency of the proposed method, we conduct several experiments on both synthetic data and high resolution wild image data. Note that these test cases are never involved in training. For both types of data, we implement a diverse collection across various races, genders, and age groups, to ensure comprehensive representativeness. For synthetic data, we ensure that the facial area occupies as large a portion of the image as possible during data generation, in order to fully leverage the high resolution of the input images. For wild image data, although there may be fluctuations in resolution depending on various data source, we guarantee the resolution of the primary facial area is more than 3000 pixels. Both types of data will be crop according to the facial area and resizing to the input size of 4096x4096. We compare our method against several baselines in image-to-avatar generation with physically based rendering (PBR) textures (DreamFace [55] and NextFace [15]).

Experiment Setup. For all test methods, including ours, we retain the original resolution of the generated geometry and texture assets, importing them into a unified Blender rendering pipeline. We do not enforce a uniform UV mapping across different methods; instead, each method's native UV mapping is preserved. For NextFace, we use the upsampling module provided in their official implementation to get 2K texture. For DreamFace, we utilize the 'detail shot' mode available on their official website, generating assets at the highest resolution (4K) permitted by the platform to fully leverage DreamFace's capabilities, and the downloaded assets are then imported via DreamFace's Blender importer [3], where we retain the albedo, normal map, and specular map settings, with all other parameters kept consistent across methods for a fair comparison. For NextFace and our method, we use the Principled BSDF [4] material node, setting the albedo, normal, and specular texture inputs, and assigning any missing values to standardized defaults. For the same test samples, we use identical environment maps and viewing angles to ensure consistent lighting. We set the sample count to 1024 samples per pixel (spp).

Method	Albedo		MicroNormal		Specular	
	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
DreamFace	0.852	0.173	0.672	0.261	0.713	0.243
NextFace	0.666	0.371	-	-	0.604	0.481
Ours	**0.903**	**0.066**	**0.732**	**0.141**	**0.764**	**0.088**

Table 1. Results of quantitative comparison of our method with DreamFace and NextFace.

5.1. Performance Analysis on Synthetic Data

All tests were performed on a PC with an Intel i9-14900KF CPU, 32 GB RAM, and a NVIDIA GeForce RTX 4090 GPU. For a single test sample, our method completes the entire reconstruction pipeline in under 40 seconds, faster than DreamFace, which takes approximately 2 minutes, and NextFace, which exceeds 5 minutes. Notably, other methods require downsampling the input image to meet input size constraints, whereas our method can directly process 4K inputs at full resolution, while maintaining high efficiency and reasonable GPU memory usage.

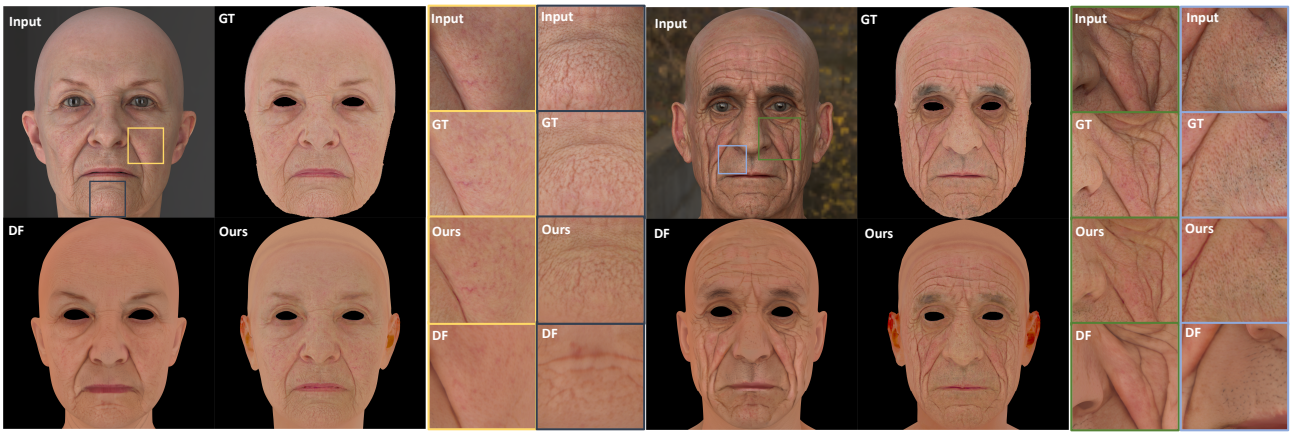

Figure 6. Quality comparison of reconstruction between ours method and DreamFace (DF in short) on synthetic data, test samples used are excluded from any training. Our method is able to directly reconstruct textures from 4096x4096 input, without supersampling network, which allows us to capture extremely high-frequency and subtle details from input, such as facial capillaries and pores, in an end-to-end manner. Compared to DreamFace, our method does not tend to bake shadows into the albedo, resulting in cleaner and more accurate texture reconstruction.

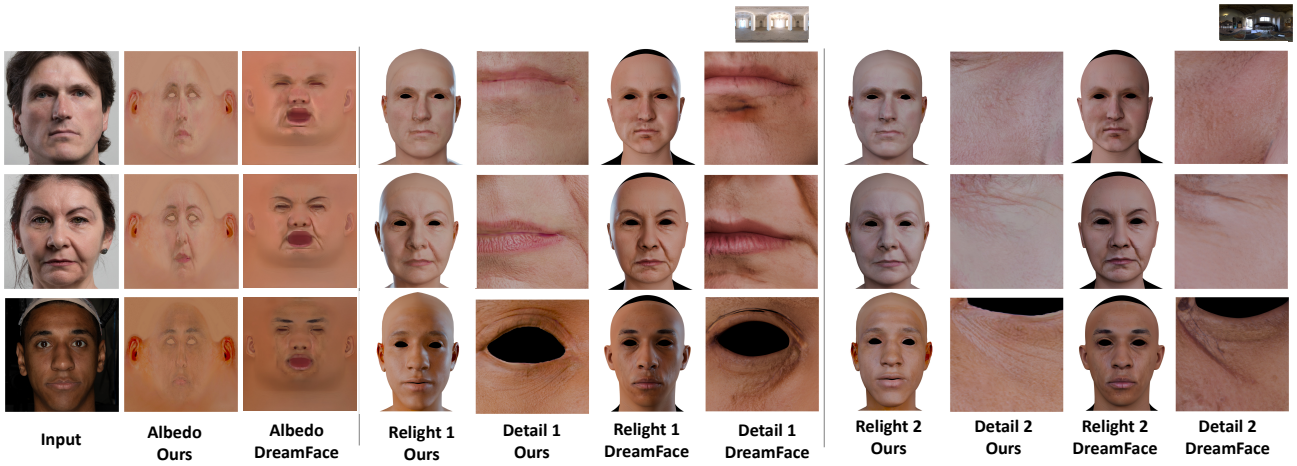

Figure 7. Comparison of texture reconstruction quality between Ours and DreamFace methods. Our pipeline shows better quality in terms of texture clarity and detail. As shown, DreamFace tends to bake shadows into the texture (row 1 and row 2), whereas our end-to-end pipeline precisely reconstructs high-frequency details, such as skin's micro-topography , which are difficult to achieve by methods like supersampling to 4K. Please zoom in to see the details.

To assess the reconstruction capabilities of the compared methods for ultra-high-resolution inputs, we utilized synthetic data containing Albedo, MicroNormal, and Specular ground truth (GT) to perform a quantitative analysis. For all methods, the evaluation region was defined as the overlapping area of the frontal projections from the aligned geometries, with the corresponding results shown in Table 1. The visual comparison of reconstructed albedo between ours and DreamFace is presented in Figure 6, which includes both the overall head result and zoomed-in views of the albedo maps. In the cases illustrated in Figure 6, our method outperforms DreamFace in terms of texture clarity and fidelity, while the maps generated by DreamFace exhibit notable deficiencies in high-frequency detail. Specifically, in the left case, our approach successfully reconstructs subtle details (e.g. facial blood vessels and subtle chin wrinkles), while DreamFace tends to produce a blurrier and more fused pattern, even at 4K resolution. We attribute this discrepancy to the loss of significant high-frequency details resulting from the downsampling process employed by DreamFace. Although DreamFace's robust augmentation module leverages priors learned from extensive high-resolution datasets and upsamples its original output from 512×512 to 4096×4096, the absence of pixel-to-pixel correspondence at the same resolution hampers its ability to accurately restore high-frequency details from the input.

Instead, it generates unbounded content based on a relatively lower-resolution original output. Furthermore, our predicted albedo are generally closer to the ground truth (GT) in terms of base color. Additional comparisons can be found in Figure 9.

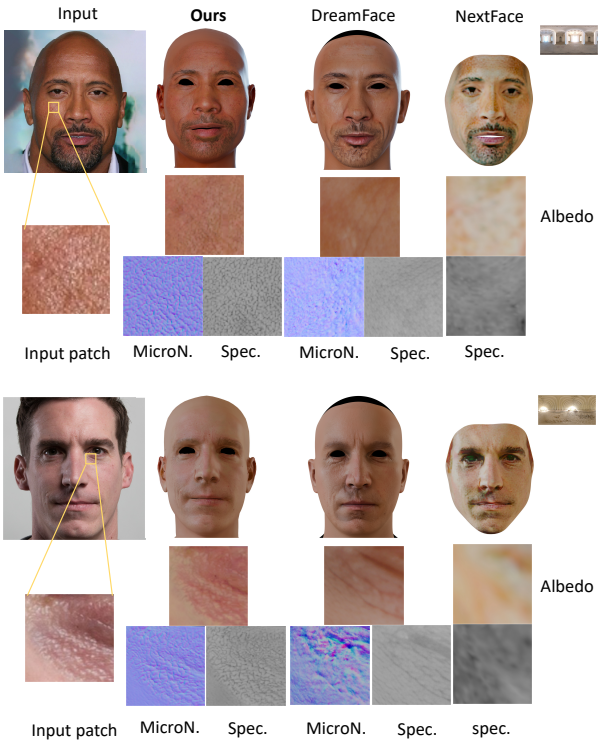

Figure 8. Comparison of head relighting results and zoomed-in views of the reconstructed textures demonstrates that our method produces cleaner, more accurately restored high-frequency details.

5.2. Wild Image Reconstruction

We conducted extensive tests on high-resolution wild input images across a wide range of ages, genders, and ethnicities to evaluate the reconstruction capabilities of different methods. The performance of these methods is illustrated from three perspectives: 1) the clarity and detail of rendered results, as shown in Figure 7; 2) a zoomed-in comparison of texture details, presented in Figure 8; and 3) a visualization of global comparison of the radiance and albedo maps generated from different inputs, as shown in Figure 9.

Among the methods compared, we observed that DreamFace performs well in maintaining consistency between the frontal renderings and the input, particularly in areas like the major facial folds on the sides of the nose and the shadows under the lower lip. However, as shown in Figure 7, DreamFace's strong emphasis on input consistency results in shadows being baked into the generated albedos, which is shown in the first and second rows of Detail 1 in Figure 7, where after relighting, the lower lip region still exhibits overly deep artifacts. In contrast, our method outperforms in both clarity and high-frequency detail, with significantly fewer baked-in shadows and artifacts. The incorrect self-occlusion baking is even more evident in the comparison around the eyes in the third row of Figure 7. DreamFace tends to bake self-shadows from fine wrinkles around the eyes directly into the albedo, aiming for input consistency. This results in unnatural dark streaks in the albedo, which are unlikely to appear in real skin, leading to color inconsistencies in albedo. We hypothesize that this phenomenon is due to DreamFace struggles to differentiate between local shadows caused by geometric self-occlusion and inherent skin properties in the input. Leveraging our end-to-end reconstruction strategy and facial hash encoding, our method successfully mitigates the influence of self-shadowing from geometric details while preserving high-frequency texture details, resulting in albedo maps with a more consistent and accurate color distribution.

In the comparison shown in Figure 8, we evaluate the quality of ultra-high-resolution texture reconstruction between DreamFace, NextFace, and our method. Due to image size limitations, we present the texture details using a zoom-in approach. As illustrated in Figure 8 ,our method, which employs a patch-based strategy and leverages hash encoding to capture local texture distributions via self-attention, enables pixel-to-pixel reconstruction of the input image at its original resolution. This results in significantly better reconstruction of albedo, micronormal, and specular high-frequency details, which are generally more robust and faithful compared to those generated by other methods. For DreamFace, since its textures are obtained via upsampling on 512x512 original output, it blends priors learned from other distributions to compensate for the missing details, which leads to unbounded outputs and a tendency to generate disorganized patterns for high-frequency reflectance, which is particularly noticeable in the Micronormal map. As for NextFace, its reconstruction strategy heavily relies on differential path tracing, a method that highly depends on accurate light transport modeling [54]. The quality of its texture reconstruction is therefore strongly correlated with the correctness of environmental lighting estimation. However, accurately estimating environmental lighting in monocular image reconstruction tasks is highly challenging, which cause some features to be incorrectly baked into the predicted environment map, resulting in degraded albedo quality. Consequently, shadows and highlights are incorrectly merged into the albedo, leading to unrealistic skin tones, as seen in Figure 9.

5.3. Ablation Study

hash encoding ablation.

Figure 2 shows the principal component analysis (PCA) performed on the features generated by hash encoding and

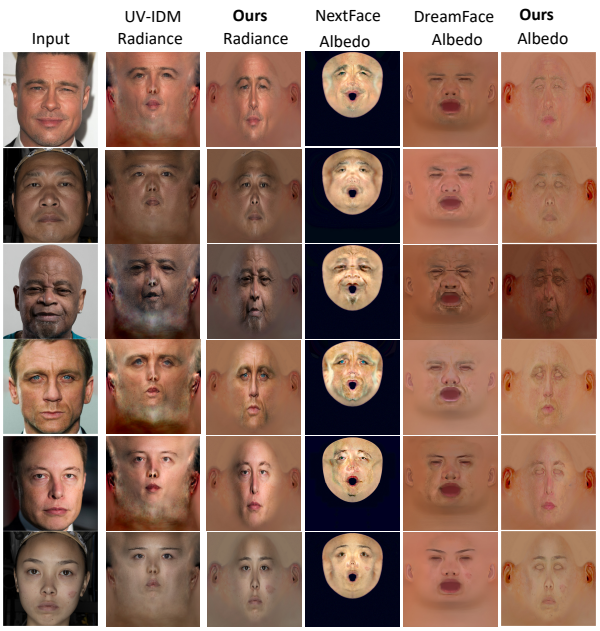

Figure 9. Examples of generated radiance or albedo textures from in-the-wild images using UV-IDM, DreamFace, NextFace and our method. The skin tones reconstructed by our method more closely match the input compared to other approaches.

Metrics	Albedo	MicroN	Specular
SSIM w/ Hash	**0.55**	0.42	**0.21**
SSIM w/o Hash	0.52	**0.43**	0.21
PSNR w/ Hash	**15.10**	**6.11**	**12.62**
PSNR w/o Hash	13.54	6.08	12.51

Table 2. SSIM and PSNR quality comparison for GDE's output feature between variants with and without Hash Encoding. It can be observed that the encoding module helps GDE learn the distribution of albedo in different regions.

visualized the three most significant channels. The visualization shows that applying UV hash encoding to the face allows the model to adaptively learn the distribution of distinct features across different facial regions from dataset, which benefits the predictions made by the GDE module in a manner similar to self-attention. To further validate the impact of the hash encoding module on the ability of the GDE network to learn global features, we conducted an ablation study. In Figure 10, we compare the quality of the output features with and without the hash encoding module. As shown, adding the hash encoding module enhances the GDE's ability to learn region-dependent distributions. Notice the lip area: while the variant without hash encoding captures the overall albedo distribution well, it struggles to distinguish the color differences in the lips. We tested both variants on the test set, evaluating texture reconstruction quality using SSIM, LPIPS and PSNR metrics, and found an improvement after adding the hash-encoding module, as shown in Table 2.

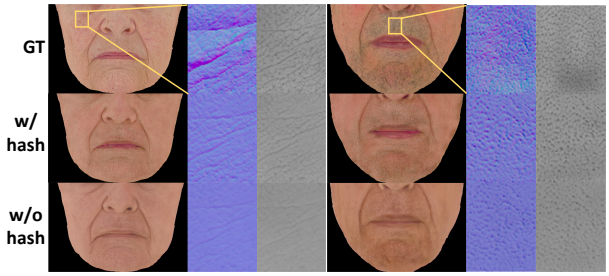

Figure 10. Comparison for GDE variant w/ and w/o hash encoding module. from up to bottom: GT, w/ hash encoding, w/o hash encoding. Adding hash encoding module can boost the GDE's ability on region dependent features, see lips' color. Meanwhile, hash encoding can enhance GDE's high-frequency feature predict quality.

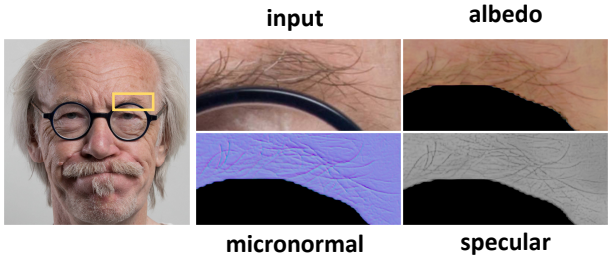

Figure 11. Failure case. Our approch does not explicitly model external occluders, such as glasses or hairs.

6. Limitation and Future Work

Our method does not explicitly model external occluders, such as glasses or hair, which can sometimes be baked into the Albedo map, and affect the quality of the Micronormal and Specular maps (see Figure 11). Additionally, due to dataset distribution biases, the reconstruction quality for certain ethnic groups may degrade slightly due to the scarcity of training data. A promising future direction could involve leveraging widely-used datasets like FFHQ [25] to learn coarse priors, followed by fine-tuning with high-resolution datasets to capture high-frequency details. This hybrid training approach could potentially enhance the generalizability of the model, and we leave this as an interesting direction for future exploration.

7. Conclusion

In this paper, we have proposed a novel approach for reconstructing 4K Albedo, Micronormal, and Specular textures directly from ultra-high-resolution portrait images. Our approach addresses the limitations of many previous

facial texture reconstruction methods, which couldn't handle ultra-high-resolution (UHR) input end-to-end, and results in substantial information loss by input downsampling. Our two-stage approach fully leverages 4K UHR inputs and captures high-frequency detail textures while maintaining global consistency. In addition, we utilize a novel facial hash encoding method to enhancing the network's ability to capture both regional features and high-frequency details, and use global normal prior to address the near-planar assumption. Experiment results have demonstrated that our method substantially enhances physically-based rendering (PBR) texture reconstruction quality. Its independence from geometric reconstruction allows for easy integration into state-of-the-art facial geometry construction methods. Furthermore, by leveraging advances in mobile camera technology, our approach sets a new standard for single image facial reconstruction, opening the door to even higher-quality results in future applications.

Acknowledgement

Special thanks to HongTao Sheng for participating in idea discussions, and to Yun Lin and Ziyi Xu for their contributions to user testing. The work was partially supported by National Key R&D Program of China (No. 2023YFF0905102), Key R&D Program of Zhejiang Province (No. 2023C01039).

References

[1] 3dscan store. https://www.3dscanstore.com/. 4, 8

[2] Blender cycles. https://docs.blender.org/manual/en/latest/render/cycles/index.html. 4, 8

[3] Chatavatar import tool. https://deemos.gumroad.com/l/ChatAvatarImportTool. 8

[4] Principled bsdf, blender. https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/principled.html. 8

[5] Reducelronplateau. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html. 8

[6] The smartphone camera evolution. https://autopix.no. 1

[7] T. J. Andrews, J. Davies-Thompson, A. Kingstone, and A. W. Young. Internal and external features of the face are represented holistically in face-selective regions of visual cortex. *Journal of Neuroscience*, 30(9):3544–3552, 2010. 3

[8] T. Bagautdinov, C. Wu, J. Saragih, P. Fua, and Y. Sheikh. Modeling facial geometry using compositional vaes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018. 2

[9] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 157–164. 2023. 2

[10] Z. Chai, T. Zhang, T. He, X. Tan, T. Baltrusaitis, H. Wu, R. Li, S. Zhao, C. Yuan, and J. Bian. Hiface: High-fidelity 3d face reconstruction by learning static and dynamic details. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9087–9098, 2023. 2

[11] R. Daněček, M. J. Black, and T. Bolkart. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20322, 2022. 2

[12] P. Debevec. The light stages and their applications to photoreal digital actors. *SIGGRAPH Asia*, 2(4):1–6, 2012. 2

[13] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2, 3

[14] A. Dib, L. G. Hafemann, E. Got, T. Anderson, A. Fadaeinejad, R. M. Cruz, and M.-A. Carbonneau. Mosar: Monocular semi-supervised model for avatar reconstruction using differentiable shading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1770–1780, 2024. 1, 3

[15] A. Dib, C. Thebault, J. Ahn, P.-H. Gosselin, C. Theobalt, and L. Chevallier. Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12819–12829, 2021. 1, 2, 8

[16] P. K. Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014. 8

[17] Y. Feng, H. Feng, M. J. Black, and T. Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 2, 3

[18] B. Gecer, J. Deng, and S. Zafeiriou. Ostec: One-shot texture completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7628–7638, 2021. 3

[19] A. Gruber, E. Collins, A. Meka, F. Mueller, K. Sarkar, S. Orts-Escolano, L. Prasso, J. Busch, M. Gross, and T. Beeler. Gantlitz: Ultra high resolution generative model for multi-modal face textures. In *Computer Graphics Forum*, volume 43, page e15039. Wiley Online Library, 2024. 1

[20] J. Guo, S. Lai, Q. Tu, C. Tao, C. Zou, and Y. Guo. Ultra-high resolution svbrdf recovery from a single image. *ACM Transactions on Graphics*, 42(3):1–14, 2023. 2, 4

[21] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[22] T. Igarashi, K. Nishino, S. K. Nayar, et al. The appearance of human skin: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(1):1–95, 2007. 1, 5

[23] M. M. Kalayeh, B. Gong, and M. Shah. Improving facial attribute prediction using semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6942–6950, 2017. 5

[24] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 1

[25] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019. 11

[26] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3

[27] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon. Differentiable rendering: A survey. *arXiv preprint arXiv:2006.12057*, 2020. 3

[28] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[29] K. Khabarlak and L. Koriashkina. Fast facial landmark detection and applications: A survey. *Journal of Computer Science and Technology*, 22(1):e02, Apr. 2022. 3, 5

[30] T. Kirschstein, S. Giebenhain, and M. Nießner. Diffusionavatars: Deferred diffusion for high-fidelity 3d head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5481–5492, 2024. 3

[31] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 4, 7

[32] A. Lattas, S. Moschoglou, S. Ploumpis, B. Gecer, J. Deng, and S. Zafeiriou. Fitme: Deep photorealistic 3d morphable model avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8629–8640, 2023. 3

[33] B. Lei, J. Ren, M. Feng, M. Cui, and X. Xie. A hierarchical representation network for accurate and detailed face reconstruction from in-the-wild images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 394–403, 2023. 2, 3, 4, 7

[34] H. Li, Y. Feng, S. Xue, X. Liu, B. Zeng, S. Li, B. Liu, J. Liu, S. Han, and B. Zhang. Uv-idm: Identity-conditioned latent diffusion model for face uv-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10585–10595, 2024. 1, 3

[35] M. Mori, K. F. MacDorman, and N. Kageki. The uncanny valley [from the field]. *IEEE Robotics & automation magazine*, 19(2):98–100, 2012. 3

[36] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 5

[37] F. P. Papantoniou, A. Lattas, S. Moschoglou, and S. Zafeiriou. Relightify: Relightable 3d faces from a single image via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8806–8817, 2023. 3

[38] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 3

[39] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1259–1268, 2017. 2

[40] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 42(1):1–13, 2022. 3

[41] Z. Ruan, C. Zou, L. Wu, G. Wu, and L. Wang. Sadrnet: Self-aligned dual face regression networks for robust 3d dense face alignment and reconstruction. *IEEE Transactions on Image Processing*, 30:5793–5806, 2021. 2

[42] S. Saito, G. Schwartz, T. Simon, J. Li, and G. Nam. Relightable gaussian codec avatars. In *CVPR*, 2024. 3

[43] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772, 2019. 2

[44] M. Sela, E. Richardson, and R. Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 1576–1585, 2017. 2

[45] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024. 3

[46] R. Slossberg, I. Jubran, and R. Kimmel. Unsupervised high-fidelity facial texture generation and reconstruction. In *European Conference on Computer Vision*, pages 212–229. Springer, 2022. 3

[47] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1274–1283, 2017. 2

[48] L. Tran, F. Liu, and X. Liu. Towards high-fidelity nonlinear 3d face morphable model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1126–1135, 2019. 2, 3

[49] L. Tran and X. Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018. 2

[50] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 1, 3

[51] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 1

[52] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 1931–1941, 2024. 3

[53] X. Zeng, X. Peng, and Y. Qiao. Df2net: A dense-fine-finer network for detailed 3d face reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2315–2324, 2019. 2

[54] C. Zhang. *Path-space differentiable rendering*. University of California, Irvine, 2022. 10

[55] L. Zhang, Q. Qiu, H. Lin, Q. Zhang, C. Shi, W. Yang, Y. Shi, S. Yang, L. Xu, and J. Yu. Dreamface: Progressive generation of animatable 3d faces under text guidance. *arXiv preprint arXiv:2304.03117*, 2023. 1, 3, 8

[56] M. Zhou, R. Hyder, Z. Xuan, and G. Qi. Ultravatar: A realistic animatable 3d avatar diffusion model with authenticity guided textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1238–1248, 2024. 3, 5

[57] X. Zhu, X. Liu, Z. Lei, and S. Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017. 2

[58] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer graphics forum*, volume 37, pages 523–550. Wiley Online Library, 2018. 2