Rapid Face Asset Acquisition with Recurrent Feature Alignment

SHICHEN LIU, University of Southern California, USC Institute for Creative Technologies, USA
YUNXUAN CAI, USC Institute for Creative Technologies, USA
HAIWEI CHEN, University of Southern California, USC Institute for Creative Technologies, USA
YICHAO ZHOU, University of California Berkeley, USA
YAJIE ZHAO, USC Institute for Creative Technologies, USA

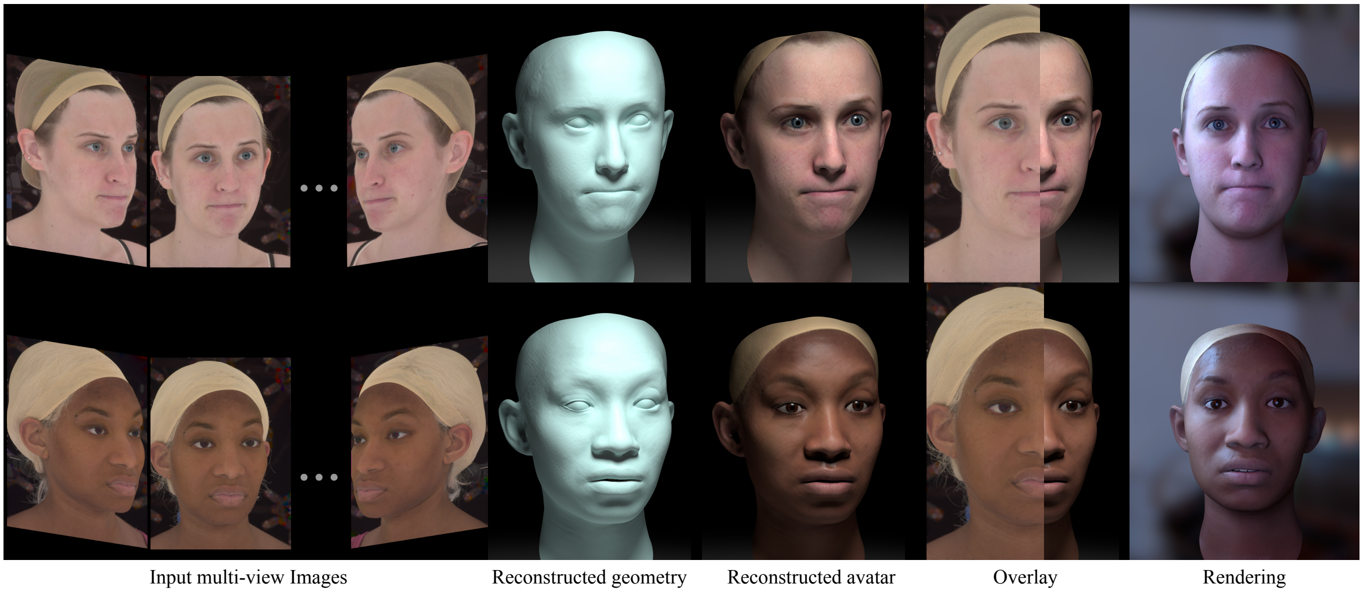


Fig. 1. Our end-to-end framework infers production-ready face assets from multi-view images, with a state-of-the-art efficiency at 4.5 frame per second. The inferred assets contain both the pore-level geometry and a skin reflectance property maps (specularity and diffuse maps), allowing physically-based renderings in various lighting conditions. Notably, our framework is fully automatic: the results shown are direct output of our designed neural network without any manual editing and post processing.

We present **Re**current **F**eature **A**lignment (ReFA), an end-to-end neural network for the very rapid creation of production-grade face assets from multi-view images. ReFA is on par with the industrial pipelines in quality for producing accurate, complete, registered, and textured assets directly applicable to physically-based rendering, but produces the asset end-to-end, fully automatically at a significantly faster speed at 4.5 FPS, which is unprecedented among neural-based techniques. Our method represents face geometry as a position map in the UV space. The network first extracts per-pixel features in both the multi-view image space and the UV space. A recurrent module then iteratively optimizes the geometry by projecting the image-space features to the UV space and comparing them with a reference UV-space feature. The optimized geometry then provides pixel-aligned signals for the inference of high-resolution textures. Experiments have validated that ReFA achieves a median error of $0.603mm$ in geometry reconstruction, is robust to extreme pose and expression, and excels in sparse-view settings. We believe that the progress achieved by our network enables lightweight, fast face assets acquisition that significantly boosts the downstream applications, such as avatar creation and facial performance capture. It will also enable massive database capturing for deep learning purposes.

CCS Concepts: • **Computing methodologies → Shape inference**; *Computational photography*; *Motion capture*; *Neural networks*; *Shape representations*.

Additional Key Words and Phrases: Human Face Capture, Face Animation, Facial performance capturing, Geometry registration

ACM Reference Format:
Shichen Liu, Yunxuan Cai, Haiwei Chen, Yichao Zhou, and Yajie Zhao. 2022. Rapid Face Asset Acquisition with Recurrent Feature Alignment. *ACM Trans. Graph.* 1, 1, Article 214 (September 2022), 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Shichen Liu, University of Southern California, USC Institute for Creative Technologies, USA, liushichen95@gmail.com; Yunxuan Cai, USC Institute for Creative Technologies, USA, ycai@ict.usc.edu; Haiwei Chen, University of Southern California, USC Institute for Creative Technologies, USA, chw9308@hotmail.com; Yichao Zhou, University of California Berkeley, USA, zyc@berkeley.edu; Yajie Zhao, USC Institute for Creative Technologies, USA, zhao@ict.usc.edu.

0730-0301/2022/9-ART214 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Photo-realistic face avatar capture has become a key element in entertainment media due to the realism and immersion it enables. As the digital assets created from photos of human faces surpass their artist-created counterparts in both diversity and naturalness, there are increasing demands for the digitized face avatars in the majority of the sectors in the digital industry: movies, video games, teleconference, and social media platforms, to name a few. In a studio setting, the term "avatar" encompasses several production standards for a scanned digital face, including high-resolution geometry (with pore-level details), high-resolution facial textures (4K) with skin reflectance measurements, as well as a digital format that is consistent in mesh connectivity and ready to be rigged and animated. These standards together are oftentimes referred to as a production-ready face avatar.

In this paper, we consider a common face acquisition setting where a collection of calibrated cameras capture the color images that are processed into a full set of assets for a face avatar. In general, today's professional setting employs a two-step approach to the creation of the face assets. The first step computes a middle-frequency geometry of the face (with noticeable wrinkle and facial muscle movement) from multi-view stereo (MVS) algorithms. A second registration step is then taken to register the geometries to a template meth connectivity, commonly of lower resolution with around 10k to 50k vertices. For production use, the registered base mesh is augmented by a set of texture maps, composed of albedo, specular and displacement maps, that are computed via photogrammetry cues and specially designed devices (e.g. polarizers and gradient light patterns in [Ghosh et al. 2011a; Ma et al. 2008]). The lower-resolution base mesh is combined with a high resolution displacement maps to represent geometry with pore, freckle-level details. Modern physically-based rendering agents further utilize the albedo and specularity maps to render the captured face in photo-realistic quality.

While the avatars acquired thereby achieve satisfactory realism, many difficulties in this setting inevitably pose high-quality face avatar capturing as a costly operation that is far from mass production and easy accessibility. More specifically, traditional MVS algorithms and registration take hours to run for a single scan frame. The registration process is also error-prone, oftentimes requiring manual adjustment to the initialization and clean-up by the professional artists. In addition, special devices (e.g. polarizers) are needed for capturing skin reflectance properties. The long production cycle, intensive labor work and equipment cost for special devices holds back a wider availability of high-quality facial capturing.

The growing demands for face acquisition in both research and digital production would greatly benefit from a much faster and fully automatic system that produces professional-grade face avatars. Fortunately, efforts towards this end are well found in recently proposed neural-learning-based techniques. Model-based approaches such as DFNRMVS [Bai et al. 2020] incorporate a 3D morphable model as the prior to reconstruct face geometry from a sequence of image input. Despite achieving a vast increase in efficiency, they have yet to succeed in matching the quality and completeness of production-ready avatars, due to the limited expressiveness and flexibility of the parametric space of the morphable model. On the other hand, deep stereo matching approaches, such as [Im et al. 2018], achieve higher accuracy in 3D reconstruction by accurately regressing depth under geometric priors. Our adaptation of these methods to the facial reconstruction settings has revealed that the best performing deep MVS framework [Im et al. 2018] infers shapes within 0.88mm median error, and the inference time is within a second. However, nontrivial steps are still required to obtain the registered meshes and the corresponding texture maps. Recently, ToFu [Li et al. 2021] have shown dedicated designs for neural face acquisition, achieving state-of-the-art accuracy in the face reconstruction and providing a solution that combines reconstruction with registration in an end-to-end manner. ToFu learns the probability distributions of the individual vertices in a volumetric space, posting the reconstruction as a coarse-to-fine landmark classification problem. However, the formulation limits ToFu to use a relatively low-resolution geometry representation, which is in addition incompatible with texture inference.

In light of the progress needed to be made, our goal is a comprehensive, animation-ready neural face capturing solution that can produce production-grade dense geometry (combining a mid-frequency mesh with high-resolution displacement maps), complete texture maps (high-resolution albedo and specular maps) required by most PBR skin shaders, and consistent mesh connectivity[1] across subjects and expressions. Most importantly, our proposed model aims to be *highly efficient*, creating the comprehensive face assets within a second, *fully automatic*, an end-to-end system without the need for manual editing and post-processing, and *device-agnostic*, easily adaptable to any novel multi-view capturing rigs with minimal fine-tuning, including light-weight systems with sparse views.

Our proposed model is Recurrent Feature Alignment (ReFA), the first end-to-end neural-based system designed to faithfully capture both the geometry and the skin assets of a human face from multi-view image input and fully automatically create a face avatar that is production-ready. Compared to the state-of-the-art method [Li et al. 2021], ReFA boosts both the accuracy by 20% to a median error of 0.608mm, and a 40% increase in speed, inferring high-quality textured shapes at 4.5FPS. The face geometries inferred by ReFA not only outperforms the best deep MVS method [Im et al. 2018], but they are reconstructed at a representation consistent in mesh connectivity that provides dense correspondences across subjects and expressions. In addition, a parallel texture inference network that shares the same representation with the geometry produces a full set of high-resolution appearance maps that allow for photo-realistic renderings of the reconstructed face.

ReFA is based on two key designs to realize the aforementioned goals. The first is the adoption of the position map [Feng et al. 2018] for representing geometry in a UV space. Such representation is not only amenable to effective processing with image convolution networks, but it offers an efficient way to encode dense, registered shape information (a 128×128 size of position map encodes up to 16K vertices) across subjects and expressions, and organically aligns the geometry and texture space for the inference of high-frequency

[1] We use mesh connectivity to describe a mesh with well-define faces "f" and vertex texture coordinates "vt" but without vertex coordinates "v".

displacement maps and high-resolution textures. The position map also provides pixel-level flexibility for geometry optimization, which allows modeling of extreme expression, non-linear muscle movement and other challenging cases. In this paper, we adopt a position map of 512 × 512 size, with a capacity of around 260K vertices that are well capable of modeling middle-frequency details directly using a neural network. The second design is a learned recurrent face geometry optimizer that effectively aligns UV-space semantic feature with the multi-view visual features for reconstruction with consistent mesh connectivity. The recurrent optimization is centered around a per-pixel visual semantic correlation (VSC) that serves to iteratively refine the face geometry and a canonical head pose. The refined geometry then provides pixel-aligned signals to a texture inference network that accurately infers albedo, specular and displacement map in the same UV space.

Experiments in Section 6 have validated that our system ReFA meets its goal in fast and accurate multi-view face reconstruction, outperforming the previous state-of-the-art methods in both visual and numerical measurements. We further show in an ablation study that our design choices are effective and our model is robust to sparse view input. As ReFA utilizes a flexible shape representation and produces a full set of face assets that is ready for production-level animation, we demonstrate applications in avatar creation, 4D capture and adaptation of our model to the productions of other digital formats.

In summary, our contributions are:

- ReFA, the first neural-based comprehensive face capturing system that faithfully reconstructs both the geometry and the skin assets of a human face from multi-view images input and fully automatically create a 3D face avatar that is production-ready. Our model outperforms previous neural-based approaches in both accuracy and speed, with a median error of 0.6*mm* and a speed at 4.5FPS.
- Novel formulations of a recurrent geometry optimizer that operates on UV-space geometry features and provides an effective solution to high-quality face asset creation.
- The proposed system has great application values in many downstream tasks including rapid avatar creation and 4D performance capture. We believe the improvement in speed and accuracy brought by our system would greatly facilitate the accessibility of face capturing to support an emerging industrial field that is data-hungry.

2 RELATED WORK

Multi-view Stereo Face Capture. Today's high-quality performance capture of human face is commonly done with passive or active MVS capture systems (e.g. [Beeler et al. 2010; Ghosh et al. 2011b; Ma et al. 2007]). The complete procedures to acquire 3D avatars from the captured data involve two major steps from multi-view stereopsis to registration, and each of them has been studied as individual problem.

Multi-view stereopsis is commonly the first step for acquiring dense 3D geometry and the algorithms proposed in the past have emphasized various deigns for both joint view selection [Kang et al. 2001; Schönberger et al. 2016; Strecha et al. 2006] and normal/depth estimation [Bleyer et al. 2011; Gallup et al. 2007; Schönberger et al. 2016]. Neural-based MVS approaches proposed in recent years [Chang and Chen 2018; Gu et al. 2020; Huang et al. 2018; Im et al. 2018] have significantly increased the efficiency and generalized well to as few as a pair of stereo images. Since our focus in on the digital face reconstruction, we refer interested readers to [Ackermann and Goesele 2015; Laga et al. 2020] for more comprehensive reviews.

The output of the multi-view stereopsis is, in general, in the form of dense depth maps, point clouds, or 3D surfaces. Regardless of the specific representations, the geometries are processed into 3D meshes, and a follow-up registration process aligns all captured shapes to a predefined template mesh connectivity. The registration process is done either by explicitly regressing coefficients of a parametric face model [Amberg et al. 2008; Blanz and Vetter 1999, 2003; Li et al. 2020a], directly optimizing shape with a non-rigid Iterative Closest Point registration algorithm [Li et al. 2009] or globally optimizing over a dataset of scanned face to find the groupwise correspondences [Bolkart and Wuhrer 2015; Zhang et al. 2016].

Learned Face Reconstruction from Images. Settings where geometries are reconstructed from a monocular image or a sparse set of views are in general ill-posed. Efforts in this direction are thus mainly data-driven, where a popular line of methods can be considered as fitting parametric models to the target image space, as seen in [Garrido et al. 2016; Levine and Yu 2009; Thies et al. 2016]. Deep neural networks have been utilized in most recent works [Feng et al. 2021; Genova et al. 2018; Richardson et al. 2016, 2017; Sanyal et al. 2019; Tewari et al. 2018, 2017] for the regression of the parameters that drive a morphable model. The quality and accuracy of monocular face reconstruction, albeit appealing in some circumstances, are not suitable for production use in professional settings. The inherent ambiguity of focal length, scale, and shape oftentimes lead a monocular reconstruction network to produce different shapes for the same face viewed at different angle [Bas and Smith 2019].

Few works prior to us have attempted a data-driven approach to MVS face reconstruction. When the camera views are abundant, modern face capture pipelines, e.g. [Beeler et al. 2011; Borshukov et al. 2005; Fyffe et al. 2017], have demonstrated highly detailed and precise face reconstruction with pore-level appearance without the needs for a learned mapping in their computations. However, as introduced in the previous section, the manual costs and computation overhead of these pipelines have at least inspired many to propose neural-based frameworks that automate and accelerate key steps in face capture applications, e.g. deep stereo matching and registration. The recent work ToFu [Li et al. 2021] is a notable neural framework that offers end-to-end solution for registered face geometry reconstruction, based on the prediction of the probabilistic distributions of individual vertices of a template face mesh. ReFA expands its setting by including texture inference in the end-to-end network, while our formulation, compared to ToFu, is able to infer denser geometry at an even faster speed.

Learned Optimizers for Geometry Inference. Our method is related to a broader trend of solving geometrical optimization problem with recurrent neural networks, where feature correlations are computed iteratively to refine optical flow [Teed and Deng 2020], depth [Yao et al. 2019] or vanishing points [Liu et al. 2021]. Motivated by the success of neural optimizers in geometric refinement, we consider a novel reformulation of the face reconstruction as an iterative refinement to a UV position map. Different from past literature that computes correlations in image spaces, our method aligns vertices embedded in the UV-space position map to pixels from multiple image-space views.

Textures Inference for Photo-realistic Rendering. Controlled environments are usually needed to collect the ground-truth photo-realistic appearance of a human face, exemplified by the Light Stage [Debevec et al. 2000; Ghosh et al. 2011a; Graham et al. 2013]. Neural-based reconstruction network trained on the captured appearance information generally employs an encoder-decoder structure to simultaneously infer skin reflectance and illumination alongside the geometry [Chen et al. 2019; Yang et al. 2020], where the quality of the inferred textures were limited due to either reliance on synthetic data or oversimplified reflectance model. The recent works [Lattas et al. 2020] and [Li et al. 2020b] both utilized generative adversarial training and an image translation network to performance texture inference that are photo-realistic and render-ready, where high-quality albedo, displacement and specular maps were decoupled from the input face images.

3 DATA COLLECTION

3.1 Capture System Setup

Our training data is acquired by a *Light Stage* scan system, which is able to capture at pore-level accuracy in both geometry and reflectance maps by combining photometric stereo reconstruction [Ghosh et al. 2011b] and polarization promotion [LeGendre et al. 2018]. The camera setup consists of 25 Ximea machine vision cameras, including 17 monochrome and 8 color cameras. The monochrome cameras, compared to their color counterparts, support more efficient and higher-resolution capturing, which are essential for sub-millimeter geometry details, albedo, and specular reflectance reconstruction. The additional color cameras aid in stereo-based mesh reconstruction. The RGB colors in the captured images are obtained by adding successive monochrome images recorded under different illumination colors as shown in [LeGendre et al. 2018]. We selected a FACS set [Ekman and Friesen 1978] which combines 40 action units to a condensed set of 26 expressions for each subjects to perform. A total number of 64 subjects, ranging from age 18 to 67, were scanned.

3.2 Data Preparation

Starting from the multi-view images, we first reconstruct the geometry of the scan with neutral expression of the target subject using a multi-view stereo (MVS) algorithm. Then the reconstructed scans are registered using a linear fitting algorithm based on a 3D face morphable model, similar to the method in [Blanz and Vetter 1999]. In particular, we fit the scan by estimating the morphable model coefficients using linear regression to obtain an initial shape in the template topology. Then a non-rigid Laplacian deformation is performed to further minimize the surface-to-surface distance. We deform all the vertices on the initially fitted mesh by setting the landmarks to match their correspondence on the scan surface as data terms and use the Laplacian of the mesh as a regularization term. We adopt and implement a variation of [Sorkine et al. 2004] to solve this system. Once the neutral expression of the target person is registered, the rest of the expressions are processed based on it. We first adopted a set of generic blendshapes (a set of vertex differences computed between each expression and the neutral, with 54 predefined orthogonal expressions) and the fitted neutral base mesh to fit the scanned expressions and then performed the same non-rigid mesh registration step to further minimize the fitting error. Additionally, to ensure the cross-expression consistency for the same identity, optical flow from neutral to other expressions is added as a dense consistency constraint in the non-rigid Laplacian deformation step. This 2D optical flow will be further used as a projection constraint when solving for the 3D location of a vertex on the target expression mesh during non-linear deformation.

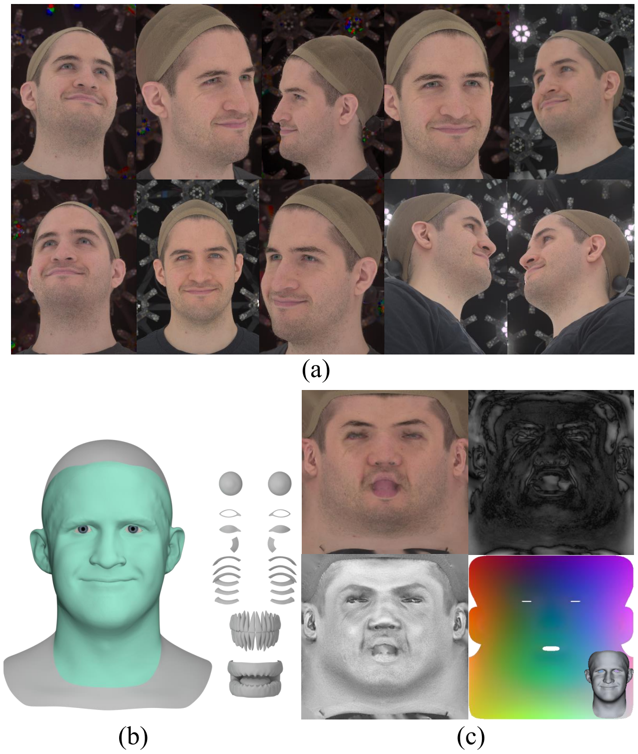


Fig. 2. An example set of subject data used for training. **(a)** Selected views of the captured images as input. **(b)** Processed geometry in the form of a 3D mesh. In addition to the face, head, and neck, our model represents teeth, gums, eyeballs, eye blending, lacrimal fluid, eye occlusion, and eyelashes. The green region denotes the face area that our model aims to reconstruct. The other parts are directly adopted from a template **(c)** $4K \times 4K$ physically-based skin properties, including albedo (bottom-left), specular (top-left) and displacement maps (top-right) used for texture supervision, and the 512×512 position map (bottom-right), converted from the 3D mesh in (b), used for geometry supervision.

All the processed geometries and textures share the same mesh connectivity and thus have dense vertex-level correspondence. The diffuse-specular separation is computed under a known spherical

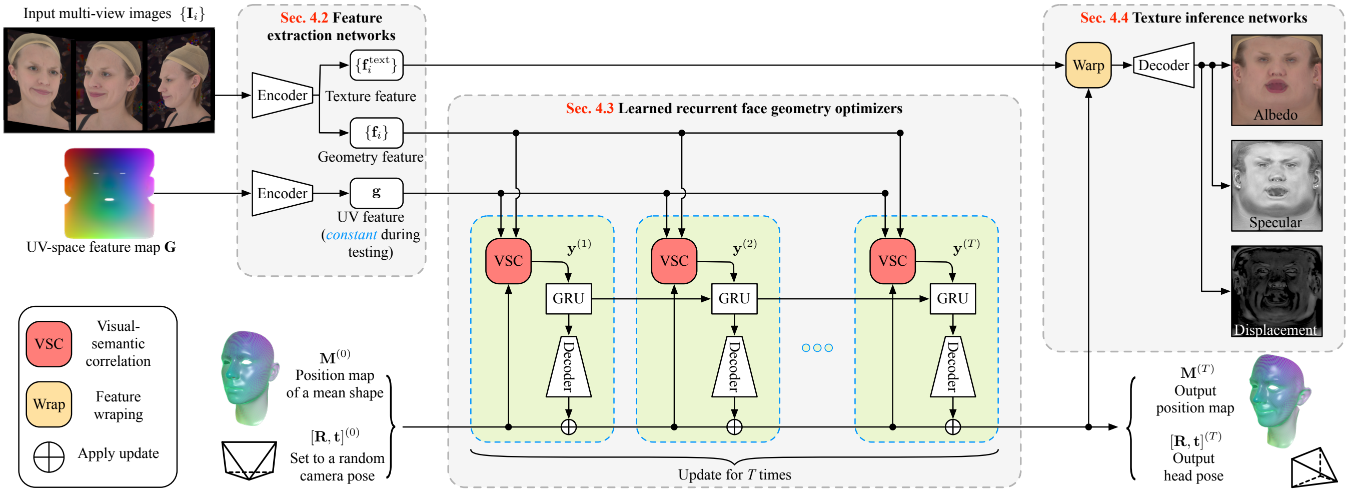

Input multi-view images $\{\mathbf{I}_i\}$
Sec. 4.2 Feature extraction networks
Encoder
$\{\mathbf{f}_i^{\text{text}}\}$
Texture feature
$\{\mathbf{f}_i\}$
Geometry feature
Encoder
$\mathbf{g}$
UV feature (constant during testing)
UV-space feature map $\mathbf{G}$
Sec. 4.3 Learned recurrent face geometry optimizers
VSC
$\mathbf{y}^{(1)}$
GRU
Decoder
VSC
$\mathbf{y}^{(2)}$
GRU
Decoder
VSC
$\mathbf{y}^{(T)}$
GRU
Decoder
Update for T times
Sec. 4.4 Texture inference networks
Warp
Decoder
Albedo
Specular
Displacement
VSC Visual-semantic correlation
Wrap Feature wraping
Apply update
$\mathbf{M}^{(0)}$ Position map of a mean shape
$[\mathbf{R},\mathbf{t}]^{(0)}$ Set to a random camera pose
$\mathbf{M}^{(T)}$ Output position map
$[\mathbf{R},\mathbf{t}]^{(T)}$ Output head pose

Fig. 3. Network architecture of ReFA. Our model recurrently optimizes for the facial geometry and the head pose based on computation of visual-semantic correlation (VSC) and utilizes the pixel-aligned signals learned thereof for high-resolution texture inference.

illumination [Ma et al. 2007]. The pore-level details of the geometry are computed by employing albedo and normal maps in the stereo reconstruction [Ghosh et al. 2011b] and represented as displacement maps to the base mesh. The full set of the generic model consists of a base geometry, a head pose, and texture maps (*albedo*, *specular intensity*, and *displacement*) encoded in $4K$ resolution. 3D vertex positions are rasterized to a three-channel HDR bitmap of 256×256 pixels resolution to enable joint learning of the correlation between geometry and albedo. 15 camera views are used for the default setting to infer the face assets with our neural network. Figure 2 shows an example of captured multi-view images and a full set of our processed face asset that is used for training. In addition to the primary assets generated using our proposed network, we may also assemble secondary components (e.g., eyeballs, lacrimal fluid, eyelashes, teeth, and gums) to the network-created avatar. Based on a set of handcrafted blendshapes with all the primary and secondary parts, we linearly fit the reconstructed mesh by computing the blending weights that drive the secondary components to travel with primary parts, such that the eyelashes will travel with eyelids. Except for the eyeball, other secondary parts share a set of generic textures for all the subjects. For eyeball, we adopt an eyeball assets database [Kollar 2019] with 90 different pupil patterns to match with input subjects. Note that all the eyes share the same shape as in [Kollar 2019] and in our database. For visualization purposes, we manually pick the matching eye color. The dataset is split into 45 subjects for the training and 19 for the evaluation. Each set of capture contains a neutral face and 26 expressions, including extreme face deformation, asymmetrical motions, and subtle expressions.

4 METHOD

Overview. As shown in Figure 3, our end-to-end system takes multi-view images and a predefined template UV position map in a canonical space as input and produces 1) an updated position map, 2) estimated head pose (3D rotation and translation) parameters to rigidly align the updated position map in camera space to the canonical template space and 3) texture maps including the albedo map, the specular map, and the displacement map. To support direct use for animation, the position map and the texture maps form the entire face assets for realistic rendering and are all conveniently defined in the same (or up-sampled) UV space. In the following Section 4.1, we detail the representations of the aforementioned entities as well as the camera model.

The subsequent sections are dedicated to the three main components of our system: (1) the feature extraction networks (Section 4.2) that extract features for the input images and a predefined UV-space feature map; (2) the recurrent face geometry networks (Section 4.3) that take the output of the feature extraction network and use a learned neural optimizer to iteratively refine the geometry from an initial condition to a finer shape; and (3) the texture inference networks (Section 4.4) that take the inferred geometry and the multi-view texture features to infer the high-resolution texture maps.

4.1 Preliminaries

Data Format. Table 1 specifies the symbols and formats of the input and output data involved in our pipeline. In addition to the details provided in the table, the input multi-view images are indexed by the camera view: $\{\mathbf{I}_i\}_{i=1}^{K}$ from K views with known camera calibrations $\{\mathbf{P}_i \mid \mathbf{P}_i \in \mathbb{R}^{3\times4}\}_{i=1}^{K}$. All feature maps are bilinearly sampled given real-valued coordinates. Specifically, the displacement map is designed to be added along the normal direction of the position map to provide high-frequency geometry details.

Geometry Representation. The position map $\mathbf{M}$ is our representation of the face geometry. $\mathbf{M}$ comes with a UV mapping from a template mesh connectivity, and thus each pixel on $\mathbf{M}$ is mapped to a vertex or a surface point of a 3D mesh. All the scanned meshes with different identities and expressions share the same UV mapping. Furthermore, each pixel in $\mathbf{M}$ stores the $\mathbb{R}^3$ coordinates of

Table 1. Symbol table. By default, we set $H = W = H_t = W_t = 512$.

Name	Symbol	Dimension
Input multi-view images	$\mathbf{I}$	$\mathbb{R}^{H\times W\times 3}$
Camera parameters	$\mathbf{P}$	$\mathbb{R}^{3\times 4}$
Head pose	$[\mathbf{R},\mathbf{t}]$	$\mathbb{R}^{3\times 4}$
UV-space position map	$\mathbf{M}$	$\mathbb{R}^{H_t\times W_t\times 3}$
UV-space albedo map	$\mathbf{A}$	$\mathbb{R}^{8H_t\times 8W_t\times 3}$
UV-space specular map	$\mathbf{S}$	$\mathbb{R}^{8H_t\times 8W_t}$
UV-space displacement map	$\mathbf{D}$	$\mathbb{R}^{8H_t\times 8W_t}$

its location in the canonical space. It therefore suffices to define a high-resolution geometry given a dense mesh and a UV mapping, as converting the position map to a 3D mesh simply amounts to setting the vertex positions of the mesh.

The UV-space representation of the geometry is in particular amenable to shape inference with a neural network, as the position map links the geometry space to a texture space that can be processed by 2D convolutional neural networks effectively. Since each pixel in $\mathbf{M}$ corresponds to a mesh vertex, a position map $\mathbf{M}$ of 512×512 resolution supports a dense geometry of up to $2.6M$ vertices. Thus we believe that the position map is a powerful geometry representation that enables inference of highly detailed face assets.

Our system uses a common UV space across all the subjects and the expressions. This ensures that all the inferred geometries are registered. An additional advantage is that we can use any mesh connectivity that embraces the same UV mapping to sample from the position map and recover the vertex coordinates.

Camera Model. We follow the pinhole camera model. For a 3D point $\mathbf{X} = [X, Y, Z]^T \in \mathbb{R}^3$ in the world space, its projection on the image plane $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ can be computed as: $z \cdot [x, y, 1]^T = \mathbf{P} \cdot [X, Y, Z, 1]^T$, where $\mathbf{P} \in \mathbb{R}^{3\times 4}$ is the camera parameters including the intrinsic and extrinsic matrices. For convenience, we denote this relationship as

$$\mathbf{x} = \Pi_{\mathbf{P}}(\mathbf{X}). \tag{1}$$

4.2 Feature Extraction Networks

Image Space Features. From the input multi-view images $\{\mathbf{I}_i\}_{i=1}^K$, we use a ResNet-like [He et al. 2016] backbone network to extract 2D features at $\frac{1}{8}$ of the image resolution. The output are split into two branches: the geometry feature $\mathbf{f}_i \in \mathbb{R}^{\frac{W}{8}\times\frac{H}{8}\times C}$ and the texture feature $\mathbf{f}_i^{\text{text}} \in \mathbb{R}^{\frac{W}{8}\times\frac{H}{8}\times C_t}$ given the view index i. The geometry feature map is used for estimating the head pose, represented as a 6-DoF rigid transformation $[\mathbf{R},\mathbf{t}]$, and the position map $\mathbf{M}$ (Section 4.3). The texture feature map is used for generating high-resolution texture maps including albedo maps $\mathbf{A}$, specular maps $\mathbf{S}$, and displacement maps $\mathbf{D}$ (Section 4.4).

UV Space Features. As shown in Figure 4, from the template mesh and its UV mapping, we assemble the UV-space feature map $\mathbf{G} \in \mathbb{R}^{W_t\times H_t\times 36}$ by concatenating the following features for each pixel $\mathbf{u}$: (1) the 2D coordinates of $\mathbf{u}$ itself, normalized to $[-1, 1]^2$ (Figure 4a); (2) the corresponding 3D coordinates of $\mathbf{u}$ in the mean face mesh (Figure 4b); (3) the one-hot encoding of its face region, where we manually create a semantic face region map including 31 regions (Figure 4c).

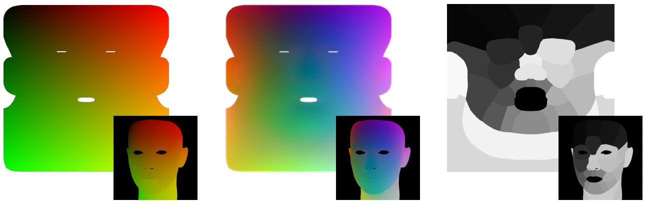

(a) UV Coordinates (b) Position Map (c) Face Region Map

Fig. 4. Composition of the UV-space feature $\mathbf{G}$. $\mathbf{G}$ is a concatenation of (a) UV space coordinates, (b) position map of a mean shape and (c) a carefully crafted face region map (31 dimensional one-hot vector). The composition serves to encode the facial semantics and the geometry priors necessary for the future steps.

We process the feature $\mathbf{G}$ using a convolutional neural network and get the resulting UV space feature map $\mathbf{g} \in \mathbb{R}^{\frac{W_t}{8}\times\frac{H_t}{8}\times C}$. Since $\mathbf{G}$ is a constant, the UV feature map $\mathbf{g}$ can also be understood as a trainable parameter, which is regularized by the CNN architecture and the construction of $\mathbf{G}$. Once trained, we discard the neural network and set $\mathbf{g}$ as a fixed constant.

4.3 Recurrent Face Geometry Optimizer

Our network tackles the reconstruction task by solving an alignment problem. A UV-space position map that represents the geometry is first initialized to be the mean face shape. In a practical face capturing setting, the pose of the head relative to the geometry defined by the position map is unknown, so we initialize the head pose as a random pose that is visible in all cameras. The initialized face geometry, when projected to the image space, will show misalignment with the underlying geometry depicted in the multi-view images. For instance, a projection of the eye on the initialized face geometry is likely not aligned with the actual eye location in the image space. Our framework thus optimizes the face geometry iteratively, such that the projection of the face in the UV space gradually converge to the ground truth locations in all image views. In order to solve the alignment problem, the features in the UV space and the image space (Section 4.2) are joined in a unified feature space, such that the corresponding locations in both spaces are trained to encode similar features. We compute a dense all-pair correlation between the UV space and the image space and use a recurrent neural network to find the optimal matching in the correlation tensor. Once the optimal matching is found in this process, the shape depicted by the position map naturally reconstructs the shape depicted in the multi-view images.

In each network step, we update the position map $\mathbf{M}$ as well as the head pose $[\mathbf{R},\mathbf{t}]$ separately, given the correlation tensor between the two misaligned feature maps of interest, namely the UV feature map $\mathbf{g}$ and the image space feature $\mathbf{f}$. We term the optimizer that performs such actions the Recurrent Face Geometry Optimizer. In the following paragraphs, we describe in detail how our optimizer initializes, updates, and finalizes the corrections in order to recover $\mathbf{M}$ and $[\mathbf{R},\mathbf{t}]$.

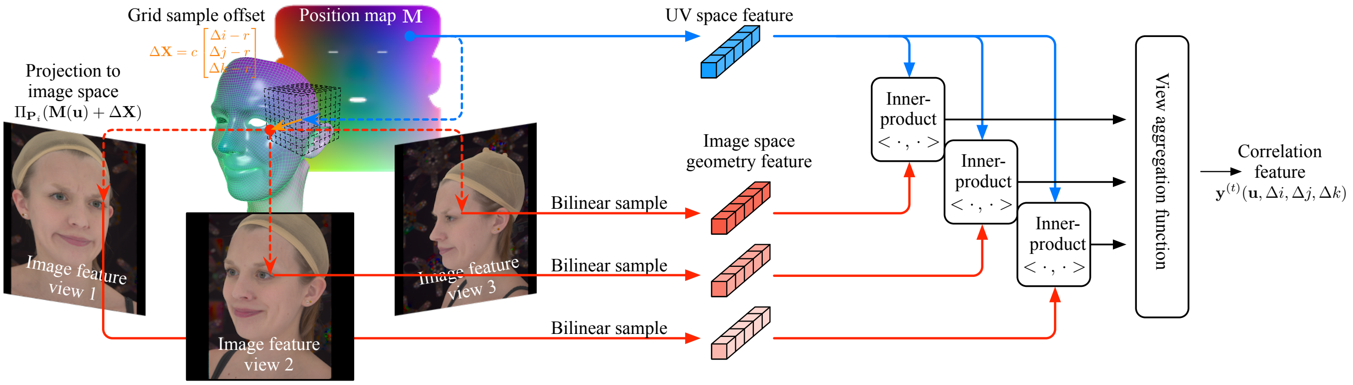

Grid sample offset
$\Delta\mathbf{X} = c \begin{bmatrix} \Delta i - r \\ \Delta j - r \\ \Delta k - r \end{bmatrix}$
Position map M
Projection to image space
$\Pi_{\mathbf{P}_i}(\mathbf{M}(\mathbf{u}) + \Delta\mathbf{X})$
Image feature view 1
Image feature view 2
Image feature view 3
UV space feature
Image space geometry feature
Bilinear sample
Bilinear sample
Bilinear sample
Inner-product $< \cdot, \cdot >$
Inner-product $< \cdot, \cdot >$
Inner-product $< \cdot, \cdot >$
View aggregation function
Correlation feature
$\mathbf{y}^{(t)}(\mathbf{u}, \Delta i, \Delta j, \Delta k)$

Fig. 5. An illustration of visual-semantic correlation (VSC). A 3D local grid is built around the 3D position of each pixel in the UV-space position map. The volume of correlation feature is then constructed by taking the inner product between each UV-space feature in the local grid the its projected features in the multi-view image space. The correlation feature is a local representation of the alignment between the observed visual information and the semantic priors.

Initialization. we initialize the head pose with a randomly selected rotation and translation of the mean camera distance (≈1.3 meters). We also initialize the position map as the mean shape $\mathbf{M}^0 = \mathcal{M}$. Such design is due to the fact that the captured subjects' head may turn from an upright position in a more practical setting. In other word, we do not assume that the absolute pose of the head is known.

Compute Gradient. The Recurrent Face Geometry Optimizer is based on a recurrent neural network (RNN) composed of Gated Recurrent Units (GRU) [Cho et al. 2014], which computes the gradient on the pose (rotation $\mathbf{R}$, translation $\mathbf{t}$) and the geometry (position map $\mathbf{M}$). At the t-th step, the neural network process could be written as:

$$\mathbf{y}^{(t)} \leftarrow \mathrm{VSC}(\{\mathbf{f}_i\}_{i=1}^{K}, \mathbf{g}, \mathbf{R}^{(t-1)}, \mathbf{t}^{(t-1)}, \mathbf{M}^{(t-1)}), \quad (2)$$

$$\mathbf{h}^{(t)} \leftarrow \mathrm{GRU}(\mathbf{y}^{(t)}, \mathbf{h}^{(t-1)}), \quad (3)$$

$$\Delta^{(t)} \leftarrow \mathrm{Decoder}(\mathbf{h}^{(t)}). \quad (4)$$

In Equation (2), our *Visual Semantic Correlation (VSC) network* (Section 4.3.1) matches the UV space feature and the image space feature, and produces a correlation feature map $\mathbf{y}^{(t)} \in \mathbb{R}^{\frac{W_t}{8} \times \frac{H_t}{8} \times C_{\mathrm{VSC}}}$. Next, $\mathbf{y}^{(t)}$ is fed to a GRU-based RNN [Cho et al. 2014] and the hidden state $\mathbf{h}^{(t)}$ is updated from the previous $\mathbf{h}^{(t-1)}$ in Equation (3). Then, the *Geometry Decoding Network* (Section 4.3.2) processes the hidden vector $\mathbf{h}^{(t)}$ and computes the geometry update tuple $\Delta^{(t)} = \left(\boldsymbol{\delta R}^{(t)}, \boldsymbol{\delta t}^{(t)}, \boldsymbol{\delta}\mathbf{M}^{(t)}\right)$. The update tuple is applied by

$$\begin{aligned} \mathbf{R}^{(t)} &\leftarrow \mathbf{R}^{(t-1)} \cdot \boldsymbol{\delta R}^{(t)} \\ \mathbf{t}^{(t)} &\leftarrow \mathbf{t}^{(t-1)} + \boldsymbol{\delta t}^{(t)} \\ \mathbf{M}^{(t)} &\leftarrow \mathbf{M}^{(t-1)} + \boldsymbol{\delta}\mathbf{M}^{(t)}. \end{aligned} \quad (5)$$

Given the total iterations T, the final output of the optimizer is simply $[\mathbf{R}, \mathbf{t}]^{(T)}$ and $\mathbf{M}^{(T)}$.

4.3.1 *Visual-Semantic Correlation (VSC) Networks.*

To predict the update tuple, we construct a 2D feature map containing the signals where $\boldsymbol{\delta}\mathbf{M}^{(t)}(\mathbf{u})$ and $[\boldsymbol{\delta R}^{(t)}, \boldsymbol{\delta t}^{(t)}]$ should orient. Our method is illustrated in Figure 5.

The 2D feature map computes similarity between the multi-view geometry features $\mathbf{f}$ and the UV features $\mathbf{g}$ by constructing a correlation volume. Specifically, let $\hat{\mathbf{M}}^{(t)} = \mathbf{R}^{(t)}\mathbf{M}^{(t)} + \mathbf{t}^{(t)}$ be the transformed position map at t-th step, we first enumerate a 3D regular grid of size $(2r + 1) \times (2r + 1) \times (2r + 1)$ around $\hat{\mathbf{M}}^{(t)}(\mathbf{u})$ for each pixel $\mathbf{u}$ in the UV space, where $r \in \mathbb{N}$ is the grid resolution. We then project the grid points to the image space using the camera parameters $\mathbf{P}_i$, and compare the features between the corresponding points in the image space of $\mathbf{f}$ and the UV space of $\mathbf{g}$.

We use bilinear sampling to sample the feature at non-integer indices in both spaces, and calculate the similarity as the inner-product between two features: the UV features that contain *semantic* information, and the image features that contain *visual* information. We therefore call this process *Visual-Semantic Correlation (VSC)*. Mathematically, this process is represented as

$$\tilde{\mathbf{y}}_i^{(t)}(\mathbf{u}, \Delta i, \Delta j, \Delta k) = \left\langle \mathbf{f}_i \left(\Pi_{\mathbf{P}_i} \left(\hat{\mathbf{M}}^{(t)}(\mathbf{u}) + c \begin{bmatrix} \Delta i - r \\ \Delta j - r \\ \Delta k - r \end{bmatrix} \right) \right), \mathbf{g}(\mathbf{u}) \right\rangle, \quad (6)$$

where $\tilde{\mathbf{y}}_i^{(t)}$ is the constructed 5D correlation tensor for i-th camera view, $\mathbf{f}_i$ and $\mathbf{g}$ are the feature maps introduced in Section 4.2, $\Pi_{\mathbf{P}_i}$ is the projection operator introduced in Equation (1), $\mathbf{u}$ is the 2D coordinates in the UV space, r is the grid resolution, c is the searching radius controlling the grid size, $\Delta i, \Delta j, \Delta k \in \{1, 2, \ldots, 2r + 1\}$ is the offset in the x-axis, y-axis, and z-axis, respectively, and "$\langle \cdot, \cdot \rangle$" is the inner-product operator. The constructed 5D correlation tensor can be understood as guidance features for drawing the 3D points, represented by $\mathbf{M}^{(t)}(\mathbf{u})$, to new locations. After the correlation tensor $\tilde{\mathbf{y}}_i^{(t)}$ is computed, we flatten it along the dimensions of Δi, Δj, and Δk. Finally, the flattened features at each view are fused by a chosen aggregation function σ to produce the input feature to the decoder, for which we have chosen the max pooling function:

$$\mathbf{y}^{(t)} = \sigma(\tilde{\mathbf{y}}_0^{(t)}, \ldots, \tilde{\mathbf{y}}_K^{(t)}) \quad (7)$$

4.3.2 *Geometry Decoding Network.*

The decoder, termed Geometry Decoding Network, decodes the the hidden state $\mathbf{h}^{(t)}$ into 1) a 7D vector representing correction to the head pose: 4D quaternion, which is then converted to a rotation matrix $\boldsymbol{\delta R}^{(t)} \in \mathbb{R}^{3\times3}$, and 3D translation $\boldsymbol{\delta t}^{(t)} \in \mathbb{R}^3$, and 2) correction to the position map $\boldsymbol{\delta}\mathbf{M}^{(t)}$. To compute the updates to the head pose, $\mathbf{h}^{(t)}$ is down-sampled with 3 2-stride convolutions, followed by a global average pooling and two fully-connected layers. Updates to the position map is processed by a standard stacked hourglass network [Newell et al. 2016].

4.4 Texture Inference

The goal of the texture inference is to predict the UV-space albedo map $\mathbf{A}$, specular map $\mathbf{S}$ and displacement map $\mathbf{D}$ from the input images and the inferred geometry. Being able to predict geometry in the UV space, our formulation offers a direct benefit to the texture inference module, as the pixel-aligned signals between the UV space and the multi-view inputs are already prepared in the previous geometry inference step. The high resolution texture maps are inferred based on the image texture features reprojected back to the UV space. Given the coordinates $\mathbf{u}$ in the UV space, the multi-view camera poses $\{\mathbf{P}_i\}_{i=1}^{K}$, the inferred position map $\mathbf{M}^{(T)}$, and the inferred head pose $[\mathbf{R}, \mathbf{t}]^{(T)}$, the pixel-aligned features for each view can be obtained as:

$$\tilde{\mathbf{y}}_i^{\text{text}}(\mathbf{u}) = \mathbf{f}_i^{\text{text}}\left(\mathbf{\Pi}_{\mathbf{P}_i}\left(\mathbf{R}^{(T)}\mathbf{M}^{(T)}(\mathbf{u}) + \mathbf{t}^{(T)}\right)\right), \tag{8}$$

where $\mathbf{f}^{\text{text}}$ is the texture feature map introduced in Section 4.2. We index the feature map using bilinear sampling for non-integer coordinates. Similar to our face geometry module, we aggregate the UV-space features with the aggregation function:

$$\mathbf{y}^{\text{text}}(\mathbf{u}) = \sigma\left(\tilde{\mathbf{y}}_1^{\text{text}}(\mathbf{u}), \ldots, \tilde{\mathbf{y}}_K^{\text{text}}(\mathbf{u})\right), \tag{9}$$

where σ is the aggregation function that aggregates the pixel-wise feature across all views, which could be max, min, var, *etc.* Once the reprojected feature is obtained, three independent decoders regress $\mathbf{A}, \mathbf{S}, \mathbf{D}$ simultaneously in the UV space in a coarse-to-fine fashion. Specifically, we first employ stacked hourglass networks to regress the diffuse albedo, specular and displacement maps in 512 × 512 resolution. We then use three consecutive image upsampling networks [Wang et al. 2018b] to upsample the texture maps to 1024 × 1024, 2048 × 2048, and 4096 × 4096, respectively. For diffuse albedo networks, tanh is used as the activation function, while we do not add any activation functions for the specular networks and displacement networks. The distribution discrepancy is large for different texture map representations, although they are defined in the same UV space. Thus the network parameters for the decoders are not shared for different map representations. In order to produce sharp and high-fidelity textures, we follow [Wang et al. 2018a; Yamaguchi et al. 2018] to use an adversarial loss in addition to the reconstruction loss for the training of the texture reconstruction.

4.5 Training Loss

The training of the face geometry is supervised using the ground truth head pose $[\mathbf{R}, \mathbf{t}]^{\text{gt}}$ and position map $\mathbf{M}^{\text{gt}}$. Both is supervised with a L_1 loss between the prediction and the ground truth, which is summed over all iterations. For the head pose, we compute the loss function as:

$$L_{\mathbf{P}} = \sum_t \left\|\mathbf{R}^{(t)} - \mathbf{R}^{\text{gt}}\right\|_1 + \left\|\mathbf{t}^{(t)} - \mathbf{t}^{\text{gt}}\right\|_1 .$$

For the position map, we supervise the network with a dense L_1 loss computed between the predicted position map and the ground truth after applying the corresponding head pose transformation:

$$L_{\mathbf{M}} = \sum_t \sum_{\mathbf{u}} \left\|\mathbf{R}^{(t)}\mathbf{M}^{(t)}(\mathbf{u}) + \mathbf{t}^{(t)} - \mathbf{R}^{\text{gt}}\mathbf{M}^{\text{gt}}(\mathbf{u}) - \mathbf{t}^{\text{gt}}\right\|_1 .$$

In order to learn accurate and photo-realistic textures, we supervise our texture inference network with L_1 and adversarial losses (adv) on all texture maps including $\mathbf{A}$, $\mathbf{S}$, and $\mathbf{D}$:

$$L_{\text{t}} = \lambda_{\text{adv}} \sum_{\mathbf{T} \in \{\mathbf{A},\mathbf{S},\mathbf{D}\}} \text{adv}\left(\mathbf{T}, \mathbf{T}^{\text{gt}}\right) + \sum_{\mathbf{u}} \left\|\mathbf{T}(\mathbf{u}) - \mathbf{T}^{\text{gt}}(\mathbf{u})\right\|_1 .$$

Overall, we jointly train all modules using a multi-task loss:

$$L = \lambda_{\mathbf{P}} L_{\mathbf{P}} + \lambda_{\mathbf{M}} L_{\mathbf{M}} + \lambda_{\text{t}} L_{\text{t}}.$$

5 IMPLEMENTATION DETAILS

Our system is fully implemented in PyTorch. All the training and testing is performed on NVIDIA V100 graphics cards. All the network parameters are randomly initialized and are trained using the Adam optimizer for 200,000 iterations with a learning rate set to 3×10^{-4}. For the recurrent face geometry optimizer, we set the inference step to $T = 10$, the grid resolution to $r = 3$, the search radius to $c = 1$mm, and the loss weights of the head pose ($\lambda_{\mathbf{P}}$) and the position map ($\lambda_{\mathbf{M}}$) to 0.1 and 1, respectively. For the texture inference network, we use three separate discriminators for $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{D}$. The loss weight of the L1 (λ_{t}) and discriminators (λ_{adv}) are set to 1 and 0.01, respectively. The dimensions for the input image H, W and the UV maps H_t, W_t are set to be 512. During training, we randomly select 8 camera views for each scan. We found it sufficient to train the network without data augmentation. The training process takes approximately 30 hours using 4 graphics cards. During inference, arbitrary numbers of camera views can be used as input since our view aggregation function is not sensitive to the number of views. For inference with 8 camera views, our network consumes approximately 2GB of GPU memory.

6 RESULTS

Figure 6 shows the rendered results using the complete set of assets produced by our system from randomly selected testing data, including the input reference images, the directly inferred texture maps, and the renderings under different illuminations. In addition, Figure 7 shows a detailed visualization of the inferred high-resolution texture maps: diffuse albedo, specular and displacement map. All results are rendered using the reconstructed geometries and texture maps with Maya Arnold using a physically-based shader under environment illumination provided by HDRI images.

In the following sections, we provide comparative evaluation to directly related baseline methods (Section 6.1) as well as an ablation study (Section 6.2). In addition, we also demonstrate three meaningful applications that ReFA enables in Section 6.3.

To quantitatively evaluate the geometry reconstruction, we first convert our inferred position map to a mesh representation as described in section 4.1. We then compute the scan-to-mesh errors using a method that follow [Li et al. 2021], with the exception that the errors are computed on a full face region including the ears. We

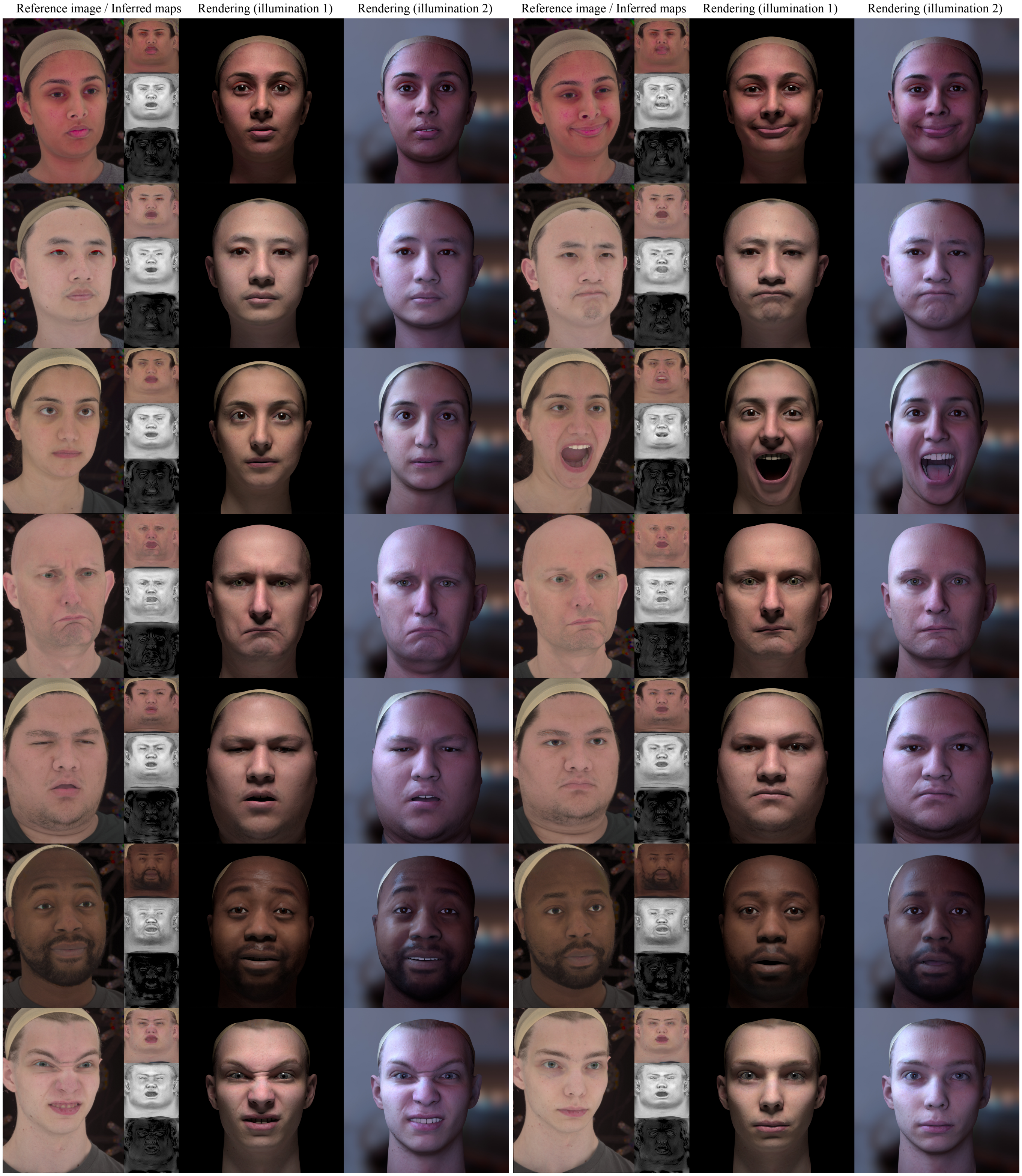

Reference image / Inferred maps
Rendering (illumination 1)
Rendering (illumination 2)
Reference image / Inferred maps
Rendering (illumination 1)
Rendering (illumination 2)

Fig. 6. Images rendered from our reconstructed face assets. Geometry constructed from the input images and the inferred appearance maps are used in the physical renderings with Maya Arnold under an lighting environments provided HDRI images. The renderings achieve photo-realistic quality that faithfully recovers the appearance and expression captured in the input photos.

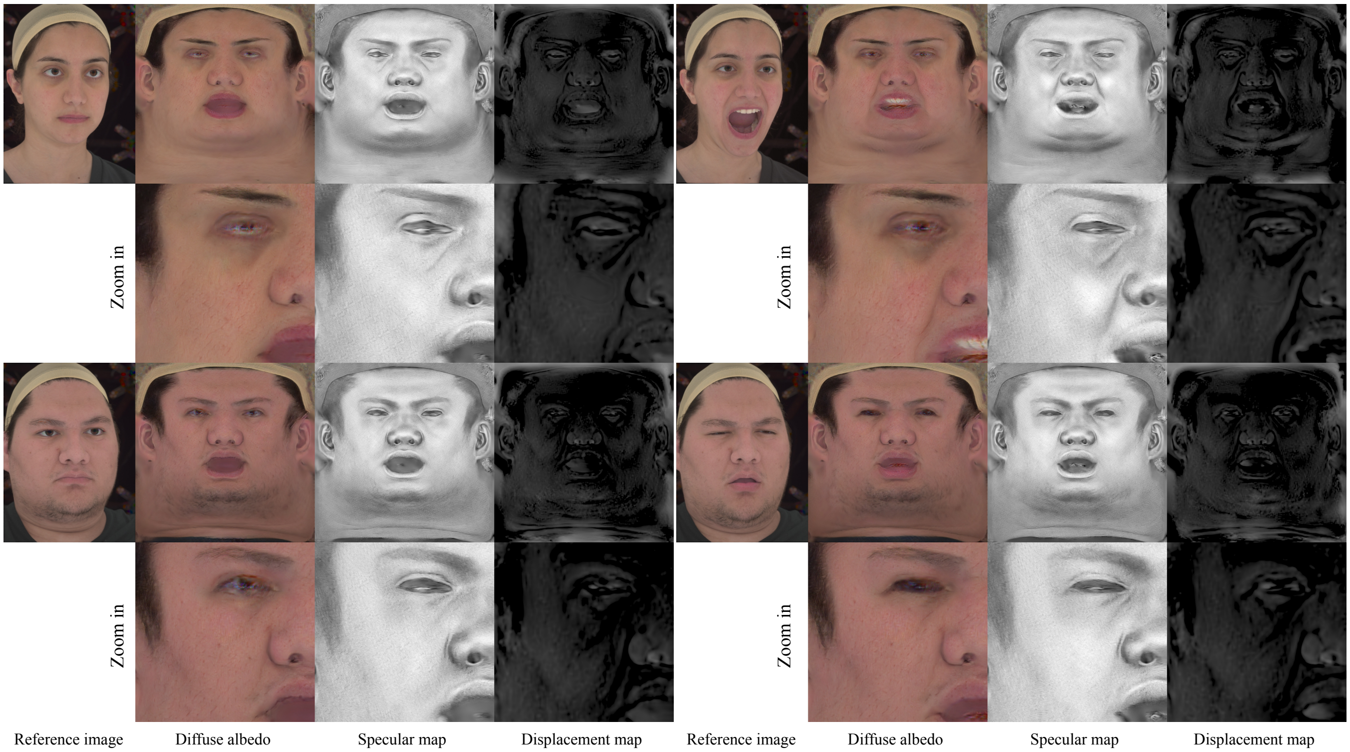


Fig. 7. Detailed results for the texture map inference. The even rows display the zoom-in images of the 4096 × 4096 texture maps. Our texture inference network constructs texture maps from the multi-view images with high-frequency details that essentially allow for photo-realistic renderings of the face assets.

measure both the mean and median errors as the main evaluation metrics, given that the two statistics capture the overall accuracy of the reconstruction models. To better quantize the errors for analysis, we additionally show the Cumulative Density Function (CDF) curves of the errors, which measure the percentage of point errors that falls into a given error threshold.

6.1 Comparisons

Baselines. We compare our method regarding to the geometry accuracy with 3 strong deep learning-based baselines from 3 different categories: (1) topologically consistent face inference network DFNRMVS [Bai et al. 2020] from a sequence of images; (2) MVS networks DPSNet [Im et al. 2018] that is a representative depth estimation network that achieved state-of-the-art results on several MVS benchmarks; (3) topologically consistent multi-view face reconstruction network ToFu [Li et al. 2021] that most resembles our setting and, prior to our work, achieved state-of-the-art result on neural-based face reconstruction. The baseline results were obtained with their publicly released codes with the 15-view input from our prepared dataset (discussed in Section 3).

Qualitative Comparisons. Figure 8 provides a visual comparison of the reconstructed geometries between our method and the baselines. Visual inspection suffices to show the qualitative improvements brought by our method. First, for a certain examples (3rd row, 5th row, 6th row), our reconstructed faces faithfully resemble the ground truth appearances whereas the model-based methods (DFNRMVS, ToFu) display apparent errors in the overall facial shape and specific parts (eye size, jaw shape). Second, our reconstruction is more robust to challenging expressions: mouth funnel (1st row), cheek raising (4th row), lip stretching (7th row) and asymmetrical and non-linear mouth movement - mouth stretching to one side (5th row). Third, as we focus on a full-face reconstruction, we note that certain methods (DFNRMVS, ToFu) fail in reconstructing the ears of the subjects, whereas ours correctly infer the shape of the ears as seen from the input images. Last but not least, our results shows the best geometry details, as our method captures the middle-frequency details where others fail, such as the wrinkle on the forehead of the 2nd and 4th row and the dimple on the face of the 2nd and 7th row.

In Figure 10, we show the comparison between our method and the traditional face reconstruction and registration pipeline (described in Section 3) given challenging inputs. In these cases of occlusion and noise, for example, due to hair occulusion (upper case) and the specific eye pose (lower case), the traditional pipeline struggles to either reconstruct the accurate face shape or fit the template mesh connectivity to the correct expressions. In practice, the raw reconstruction from MVS algorithm contains a certain geometry noise, which requires manual clean-up by professional artists to remove the errors before the registration step. In contrast, despite not trained with these examples, our network manages to infer the correct shape automatically. We believe that this is attributed to

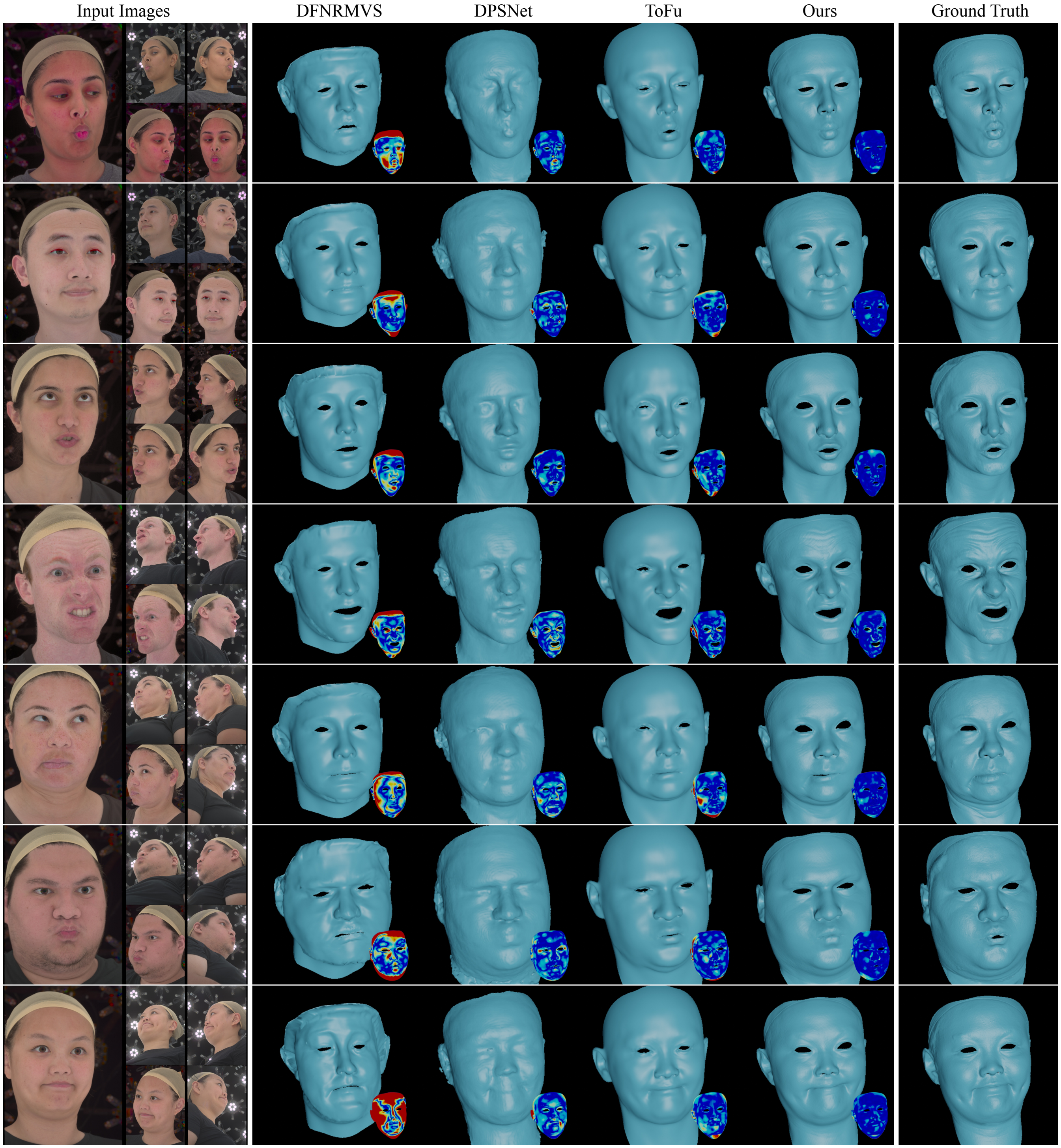


Fig. 8. Qualitative comparison with the baselines on our testing dataset. As the release codes of the baseline methods [Im et al. 2018] and [Li et al. 2021] do not produce appearance maps, the results presented here are the network direct output geometry rendered with a basic shader using Blender. Visual inspection suffices to reveal the improvement our model has achieved: ReFA produces results that are more robust to challenging expressions (row 1,4,5), facial shapes (row 6,7) and reconstructs a wider face area including the ears and forehead when compared to [Li et al. 2021] and [Bai et al. 2020].

Table 2. Quantitative comparison on our Light stage captured dataset. The table measures the percentage of points that are within Xmm to the ground truth surface (column 1-3), mean and median scan-to-mesh errors (column 4-5), and a comparison of the supported features (column 6-8). "Consistency" denotes whether the reconstructed has consistent mesh connectivity. "Dense" denotes whether the model reconstructs geometry of more than 500k vertices [Ma et al. 2008], and "Texture" denotes whether the network output includes texture information. Although the original work of ToFU includes texture inference, the module is separate from its main architecture and thus not learned end-to-end.

	< 0.2mm(%)	< 1mm(%)	< 2mm(%)	Mean (mm)	Med. (mm)	Consistency	Dense	Texture
DFNRMVS [Bai et al. 2020]	5.266	25.900	48.345	2.817	2.084	✓	✓	✗
DPSNet [Im et al. 2018]	12.645	55.042	82.171	1.197	0.882	✗	✓	✗
ToFu [Li et al. 2021]	15.245	61.493	83.162	1.273	0.742	✓	✗	✗*
ReFA (Ours)	**18.382**	**70.547**	**91.605**	**0.848**	**0.603**	✓	✓	✓

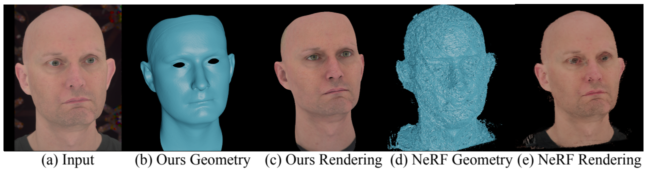


Fig. 9. Qualitative comparison with NeRF-based method.

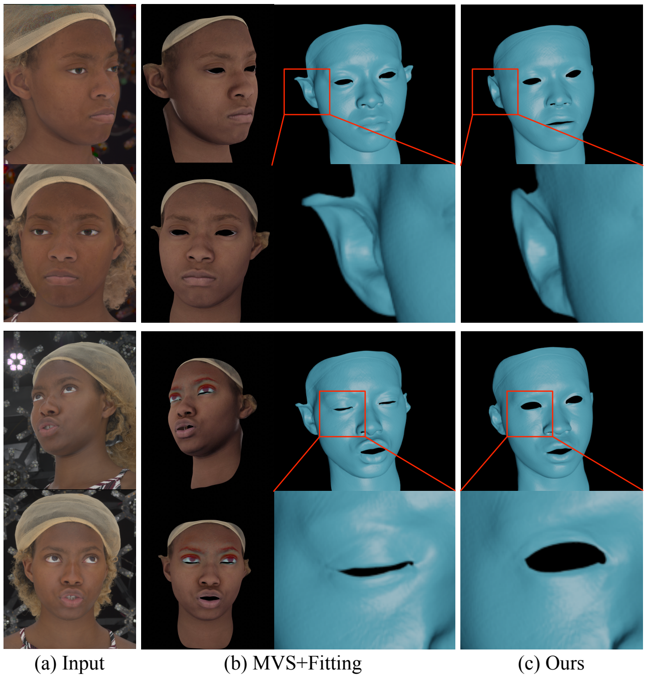


Fig. 10. A comparison between our method (c) and a traditional MVS and fitting pipeline (b). The traditional pipeline incorrectly reconstructs the two challenging input examples as shown in the figure: pointy ear in the upper case due to hair occlusion and closed eyes in the lower case. Our system not only correctly reconstructs the fine geometry details, but also at a significantly faster speed.

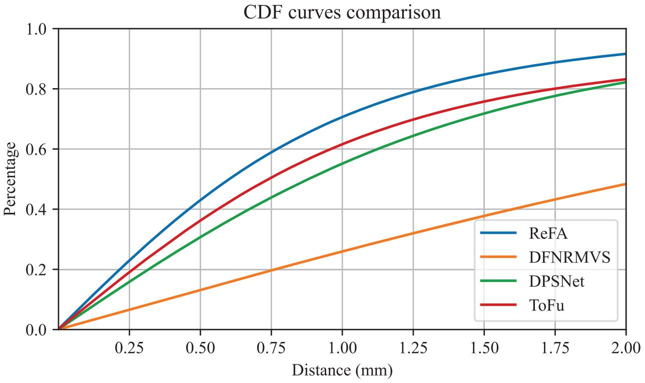


Fig. 11. Cumulative density function (CDF) curves of scan-to-mesh distance comparison on our testing dataset.

Table 3. Inference Time comparison. Our model is both more efficient and more effective to the baselines at a 4.5FPS speed. A lighter model that achieves similar accuracy to ToFu runs at 9FPS, which is close to real-time performance.

Method	Time (s)	FPS	Med. (mm)
Traditional	≈30min	-	-
DFNRMVS [Bai et al. 2020]	4.5s	-	2.084
ToFu [Li et al. 2021]	0.39s	2.6	0.742
Ours	0.22s	4.5	0.603
Ours (small)	0.11s	9	0.763

the learned data prior such as face semantic information from the training dataset. This validates that our system is more robust than the traditional pipeline in challenging and noisy situations.

In Figure 9, we also compare our method with NeRF-based method, which is a recent stream of image-based rendering (IBR) works. The NeRF baseline is an implementation from [Müller et al. 2022]. With 15 views, NeRF fails to produce production-ready geometries due to the lack of visual correspondences and facial semantics, despite the decent rendered results, while our method achieves superior qualities in geometry reconstruction. In addition, our inferred maps are amenable to renderings in different environments using established physically-based render engines.

Quantitative Comparisons. Table 2 and Figure 11 show our quantitative comparison and CDF curve comparison with the baseline methods on our test dataset, respectively. ReFA outperforms the baselines in all metrics. Remarkably, ReFA achieves a median error of 0.603mm, which outperforms the strongest baseline by 19%. In terms of recall, we observe that our model brings the best improvement in high-precision range, covering 20.6% more points within the 0.2mm precision when compared to the best baseline, and 14.7% and

Table 4. Ablation study on our dataset. Underlined items are our default settings. Correlation feature: whether use as default ("correlation") or simply concatenate the semantic and visual features ("concat"). View aggr func.: choice of the pooling function. Grid size: the total size length of the 3D grid built for computing correlation. Search radius: the search radius in computing the visual-semantic correlation. Recurrent Layer: whether GRU is used or is replaced by convolution layers. UV-space Feature: components of the UV-space features: UV coordinates (U), position map (P), and face region map (R). UV-space Embedding: whether the UV-space feature g is learned by a neural network ("Network") or directly set as learnable parameters ("Parameter"). Input View: number of views used as input in the inference. Notably, decreasing the number of views for the inference only results in a slight decrease in performance. Our model's performance with only 4 views still achieve the best accuracy when compared to the best baseline that utilizes 15 views.

	Method	Mean (mm)	Med. (mm)
Correlation Feature	Correlation	0.848	0.603
	Concat	0.990	0.713
View Aggr. Func.	Max	0.848	0.603
	Mean	1.083	0.827
	Var	1.169	0.865
Search Radius	1mm	0.869	0.615
	3mm	0.848	0.603
	5mm	0.901	0.640
UV-space Resolution	512	0.848	0.603
	256	0.872	0.620
	128	0.966	0.668
Recurrent Layer	GRU	0.848	0.603
	Conv.	0.880	0.623
UV-space Feature	U+P+R	0.848	0.603
	U+P	1.225	0.918
	U	1.246	0.951
UV-space Embedding	Network	0.848	0.603
	Parameter	0.935	0.692
Input View	15	0.848	0.603
	8	0.901	0.632
	6	0.930	0.652
	4	1.014	0.720

10.1% in the 1mm and 2mm thresholds, respectively. The increased accuracy of our model is augmented with the fact that, to our knowledge, our model is the only neural-based face asset creation system that simultaneously generates consistent, dense and textured assets in an end-to-end manner (see the right panel in Table 2).

Besides the accuracy, our system also runs significantly faster than previous works. We show the inference time comparison in Table 3 and Figure 13. The traditional method takes hours to process a single frame of a multi-view capture. Despite being accurate, the time consumption becomes tremendous for processing large scale data and performance sequences. Compared to previous deep learning-based works, our system achieves significantly better accuracy while being 40% faster.

To draw a controlled comparison for showing the speed improvement, we have designed a smaller model by slightly modifying our network by: (1) using a light-weight feature extraction network; (2) reducing the searching grid dimension from $r = 3$ to $r = 2$; (3) reducing the UV space resolution to 128 × 128. This smaller model achieves similar accuracy and model resolution as the previous state-of-the-art method [Li et al. 2021], while achieving an inference speed of 9 frames per second (FPS).

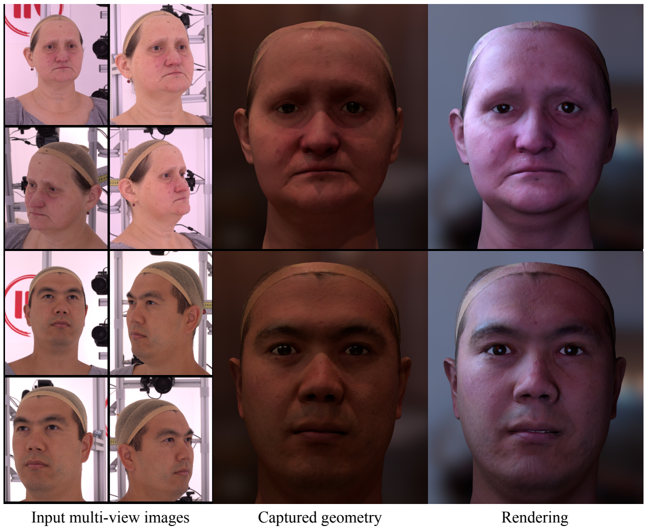


Fig. 12. Testing result on the Triple Gangers [Triplegangers 2021] dataset, whose capturing setup contains different camera placements, no polarizer and a lighting condition that is not seen in our training dataset. The result demonstrated here shows that our model generalizes well to unseen multi-view dataset that has been captured in different settings.

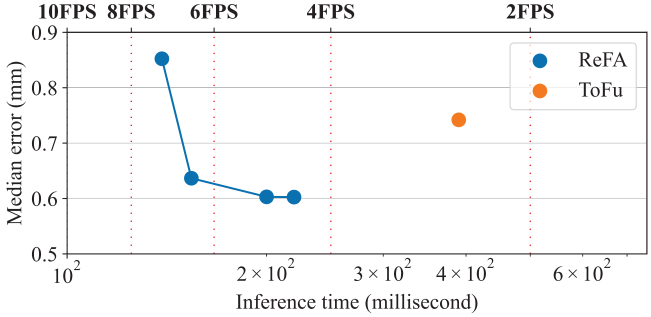


Fig. 13. Accuracy-time graph against varying numbers of the inference iteration for ReFA.

We have also quantitatively evaluated the accuracy of the head pose and the quality of the inferred albedo maps. The estimated head poses are found to have a rotation error of 1.429°(mean) and 1.255°(median) and a translation error of 2.91mm (mean) and 2.70mm (median). The inferred albedo maps, when compared to the ground truths, have a mean PSNR of 28.29dB and a mean SSIM of 0.75.

Generalization. Our model is tested on the Triple Gangers [Triplegangers 2021] dataset in Figure. 12, which contains different illumination and camera placements from our training set. In this generalized setting, our model produces high reconstruction quality comparable to the results on our testing set. In particular, our model is able to infer the displacement and specular maps, which would not have been accessible using the traditional approach, due to the hardware limits in its capturing system (without the use of polarizers, it is hard to separate the skin reflectance). In addition, our model also achieves a 0.89mm median error on images with random light sources (shadows), which indicates that our model can generalize to different lighting conditions to certain extent.

6.2 Ablation Study

To validate our design choices, we conduct extensive ablation studies by altering our key building modules, including 1) the choices of the correlation feature and the learned embedding feature, 2) the choices of view aggregation function and the search radius in computing the visual-semantic correlation, 3) the resolution of the UV map 4) removal of GRU in recurrent layer and 5) the number of input views. The detailed statistics of the ablation study is shown in Table 4.

Correlation features. Based on designs experimented in prior works [Im et al. 2018; Teed and Deng 2020], we have altered the correlation feature by directly concatenating the UV features and the multi-view image features ("concat" in the first row of Table 4), instead of computing their similarity through an inner product. This change has increased training difficulty and is shown to be less effective in the position matching task.

View aggregation function. Three different functions for fusing features across view, max pooling, mean pooling and variance pooling, are investigated and the results are shown in the second row of Table 4. A little to our surprise, the max aggregation function performs significantly better when compared to the others, although mean pooling is commonly utilized in other multi-view frameworks (e.g. [Im et al. 2018]). We speculate that the ability of max pooling to attend to the most salient views allows it to discard noise. The behavior also suggests a difference between our formulation of visual-semantic correlation features and the typical MVS network features, which are typically based solely on visual similarity. Notably, max pooling is also more robust to scenarios where regions of the face are occluded in a certain views.

Search radius in VSC. According to the result, our model achieves the best performance at a reasonable search radius of 3mm. We believe this is because a smaller search radius requires more update steps, while a larger search radius leads to less precision.

UV-space resolution. By default, we set our UV-space resolution of the position map to be 512 × 512, which is equivalent to a dense mesh of approximately 260K faces. In many practical situations, inference speed may be preferred over precision. We thus investigate the effectiveness of our system under various UV-space resolutions, including 512 × 512, 256 × 256 (≈65K faces), and 128 × 128 (≈16K faces). From the results we can observe that decreasing the UV-space resolution slightly decreases the performance. However, even under lowest UV-space resolution, our model still outperforms the baseline, as a 128 × 128 position map still encodes a denser mesh when compared to the parametric models utilized by the baselines (e.g. [Li et al. 2021]). This validates the position map as a more advantageous representation in the face asset inference task.

Recurrent layer. We investigate the effectiveness of the gated recurrent unit (GRU) module. GRU layers take the input of the current step and the hidden layer from the previous step as inputs, and outputs the updated hidden layer. We replace the GRU layers with convolution layers. The results show that GRU performs significantly better than convolution. This shows that GRU layers can better capture the long-term memory.

UV features. We remove different components of the UV features, including UV coordinates (U), position map (P), region map (R) (see "UV-space Feature" row of Table 4) and found that removing any component decreases the performance, where removing the region map causes the largest decrease. We speculate that this is because the region map, when compared to the UV coordinate feature, explicitly encodes the semantics. In addition, we have tried to directly use an embedding tensor as a learnable parameter for the UV features ("Parameter" in the "UV-space Embedding" row of Table 4) instead of filtering the input UV maps with a neural network ("Network" in the "UV-space Embedding" row of Table 4). This drastically increases the number of parameters that need to be trained without regularization, the performance under such design decreases slightly. To better understand the effectiveness of the learned embedding network, we visualize the UV features of both designs with T-SNE [Van der Maaten and Hinton 2008] in Figure 15. We can observe that both feature maps exhibit symmetry property and regional distribution that complies the actual face region. However, a learned embedding produces much better regularized feature maps, which validates the effectiveness of our UV-space embedding network.

Number of views. Decreasing the number of capture views needed to faithfully reconstruct a face asset is essential for supporting a light-weighted capturing system. The benefits from fewer view come in twofold: (1) obviously, the capturing system needs fewer cameras; (2) the saving on storage space needed for the raw data is quite significant - approximately 2 terabytes of storage space can be free if only half of the 16 cameras are needed for a 10 minutes video of a subject. The "Input View" row of Table 4 shows our model's performance given different number of views as input. It is noteworthy that while the performance decreases the less the available views, reducing 16 views to 6 views only resulted in a 9% increase in median error and the achieved precision is enough for use in a professional setting (<1mm). Reducing to 4 views comes with a 20% increase in median error. However, even the 4-view reconstruction with our model outperforms all compared baselines that utilize 15-view input (see Table 2). We believe that the results encourages a practical solution to reconstructing face assets from sparse views, where a traditional acquisition algorithm would struggle.

6.3 Applications

Avatar creation. We show animation sequences in the **supplementary video** using the avatar directly generated from our pipeline without any manual tweaking.

Performance capture. Fast geometry inference is a notable strength of our model. In particular our small model (128 × 128 resolution) achieves a close-to-real-time performance at 9FPS. The efficiency

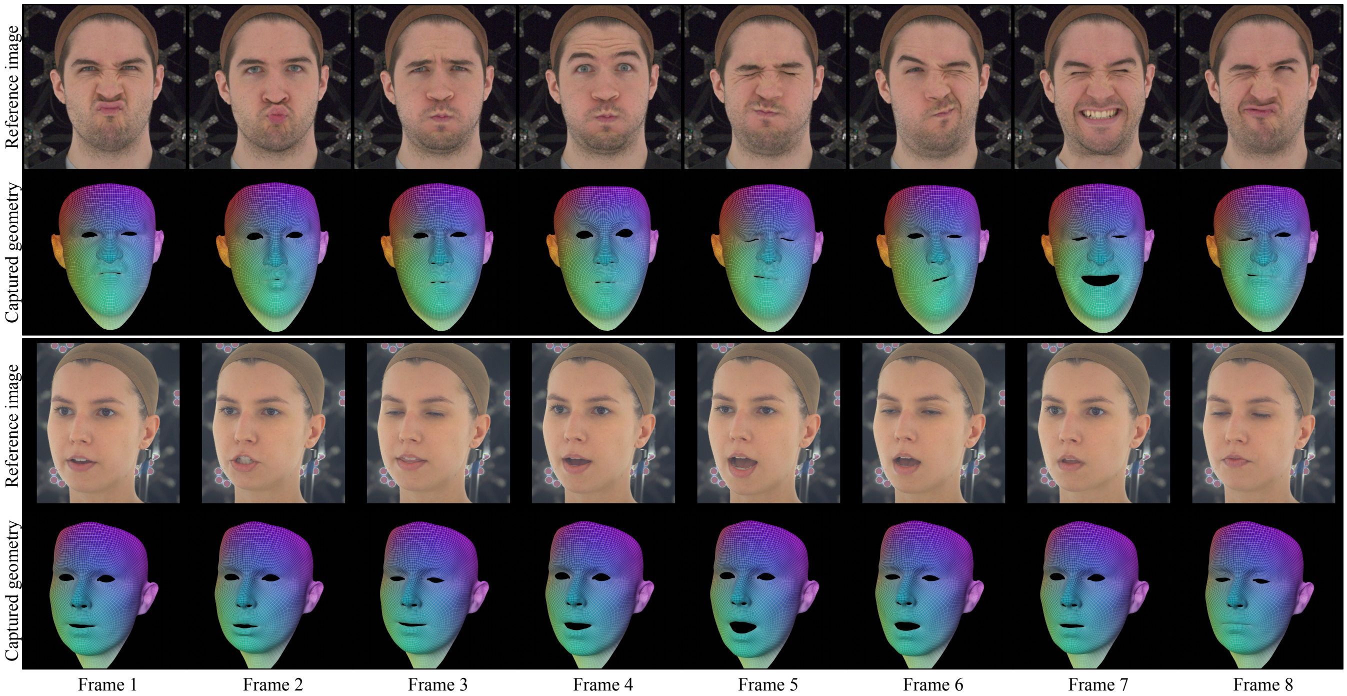
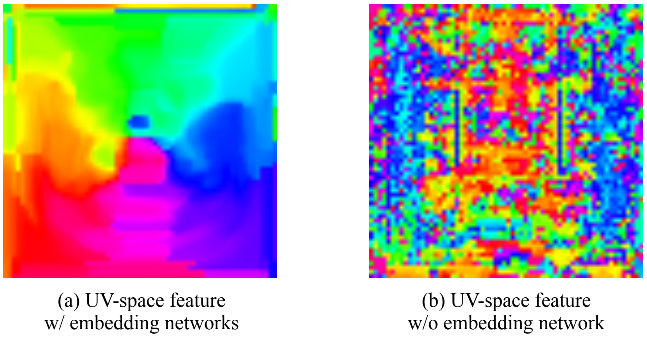


Fig. 14. Reconstruction of a video sequence at 4.5FPS, where the expression and the head pose of the subject changes over time.


(a) UV-space feature w/ embedding networks

(b) UV-space feature w/o embedding network

Fig. 15. Ablation study on the UV-space embedding network: (a) using our proposed UV-space features along with a neural network; (b) directly setting the UV-space feature as a learnable parameter. To visualize the features, we use T-SNE [Van der Maaten and Hinton 2008] to embed the feature to the 3-dimension space.

reveals the potential application of our model in neural-based performance capture. We thus demonstrate two dynamic sequences in fig. 14, where only 8 camera views are used. Both the input images and the reconstructed meshes (converted from inferred output) are visualized. Importantly, the reconstructed meshes are color-mapped by the UV coordinates. The accurate reconstruction together with the color mapping demonstrates that our system is capable of capturing accurate face geometry from a video sequence while maintaining correspondences across the captured shapes. We believe that this application is a showcase of the readiness of our system for performance capture in the digital industry. For more details, please watch the **supplementary video**.

Extended Representation. The UV-space position map offers significant flexibility in supporting various types of output. Figure 16 offers different use cases, where the same position map is shown converted to meshes of various densities, point cloud, landmarks and region segmentation maps. Specifically, our position map can be converted to different mesh topologies seamlessly as long as a solid UV mapping is provided. It thus opens up potential use in Level of Detail (LOD) rendering, where faces of different detail levels are needed in real-time applications. In addition, by choosing specific points in the UV space, point cloud, landmarks and region maps representations can be extracted from the position maps.

7 CONCLUSION

We present an end-to-end neural face capturing framework, ReFA, that effectively and efficiently infers dense, topologically consistent face geometry and high-resolution texture maps that are ready for production use and animation. Our model tackles the challenging problem of multi-view face asset inference by utilizing a novel geometry representation, the UV-space position map, and a recurrent face geometry optimizer that iteratively refines the shape and pose of the face through an alignment between the input multi-view images and the UV-space features. Experiment results have demonstrated that our design choices allow ReFA to improve upon previous neural-based methods and achieve the state-of-the-art results in accuracy, speed and completeness of the shape inference. In addition, our

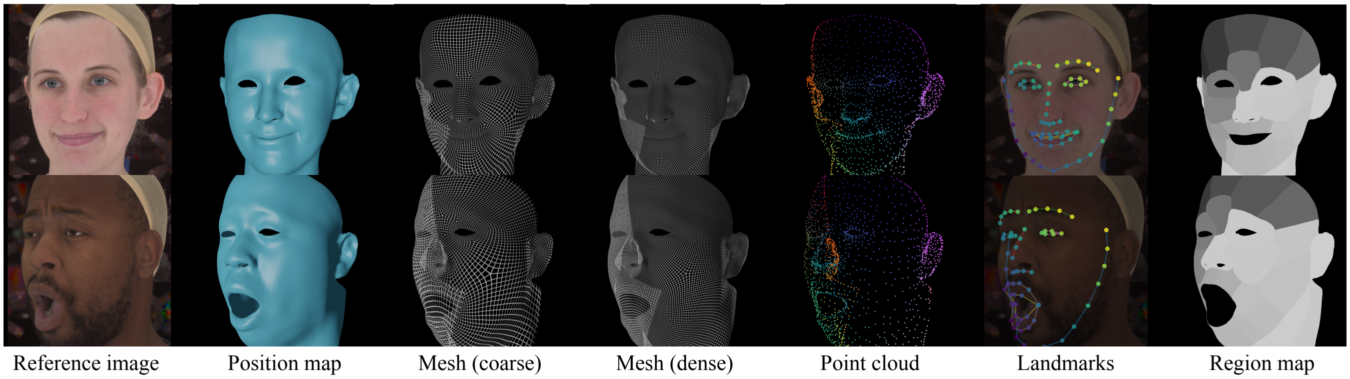


Fig. 16. Given valid UV mappings, the position map representation is amenable to conversions to various representations, as shown in each column. This include 3D meshes of different subdivisions, which enables Level of Detail (LOD) rendering; point cloud, landmarks and region map representation that are commonly used in mobile applications.

model is shown to be device-agnostic to various capture settings, including sparse views and views under different lighting conditions, with little compromise taken on its performance. We believe that the progress we make opens up ample opportunities for rapid and easily accessible face acquisition that meets the high demand for face asset capturing in the digital industry.

Future work. Our current network is not originally designed for performance capture, as it is trained with a database consisting of static scans. A natural future step of this work is to extend our design to specifically process video sequences for performance capturing. We believe that features specifically designed for temporal integration may enhance the speed and temporal consistency based on what our current framework can achieve. Another meaningful direction to explore is to extend our approach to a single-view setting, or the more challenging setting where the input is in-the-wild. As occlusion, shadows, and noise may become a major obstacle in limiting the performance of a single-view reconstruction algorithm, we believe that leveraging additional priors, such as symmetry assumption on the face, may be a meaningful direction to explore.

ACKNOWLEDGMENTS

This research is sponsored by the Department of the Navy, Office of Naval Research under contract number N00014-21-S-SN03 and in part sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005. Army Research Office also sponsored this research under Cooperative Agreement Number W911NF-20-2-0053. Statements and opinions expressed, and content included, do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Further, the views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

Jens Ackermann and Michael Goesele. 2015. A survey of photometric stereo techniques. *Foundations and Trends® in Computer Graphics and Vision* 9, 3-4 (2015), 149–254.

Brian Amberg, Reinhard Knothe, and Thomas Vetter. 2008. Expression invariant 3D face recognition with a morphable model. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*. IEEE, 1–6.

Ziqian Bai, Zhaopeng Cui, Jamal Ahmed Rahim, Xiaoming Liu, and Ping Tan. 2020. Deep facial non-rigid multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5850–5860.

Anil Bas and William AP Smith. 2019. What does 2D geometric information really tell us about 3D face shape? *International Journal of Computer Vision* 127, 10 (2019), 1455–1473.

Thabo Beeler, Bernd Bickel, Paul A. Beardsley, Bob Sumner, and Markus H. Gross. 2010. High-quality single-shot capture of facial geometry. In *ACM Transactions on Graphics (TOG)*.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W Sumner, and Markus Gross. 2011. High-quality passive facial performance capture using anchor frames. In *ACM SIGGRAPH 2011 papers*. 1–10.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 187–194.

Volker Blanz and Thomas Vetter. 2003. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence* 25, 9 (2003), 1063–1074.

Michael Bleyer, Christoph Rhemann, and Carsten Rother. 2011. Patchmatch stereo-stereo matching with slanted support windows.. In *Bmvc*, Vol. 11. 1–11.

Timo Bolkart and Stefanie Wuhrer. 2015. A groupwise multilinear correspondence optimization for 3d faces. In *Proceedings of the IEEE international conference on computer vision*. 3604–3612.

George Borshukov, Dan Piponi, Oystein Larsen, John P Lewis, and Christina Tempelaar-Lietz. 2005. Universal capture-image-based facial animation for" The Matrix Reloaded". In *ACM Siggraph 2005 Courses*. 16–es.

Jia-Ren Chang and Yong-Sheng Chen. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5410–5418.

Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. 2019. Photo-realistic facial details synthesis from single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9429–9439.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 145–156.

Paul Ekman and Wallace V. Friesen. 1978. Facial action coding system: a technique for the measurement of facial movement. In *Consulting Psychologists Press*.

Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.

Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. 2018. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 534–551.

Graham Fyffe, Koki Nagano, Loc Huynh, Shunsuke Saito, Jay Busch, Andrew Jones, Hao Li, and Paul Debevec. 2017. Multi-View Stereo on Consistent Face Topology. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 295–309.

David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. 2007. Real-time plane-sweeping stereo with multiple sweeping directions. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of personalized 3D face rigs from monocular video. *ACM Transactions on Graphics (TOG)* 35, 3 (2016), 1–15.

Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. 2018. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8377–8386.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul Debevec. 2011a. Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–10.

Abhijeet Ghosh, Graham Fyffe, Borom Tunwattanapong, Jay Busch, Xueming Yu, and Paul E. Debevec. 2011b. Multiview face capture using polarized spherical gradient illumination. *ACM Transactions on Graphics (TOG)* 30 (2011), 129.

Paul Graham, Borom Tunwattanapong, Jay Busch, Xueming Yu, Andrew Jones, Paul Debevec, and Abhijeet Ghosh. 2013. Measurement-based synthesis of facial microgeometry. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 335–344.

Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. 2020. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2495–2504.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. 2018. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2821–2830.

Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. 2018. DPSNet: End-to-end Deep Plane Sweep Stereo. In *International Conference on Learning Representations*.

Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. 2001. Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 1. IEEE, I–I.

Andor Kollar. 2019. Realistic Human Eye. http://kollarandor.com/gallery/3d-human-eye/. Online; Accessed: 2022-3-30.

Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. 2020. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. 2020. AvatarMe: Realistically Renderable 3D Facial Reconstruction" In-the-Wild". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 760–769.

Chloe LeGendre, Kalle Bladin, Bipin Kishore, Xinglei Ren, Xueming Yu, and Paul Debevec. 2018. Efficient multispectral facial capture with monochrome cameras. In *Color and Imaging Conference*.

Martin D Levine and Yingfeng Chris Yu. 2009. State-of-the-art of 3D facial reconstruction methods for face recognition based on a single 2D training image per person. *Pattern Recognition Letters* 30, 10 (2009), 908–913.

Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. 2009. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (ToG)* 28, 5 (2009), 1–10.

Jiaman Li, Zhengfei Kuang, Yajie Zhao, Mingming He, Karl Bladin, and Hao Li. 2020b. Dynamic facial asset and rig generation from a single scan. *ACM Trans. Graph.* 39, 6 (2020), 215–1.

Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. 2020a. Learning formation of physically-based face attributes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3410–3419.

Tianye Li, Shichen Liu, Timo Bolkart, Jiayi Liu, Hao Li, and Yajie Zhao. 2021. Topologically Consistent Multi-View Face Inference Using Volumetric Sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3824–3834.

Shichen Liu, Yichao Zhou, and Yajie Zhao. 2021. Vapid: A rapid vanishing point detector via learned optimizers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12859–12868.

Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Félix Chabert, Malte Weiss, and Paul E. Debevec. 2007. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Rendering Techniques*.

Wan-Chun Ma, Andrew Jones, Tim Hawkins, Jen-Yuan Chiang, and Paul Debevec. 2008. A high-resolution geometry capture system for facial performance. In *ACM SIGGRAPH 2008 talks*. 1–1.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

Elad Richardson, Matan Sela, and Ron Kimmel. 2016. 3D face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision (3DV)*. IEEE, 460–469.

Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. 2017. Learning detailed face reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1259–1268.

Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. 2019. 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*. 7763–7772.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*. Springer, 501–518.

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 175–184.

Christoph Strecha, Rik Fransens, and Luc Van Gool. 2006. Combined depth and outlier estimation in multi-view stereo. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 2394–2401.

Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*. Springer, 402–419.

Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. 2018. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2549–2559.

Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 1274–1283.

Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2387–2395.

Triplegangers. 2021. Triplegangers Face Models. https://triplegangers.com/. Online; Accessed: 2021-12-05.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018a. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8798–8807.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018b. ESRGAN: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*.

Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. 2018. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 601–610.

Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. 2019. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5525–5534.

Chao Zhang, William AP Smith, Arnaud Dessein, Nick Pears, and Hang Dai. 2016. Functional faces: Groupwise dense correspondence using functional maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5033–5041.