Learning to Reconstruct 3D Faces by Watching TV

Mengya Liu Jiezhang Cao Tianhao Li Xiyi Chen

ETH Zürich

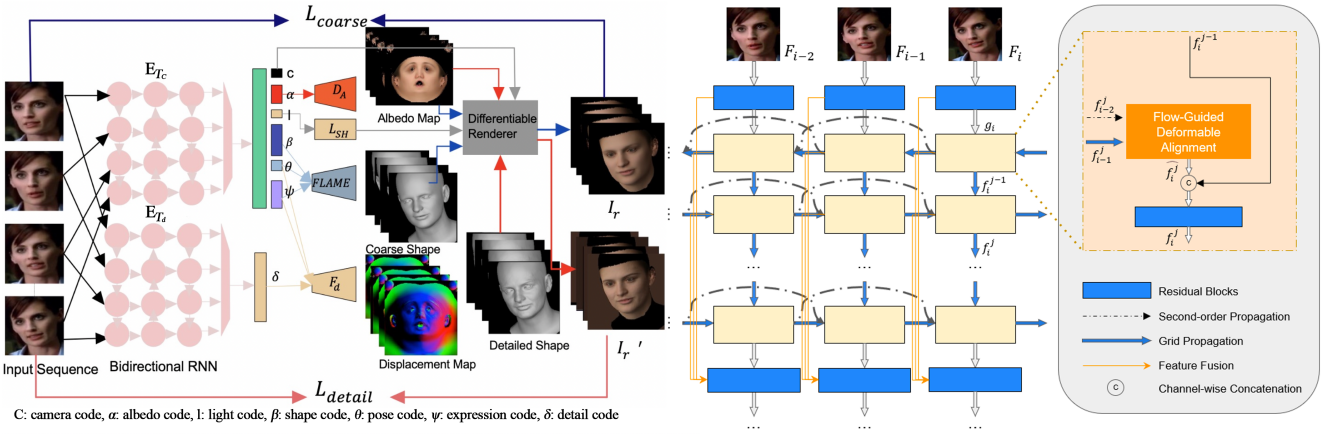

Figure 1: (Left) Our network architecture. Modified based on DECA [9], we replace DECA's encoders with temporal encoders to encode temporal information from video sequences. (Right) Details of our bidirectional-RNN temporal feature extractors with second-order grid propagation and flow-guided deformable alignment, modified based on BasicVSR++ [5].

Abstract

Most 3D face reconstruction methods have limited abilities to capture fine-scale details. Seeking to produce more accurate person-specific details, DECA produces UV displacement map, which contains parameters on both person-specific details and generic expressions, from low-dimensional latent codes regressed from input images. Although it demonstrates state-of-the-art performance on benchmark datasets, computational efficiency, and robustness to occlusions, it could still suffer from the severe ambiguity and ill-posedness of in-the-wild images. In this work, we propose to use the abundant information across frames from TV series videos. We collect a sequential face dataset of subjects with continuous movements. We modify DECA and include bidirectional RNN based temporal feature extractors to propagate and aggregate temporal inforamtion across frames. Qualitative evaluation results show that our method captures more details in facial expressions compared to DECA.

1. Introduction

3D face reconstruction is the task of reconstructing a face from a 2D image into a 3D mesh. It is a longstanding challenge yet fundamental problem in computer vision, with wide applications in face recognition [37], face alignment [10], image and video editing [13][18], image synthesis [14], and speech-driven facial animation [27].

Recent methods have been able to produce 3D meshes with fine-scale details such as expression-dependent wrinkles and characteristic features such as pores and spots [12][6][36][15][9]. Among them, DECA [9] robustly produces a UV displacement map from low-dimensional latent codes The displacement map contains parameters on both person-specific details and generic expressions. Along with the coarse shape, it continues to form a detail shape that captures detailed facial expressions such as wrinkles. Compared to previous methods on expression-dependent detail models that require detailed 3D scans as training data [34], DECA forms the loss functions to compare rendered 2D faces with inputs without utilizing 3D ground-truths. Therefore, DECA models can be easily trained on in-the-wild datasets to enable better generalization.

Although DECA demonstrates the feasibility of reconstructing a robust and high-fidelity 3D face mesh with detailed person-specific expressions for a single RGB image without any human annotations and achieves state-of-the-art performance on benchmarks datasets, it could still suffer from the severe ambiguity and ill-posedness of in-the-wild images. To tackle this problem, we utilize the abundant temporal information in videos. Due to the fact that main characters frequently show up in TV series in different scenes with variant environment conditions, we seek to improve DECA by reconstructing high-fidelity 3D face meshes from TV series sequences.

We collect a dataset of face sequences from popular TV series to allow training 3D face reconstruction models with sequential inputs. We then perform facial keypoints and face segmentation detections on the data we collect, as required for training DECA, and transfer a pre-trained DECA model to our dataset as a baseline test.

Inspired by the exploitation and aggregation of spatio-temporal information across misaligned video frames in video super-resolution methods [4][5], we replace the ResNet50-based [16] encoders in DECA with bidirectional RNN based temporal encoders with second-order grid propagation. We show in our qualitative evaluations that with the temporal information, our method could reconstruct better facial expression details than DECA.

In summary, our main contributions are: 1) Collect 3000+ valid, clear, sequential facial images from 2 TV series. 2) Preprocess the face captures to obtain facial keypoints and face segmentation masks. 3) Perform a baseline test to reproduce DECA performance on our own dataset. 4) Improve the model by using temporal encoders to extract temporal information from the input face sequences.

2. Related Works

In this section, we present a brief literature review on 3D face reconstruction and temporal feature extraction. We only review methods to recover 3D faces from monocular 2D images. For a more comprehensive overview on 3D face reconstruction and multi-view approaches, we direct readers to [38] and [8].

2.1. 3D Face Reconstruction

It has been more than two decades since the groundbreaking work [32] in 1998 that first showed how to derive 3D shape and surface texture of a human face from a single image. Many methods have followed this work and compute parameters (pose, shape, expression, etc.) by utilizing priors on geometries and appearances from precomputed 3D face models, either in optimization-based [2] or learning-based [19] way. However, these methods can only model limited shape information and produce coarse reconstructions. Although some other methods directly predict 3D faces from 2D images without a face model and could capture more shape variations, these methods require 3D ground-truth [10] or synthetic 3D faces [7] for training. Therefore, they still have limited capability on capturing fine-scale detail variations such as expression-dependent wrinkles.

To obtain more person-specific details beyond the overly-smooth coarse shapes, another group of methods aim to personalize the model and recover the missing dimensions for facial details through the following approaches: personalized reflectance maps, multi-region and sparse deformation models, reconstruction of personalized face rigs, anatomical models and physics, medium-level shape correctives, and fine-scale detail estimation, etc. [38].

We focus on fine-scale detail estimation that learns from shading cues of the input image to improve upon coarse reconstructions. Optimization-based methods, such as [28], uses shape-from-shading technique to be able to recover small surface details from monocular images. However, these methods suffer from computational complexity and the lack of robustness on occlusions. Learning-based methods apply refinements upon reconstructed coarse shapes to obtain more detailed features. These refinements include computing local wrinkles from high-resolution scans [6] and predicting displacement maps [15]. Other learning-base models learn to directly predict meshes [36] or add-on facial details [12]. However, these methods are either not robust to occlusions or have limited ability to capture enough details of facial expressions.

DECA [9] tackles these problems as the first approach to regress both 3D faces and person-specific animatable details that change with expression. It regresses various latent parameters, including a person-specific detail code from the input image and builds a UV displacement map. Combined with the FLAME-based [22] coarse shape, it reconstructs a detailed shape that incorporates person-specific fine-scale details. A novel detail-consistency loss is introduced to disentangle person-specific details from expression-dependent wrinkles to enable the synthesis of realistic person-specific wrinkles with person-specific details unchanged. DECA achieves state-of-the-art performance on benchmarks datasets for 3D face reconstruction, performs reconstruction in real time, and is robust to occlusions.

Although DECA has improved upon most of the limitations of previous works on detail reconstruction, it is still prone to suffer from the severe ambiguity and ill-posedness of in-the-wild images. In this work, we propose to utilize the abundant information in videos by extracting temporal features from variant facial expressions in consecutive frames. We review methods for temporal feature extraction in 2.2.

2.2. Temporal Feature Extraction

Temporal feature extraction is widely used in common video processing tasks such as video super-resolution (VSR), video deblurring, and frame interpolation. To extract temporal information across frames, many methods adopt recurrent frameworks. In the domain of VSR, RSDN [17] uses a uni-directional recurrent network to decompose sequence frames into components and aggregate information from current and previous frames to super-resolve each frame. BasicVSR [4] proposes to use bidirectional propagation with optical flow-based feature alignment and outperforms most of the previous works. BasicVSR++ [5] builds upon BasicVSR and introduces two modifications: second-order grid propagation and flow-guided deformable alignment. It more effectively propagates information through misaligned frames and outperforms existing state-of the-art methods while maintaining efficiency.

Inspired by the effectiveness and efficiency of feature propagation and aggregation across frames, we apply it to the domain of 3D face reconstruction and demonstrate its effectiveness for detail reconstruction on face sequences.

3. Review of DECA Reconstruction Pipeline

In this section, we provide a brief review of the DECA reconstruction pipeline. DECA applies two ResNet50-based [16] encoders E_c and E_d to encode parameters from the input image. E_c regresses 3 camera parameters (c), 50 albedo parameters (α), 27 light parameters (l), 100 FLAME shape parameters (β), 6 pose parameters (θ), 50 expression parameters (ψ). In total, it predicts a 236-dimensional latent code. These parametes are then used to calculate an albedo map, a shaded face image, and a coarse shape. E_d predicts a 128-dimensional detail code (δ) which represents subject-specific details. This derail code is then used to produce a UV displacement map.

The FLAME-based [22] coarse shape $M(\beta, \theta, \psi)$ is computed as:

$$M(\beta, \theta, \psi) = W(T_p(\beta, \theta, \psi), J(\beta), \theta, \mathcal{W}), \quad (1)$$

where W represents blend skinning function that rotates vertices T around joints J. $\mathcal{W}$ represents the blend weights. $T_p(\beta, \theta, \psi)$, denotes the template with neutral face with added shape, pose, and expression offsets, computed as:

$$T_p(\beta, \theta, \psi) = T + B_S(\beta; \mathcal{S}) + B_P(\theta, \mathcal{P}) + B_E(\psi, \mathcal{E}), \quad (2)$$

where B_S, B_P and B_E are shape, pose, and expression blendshapes, respectively.

To compute the detailed shape, M and its surface normal N are first converted to UV space as M_{uv} and N_{uv}. The detail shape M' is then computed as:

$$M'_{uv} = M_{uv} + D \odot N_{uv}, \quad (3)$$

where D denotes the vector obtained from decoding the concatenation of δ, θ, and ψ by F_d. $\odot$ denotes the Hadamard product.

DECA uses a Spherical Harmonics (SH) based [23] illumination model. The shaded face image B is computed as:

$$B(\alpha, l, N_{uv}) = A(\alpha)_{i,j} \odot \sum_{k=1}^{9} l_k H_k(N_{i,j}), \quad (4)$$

where A represents albedo map computed from albedo parameters α, and H denotes SH basis and coefficients.

Finally, DECA renders 2D image I_r using the differentiable rasterizer from Pytorch3D [25] $\mathcal{R}$:

$$I_r = \mathcal{R}(\bar{M}, B, c), \quad (5)$$

where $\bar{M} = M$ for coarse reconstruction and $\bar{M} = M'$ for detailed reconstruction.

4. Approach

4.1. Data Collection

Figure 2: Three sequences from our dataset. Row 1 and row 2 are selected from Castle and row 3 is selected from Friends. Note that the face captures from Friends have lower resolutions and more motion blurs (see column 6 of row 3 as an example). In addition, since characters in Friends may not show up in continuous frames, sequences of Friends contain less temporal information.

We collect face sequences of TV series characters from the TVQA dataset [21], a large-scale video QA dataset based on 6 popular TV shows. Frames are all selected from 2 of the TV shows: Castle and Friends, where only one character shows up and has consecutive movements for over 10 frames in the same scene (not necessarily in continuous frames). Among these we build sequences of 10 frames for each individuals. In total, we collect 3000+ valid, clear facial images from sequences under a diverse set of environment conditions (lighting, occlusion, resolution, etc.). We use FaceNet [30] to perform face detection and recognition. Each detected face is cropped and resized into 224x224. It is worth noticing that the face captures from Friends originally have lower resolutions and more motion blurs, and they contain less temporal information. Figure 2 shows 3 sequences from our data.

Figure 3: DECA results on Castle and Friends sequences trained with our dataset. Red points in predicted keypoints are the ones predicted incorrectly.

4.2. Data Preprocessing

As required for DECA training, we perform keypoint detection and face segmentation on each face image in our own dataset. We use 2D-FAN in [3] to extract 68 2D facial keypoints from each face image. To get segmentation masks, we use a BiSeNet [35] based face parsing tool [1]. We combine predicted regions that represent skin, brows, eyes, nose, mouth, and lips as the predicted segmentation mask [1].

4.3. Reproducing DECA Performance

To evaluate the quality of the 3D faces we can get without utilizing temporal information, we first test the original DECA architecture on our dataset. We use the pre-trained DECA model and train on our dataset, inputting individual faces. We show results on two sequences in figure 3. Although DECA is able to produce reasonable 3D faces on our test inputs, we notice several limitations: some keypoints are not predicted correctly; faces with motion blurs and profile views are not well reconstructed; some facial expressions in the detail shapes do not match well with the inputs.

4.4. Method Improvement

In order to utilize the temporal information, we modify the DECA architecture to allow inputting a sequence of faces. We replace the encoders E_c and E_d with two temporal encoders E_{T_c} and E_{T_d}. In each temporal encoder, a bidirectional RNN based temporal feature extractor is used to extract temporal information across all frames in the sequence. Figure 1 illustrates the overall architecture of our modified network.

[1]Our adapted version to perform face segmentation: https://github.com/xiyichen/face-parsing.PyTorch

4.4.1 Temporal Encoder

layer name	output size	layer components
head	224x224	3x3, 64
down1	112x112	$\begin{pmatrix} \text{3x3, 64} \\ \text{3x3, 64} \end{pmatrix}$ x5 3x3, 64, stride 2
rnn_propagation	112x112	Flow-Guided Deformable Alignment [5] Feature Concatenation 3x3, 64 LeakyReLU $\begin{pmatrix} \text{3x3, 64} \\ \text{3x3, 64} \end{pmatrix}$ x5
rnn_fusion	112x112	Feature Concatenation 3x3, 64 LeakyReLU $\begin{pmatrix} \text{3x3, 64} \\ \text{3x3, 64} \end{pmatrix}$ x5
down2	56x56	$\begin{pmatrix} \text{3x3, 64} \\ \text{3x3, 64} \end{pmatrix}$ x5 3x3, 64, stride 2
layer1	56x56	$\begin{pmatrix} \text{1x1, 64} \\ \text{3x3, 64} \\ \text{1x1, 256} \end{pmatrix}$ x3
layer2	28x28	$\begin{pmatrix} \text{1x1, 128} \\ \text{3x3, 128} \\ \text{1x1, 512} \end{pmatrix}$ x4
layer3	14x14	$\begin{pmatrix} \text{1x1, 256} \\ \text{3x3, 256} \\ \text{1x1, 1024} \end{pmatrix}$ x6
layer4	7x7	$\begin{pmatrix} \text{1x1, 512} \\ \text{3x3, 512} \\ \text{1x1, 2048} \end{pmatrix}$ x3
avgpool	1x1	average pooling

Table 1: Overall architecture of our temporal encoder. Residual blocks are shown in brackets with the numbers of blocks stacked stated on their right. All residual blocks we use contain batch normalizations between two convolutional layers. Stride and padding for convolutional layers are both set to 1.

Table 1 shows the detail components of all layers in our encoder. The input images of a sequence of T frames first go through a convolutional layer "head" to expand the channel size to 64. They are then downsampled by a factor of 2 using layer "down1", consisting of 5 residual blocks and a convolutional layer with kernel size 3 and stride 2. After that, a bidirectional RNN is used to extract temporal features, which we will explain in 4.4.2. A second downsample layer "down2" is then used to further downsample the features by 2. The rest of the architecture remains the same as the encoder in DECA, containing 4 "bottleneck" blocks in ResNet50 [16] followed by average pooling and two fully-connected layers to regress low-dimensional latent codes.

4.4.2 Temporal Feature Extractor

We apply the idea of second-order grid propagation in a bidirectional RNN with flow-guided deformable alignment as in BasicVSR++ [5] and build a temporal feature extractor. Instead of performing super-resolution, we remove the upsampling part and only use the bidirectional RNN to extract temporal features from all frames in the sequence. The temporal feature extractor contains two parts: *propagation* and *fusion*.

The *propagation* part consists of $2\mathcal{L}$ RNN layers: $backward_1, forward_1, ..., backward_{\mathcal{L}}, forward_{\mathcal{L}}$, propagating intermediate features backward and forward alternatively, as shown in the right part of figure 1. Second-order grid propagation in BasicVSR++ is applied to integrate intermediate features from the neighboring two frames (the previous two for a $forward$ layer or the next two for a $backward$ layer). The propagated feature $\hat{f}_i^j$ for the i-th frame in the j-th RNN layer is computed as:

$$\hat{f}_i^j = \mathcal{A}\left(g_i, f_{i-1}^j, f_{i-2}^j, s_{i\to i-1}, s_{i\to i-2}\right)^{2}, \quad (6)$$

where $\mathcal{A}$ represents the flow-guided deformable alignment block as in BasicVSR++, g_i denotes the input features (outputs from "down1" layer), f_{i-1}^j and f_{i-2}^j denote intermediate features from (i-1)-th and (i-2)-th frames, and $s_{i\to i-1}$ and $s_{i\to i-2}$ represent optical flows from the i-th frame to the (i-1)-th and (i-2)-th frames. As in BasicVSR++, we use a pre-trained SPyNet [24] as our flow network.

The final intermediate feature f_i^j is then computed as:

$$f_i^j = \hat{f}_i^j + \mathcal{R}\left(c\left(f_i^{j-1}, \hat{f}_i^j\right)\right), \quad (7)$$

where $\mathcal{R}$ represents a stack of residual blocks (a convolutional layer with kernel size 3, followed by a Leaky ReLU [33], followed by 5 residual blocks) and c represents channel-wise concatenation.

The *fusion* part consists of the same stack of residual blocks $\mathcal{R}$. The output temporal feature of the i-th of frame, τ_i, which we will then forward to "down2" layer, is computed as:

$$\tau_i = \mathcal{R}\left(c\left(g_i, f_i^{backward_1}, ..., f_i^{forward_{\mathcal{L}}}\right)\right), \quad (8)$$

where we concatenate the intermediate features of i-th frame in all $2\mathcal{L}$ RNN layers and the input g_i and pass it through the residual blocks.

4.5. Loss Functions

We use the same loss functions as in DECA to compare the rendered 2D (coarse/detailed) faces with the input face, but on a frame-by-frame basis. For coarse reconstruction on the i-th frame, we compute the reconstruction loss L_{coarse_i} as:

$$L_{coarse_i} = L_{lmk_i} + L_{eye_i} + L_{pho_i} + L_{id_i} + L_{sc_i} + L_{reg_i}, \quad (9)$$

where L_{lmk}, L_{eye}, L_{pho} L_{id}, L_{sc}, L_{reg} represent landmark loss, eye closure loss, photometric loss, identity loss, shape consistency loss and regularization, respectively.

For detail reconstruction on the i-th frame, we compute the reconstruction loss L_{detail_i} as:

$$L_{detail_i} = L_{phoD_i} + L_{mrf_i} + L_{sym_i} + L_{dc_i} + L_{regD_i}, \quad (10)$$

where L_{phoD}, L_{mrf}, L_{sym}, L_{dc}, L_{regD} represent photometric detail loss, ID-MRF loss, soft symmetry loss, detail consistency loss, and detail regularization, respectively.

We direct readers to the DECA paper for more details on each components of the loss functions.

4.6. Implementation & Training Details

Due to the small size of our dataset, we first train a pretrain model for 10000 epochs, and then train the coarse model for 500 epochs, and finally train the detail model for 10000 epochs. E_{T_c} is fixed while training the detail model. Batch size is set to 2. The number of frames in a sequence, T, is set to 5. The number of RNN layers in each direction, $\mathcal{L}$, is set to 2. All loss weights remain unchanged from DECA. Adam optimizer [20] is used with a learning rate of $1e-4$. Our code is modified based on DECA and BasicVSR++ and is publicly available on GitHub. [3]

5. Qualitative Evaluation

We compare our modified model with temporal encoders with the original DECA, both trained on our own dataset. We show visualization results in figure 4. Our method shows improvements in facial expression reconstruction over DECA, especially in details of mouths. Even in dark

[2] $f_i^0 = g_i; s_{0\to-1} = s_{0\to-2} = s_{1\to-1} = f_{-1} = f_{-2} = 0$

[3] https://github.com/xiaoxiaokongxi/Learning-to-Reconstruct-3D-Faces-by-Watching-TV

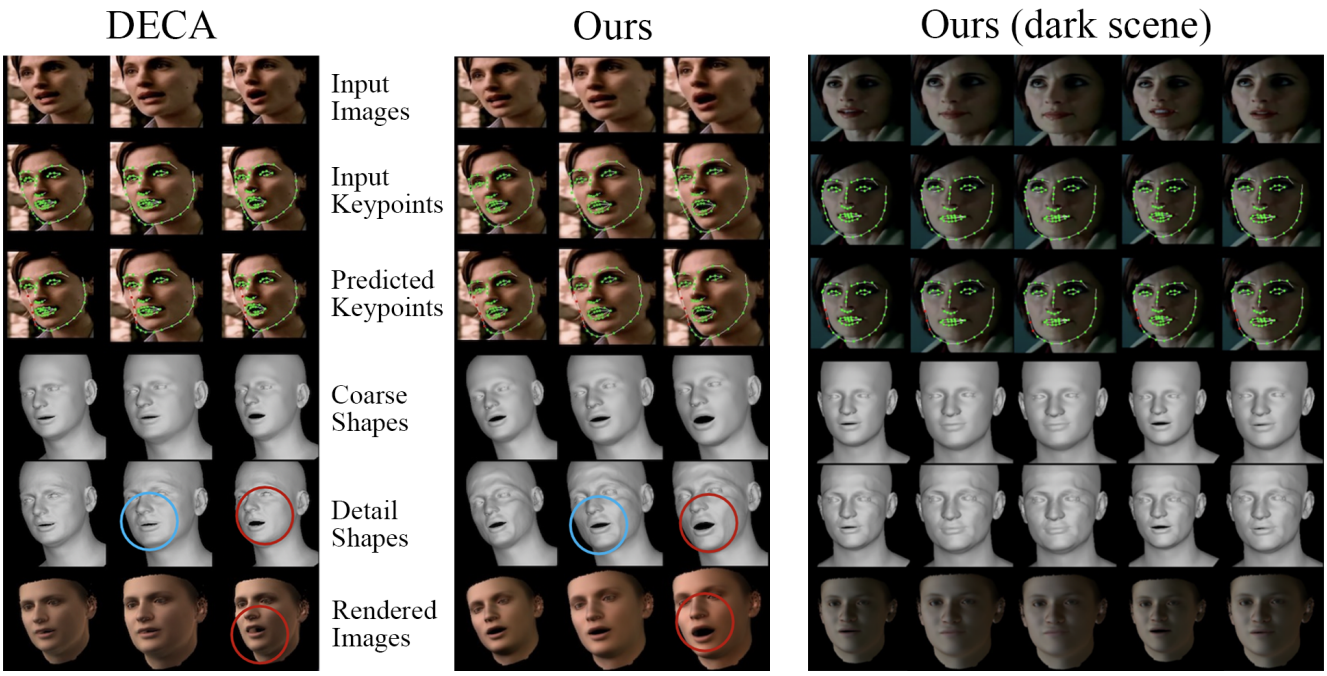

Figure 4: Qualitative evaluation results. (left) DECA v.s. our improve method with temporal encoders on a sequence from Castle in our dataset. The circles show improvements in facial expressions (especially details of mouths). Although there are still incorrect predictions for facial keypoints (see red points in row 3), they align better with the ground truths. (right) Our method on a sequence in dark scene in Castle. The facial details could still be accurate in these settings.

environments, our method could still predict accurate detail shapes.

We do not present quantitative evaluation results due to the lack of ground truth 3D face meshes of our own dataset. We direct readers who are interested in quantitative evaluation on our work to 3D face benchmark datasets such as [29] and [11] to evaluate accuracy of the coarse shapes. Although there is no temporal sequences in these benchmark datasets, we believe that solid performance on these benchmarks would ensure the predictive power of our model for both individual and sequential inputs.

6. Conclusion & Future Works

In this work, we show that using temporal encoders to extract temporal information from sequential faces is beneficial for facial details reconstruction, even in very dark light conditions. However, our method still have some limitations: 1) Our dataset contains faces with motion blurs (mostly in sequences from Friends) which could not be reconstructed accurately. Face deblurring methods for single frames [31] or videos [26] could be applied to these blurring frames. 2) Although the detail consistency loss in DECA disentangles identity-dependent and expression-dependent facial details, we notice in detail shapes that some expressions may still be influenced by features from other parts due to feature fusion. How to disentangle the detail code effectively can be a good topic for future works.

Future works could also focus on 1) reconstructing 3D meshes from a multi-view perspective to ensure shape consistency across frames in the sequence, 2) integreting rendered images and input sequences into a training loop to improve 3D reconstruction.

7. Work Distribution

Mengya Liu contributes on data collection and reproducing DECA performance. Jiezhang Cao contributes on implementing the temporal encoder, training, and evaluation. Tianhao Li contributes on data collection, data preprocessing, and reproducing DECA performance. Xiyi Chen contributes on data preprocessing and writing the report.

8. Acknowledgements

We thank Yao Feng for her supervision on this 3D Vision (252-0579-00L) course project at ETH Zürich. All of our training and evaluations are performed on ETH Euler cluster.

References

[1] Using modified bisenet for face parsing in pytorch. https://github.com/zllrunning/face-parsing.PyTorch. 4

[2] Oswald Aldrian and William A.P. Smith. Inverse rendering of faces with a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1080–1093, 2013. 2

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 4

[4] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021. 2, 3

[5] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 4, 5

[6] Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. Photo-realistic facial details synthesis from single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9429–9439, 2019. 1, 2

[7] Pengfei Dou, Shishir K. Shah, and Ioannis A. Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1503–1512, 2017. 2

[8] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models—past, present, and future. *ACM Trans. Graph.*, 39(5), jun 2020. 2

[9] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021. 1, 2

[10] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 1, 2

[11] Zhen-Hua Feng, Patrik Huber, Josef Kittler, Peter J. B. Hancock, Xiaojun Wu, Qijun Zhao, Paul Koppen, and Matthias Rätsch. Evaluation of dense 3d reconstruction from 2d face images in the wild. *CoRR*, abs/1803.05536, 2018. 6

[12] Victoria Fernández Abrevaya, Adnane Boukhayma, Philip H.S. Torr, and Edmond Boyer. Cross-modal deep face normals with deactivable skip connections. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4978–4988, 2020. 1, 2

[13] Z. Geng, C. Cao, and S. Tulyakov. 3d guided fine-grained face manipulation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9813–9822, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. 1

[14] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J. Black, and Timo Bolkart. GIF: Generative interpretable faces. In *International Conference on 3D Vision (3DV)*, 2020. 1

[15] Yudong Guo, juyong zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(6):1294–1307, jun 2019. 1, 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 3, 5

[17] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, page 645–660, Berlin, Heidelberg, 2020. Springer-Verlag. 3

[18] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Perez, Christian Richardt, Michael Zollhöfer, and Christian Theobald. Deep video portraits. *ACM Trans. Graph.*, 37(4), jul 2018. 1

[19] Hyeongwoo Kim, Michael Zollöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep single-shot inverse face rendering from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[21] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018. 3

[22] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6), nov 2017. 2, 3

[23] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 497–500, New York, NY, USA, 2001. Association for Computing Machinery. 3

[24] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[25] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 3

[26] Wenqi Ren, Jiaolong Yang, Senyou Deng, David Wipf, Xiaochun Cao, and Xin Tong. Face video deblurring via 3d facial priors. In *IEEE International Conference on Computer Vision*, 2019. 6

[27] Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando de la Torre, and Yaser Sheikh. Meshtalk: 3d face animation from speech using cross-modality disentangle-

ment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1173–1182, October 2021. 1

[28] Jérémy Riviere, Paulo Gotardo, Derek Bradley, Abhijeet Ghosh, and Thabo Beeler. Single-shot high-quality facial geometry and skin appearance capture. *ACM Trans. Graph.*, 39(4), jul 2020. 2

[29] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 7763–7772, June 2019. 6

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. 3

[31] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8260–8269, 2018. 6

[32] Thomas Vetter and Volker Blanz. Estimating coloured 3d face models from single images: An example based approach. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision — ECCV'98*, pages 499–513, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. 2

[33] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015. 5

[34] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: A large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[35] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, pages 334–349. Springer, 2018. 4

[36] Xiaoxing Zeng, Xiaojiang Peng, and Yu Qiao. Df2net: A dense-fine-finer network for detailed 3d face reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2315–2324, 2019. 1, 2

[37] Jian Zhao, Lin Xiong, Yu Cheng, Yi Cheng, Jianshu Li, Li Zhou, Yan Xu, Jayashree Karlekar, Sugiri Pranata, Shengmei Shen, Junliang Xing, Shuicheng Yan, and Jiashi Feng. 3d-aided deep pose-invariant face recognition. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 1184–1190. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 1

[38] Michael Zollhöfer, Justus Thies, Darek Bradley, Pablo Garrido, Thabo Beeler, Patrick Péerez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. 2018. 2