Supplementary Material for Topo4D: Topology-Preserving Gaussian Splatting for High-Fidelity 4D Head Capture

Xuanchen Li[1], Yuhao Cheng[1], Xingyu Ren[1], Haozhe Jia[2], Di Xu[2], Wenhan Zhu[3], and Yichao Yan[1†]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{lixc6486,chengyuhao,rxy_sjtu,yanyichao}@sjtu.edu.cn
[2] Huawei Cloud Computing Technologies Co., Ltd
{jiahaozhe1,xudi21}@huawei.com
[3] Xueshen AI whzhu@foxmail.com

1 Capture System Setup

As shown in Fig. 1 (a), our *Light Stage* system features 4D data capture capabilities, consisting of 24 time-consistent dynamic cameras capable of capturing multi-view videos at 60FPS with a resolution of 4096×3000 pixels. Each camera is hardware-controlled with a time error of less than 1 microsecond. The 24 cameras are precisely calibrated to obtain accurate intrinsic and extrinsic parameters. To minimize perspective errors caused by facial shadows, we employ multiple surrounding light sources. Subjects are constrained to chairs during performances, maintaining relatively static body movements, which aligns with the facial capture requirements in industrial production processes.

During filming, we capture videos for 10 identities with 2 segments of videos each, lasting approximately 20 seconds for a segment. In one segment, the subjects spontaneously recite a lengthy passage to simulate natural speech conditions. In the other segment, the subjects randomly change expressions, with many expressions being extreme, involving severe facial distortions and wrinkles, to validate the algorithm's generality and adaptability to extreme scenarios. We show some examples in Fig. 1 (b).

2 Implementation Details

2.1 Gaussian Normal Expansion

The Gaussian function in 3D space corresponds to an ellipsoid. Consequently, the surface shaped by the Gaussian function differs from the geometric surface formed by the Gaussian mean position, which the Gaussian functions generally encase. We thus shift each Gaussian in the direction of the vertex normal to compensate for the gap caused by Gaussian's scale and finally obtain the final

(a) Our Light Stage System (b) Examples of captured multi-view videos

Fig. 1: We show (a) the diagram of our Light Stage capture system, and (b) some Raw data for different identities.

meshes. Specifically, we offset each Gaussian's mean position $\boldsymbol{\mu}$ by its projection distance from the surface of the ellipsoid in the vertex normal direction:

$$\boldsymbol{n}' = R^{-1}\boldsymbol{n} \tag{1}$$

$$\boldsymbol{\mu}' = \boldsymbol{\mu} + \sqrt{\frac{1}{\frac{n_x'^2}{s_x^2} + \frac{n_y'^2}{s_y^2} + \frac{n_z'^2}{s_z^2}}}\boldsymbol{n}, \tag{2}$$

where $\boldsymbol{n}$ is the corresponding vertex normal of each Gaussian.

2.2 UV Space Densification

When performing UV Space Densification, we insert each quadrilateral grid in G_t' with $(N \times N)$ smaller equidistant grids. Specifically, the newly generated Gaussian's position, UV coordinates, and color are obtained by bilinear interpolation sampling the corresponding attributes of the four Gaussians (e.g. $G'_{0,0}$, $G'_{0,N-1}$, $G'_{N-1,0}$ and $G'_{N,N}$) at the original grid vertices:

$$\begin{aligned} A_{i,j} = \frac{1}{(N-1)\times(N-1)} \begin{bmatrix} N-1-i & i \end{bmatrix} \\ \begin{bmatrix} A_{0,0} & A_{0,N-1} \\ A_{N-1,0} & A_{N-1,N-1} \end{bmatrix} \begin{bmatrix} N-1-j \\ j \end{bmatrix}, \end{aligned} \tag{3}$$

where A can be color, uv coordinate, and position. Most importantly, we can easily establish topological relationships between Gaussians. For example, $G_{i,j}$ is connected to $G_{i-1,j}$, $G_{i+1,j}$, $G_{i,j-1}$ and $G_{i,j+1}$. This process is equivalent to subdividing the grid and inserting more sampling points in UV space.

In implementation, we only perform densification once in the first frame and compute every Gaussian's interpolation weights given by Eq. 3. In the texture optimization stage of each subsquent frame, we use these weights to calculate the attributes of each Gaussian in G_t' for dense Gaussian Mesh initialization.

Table 1: The initialization settings and learning strategies for each Gaussian attribute.

Symbol	Initialization	First Frame Optimization	Geometry Optimization	Texture Optimization	Learning Rate
$\boldsymbol{\mu}$	Vertex Coordinate	Fixed	Learnable	Fixed	0.000016
$\boldsymbol{q}$	Vertex Normal	Learnable	Learnable	Fixed	0.001
s	Half of Min Neighbour Distance	Learnable	Fixed	Fixed	0.001
σ	1	Fixed	Fixed	Fixed	0.0
$\boldsymbol{c}$	Texture Color	Fixed	Fixed	Learnable	0.0025

2.3 Texture Inverse Mapping

We perform a rasterization-based inverse mapping operation similar to the forward pass in NVDiffrast [9] to render texture maps. Specifically, if the Gaussian indices of the visible triangles for a pixel at (x, y) are denoted as $i_{0,1,2}$ and the center of mass is denoted as $w_{0,1,2}$, we can calculate the color $\boldsymbol{C}_{x,y}$ of the pixel at (x, y). That is:

$$\boldsymbol{C}_{x,y} = w_0\boldsymbol{c}_{i_0} + w_1\boldsymbol{c}_{i_1} + (1 - w_0 - w_1)\boldsymbol{c}_{i_2}. \tag{4}$$

2.4 Learning Strategy

As is shown in Tab. 1, we adopt different optimizing strategies at different stages in our pipeline. **1)** When optimizing the initial Gaussian Mesh at the first frame, we only optimize $\boldsymbol{q}$ and $\boldsymbol{s}$, and keep others fixed. **2)** During geometry optimization, we optimize these geometry-related attributes ($\boldsymbol{\mu}$ and $\boldsymbol{q}$) to track Gaussian motions. **3)** During texture optimization, we initialize dense Gaussian Mesh by sampling each Gaussian's position by Eq. 3 and set their opacity to 1. Since we only need to learn the exact color of the UV space sampling points represented by each Gaussian, we fix the Gaussian scale to be the minimum distance with their one-ring neighbors and only optimize Gaussian's color.

3 Additional Experiments

In this section, we conduct more comparisons with current SOTAs toward geometries and textures.

3.1 Additional Geometry Comparisons

Fig. 2 shows additional qualitative comparisons of our method with current SOTAs, *i.e.*, DECA [7], HRN [5], MVFR [16], DFNRMVS [3] and traditional multi-view stereo [12] with iterative closest point [4] (ICP) pipeline. The results indicate that our method outperforms superiorly other approaches. Besides, our method can achieve competitive results with manual registrations while avoiding interpenetration that occurs in the automated ICP method.

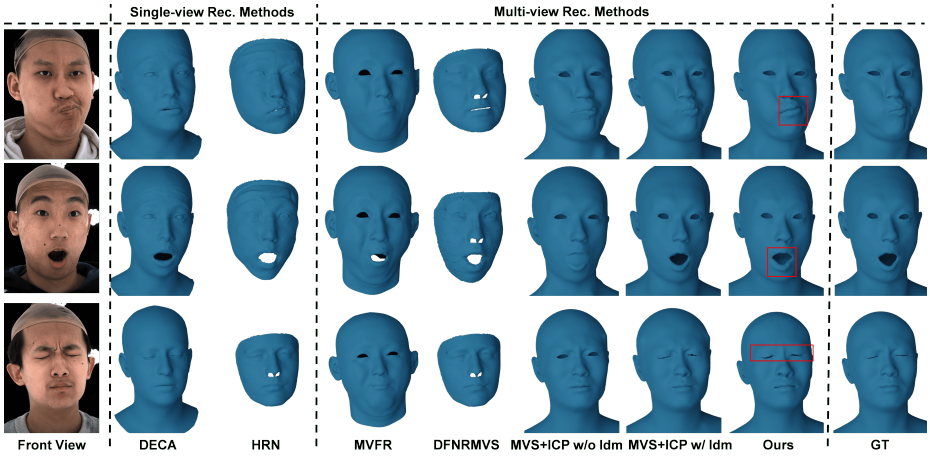

Fig. 2: Qualitative evaluation of meshes generated by our method and other topology-consistent reconstruction methods. We use artist-manually registered head mesh as the ground truth. We highlight areas that are difficult to reconstruct.

3.2 Additional Texture Comparison

Fig. 3 shows additional comparisons of rendering results of textures generated by our method, UnsupTex [13], and HRN [5]. These pre-trained model-based methods are unable to handle high-resolution data, and can only generate textures of lower resolution, losing details in rendering results. Benefiting from **UV Space Densification**, each Gaussian that represents sampling points in UV space can accurately learn pixel-level realistic details from high-resolution inputs. It's worth mentioning since it is difficult to obtain the same lighting conditions as the capture conditions in the rendering software, there is a certain difference between the rendering results and the captured images. Nonetheless, we also achieve identity consistency and the same details in textures as the images. Moreover, it can be observed that the generated texture maps maintain realistic wrinkles and pore-level details.

3.3 Qualitative Comparisons with Current SOTAs

We next supply comparisons of our method with TEMPEH [6] and ReFA [11], two state-of-the-art multi-view reconstruction methods. Due to different method settings, the mismatch in data structures, or the inaccessibility of codes, it is particularly difficult to compare our method with them. Therefore, we compare them qualitatively in some reasonable settings to show the competitiveness and extensibility of our method, which will be described detailed in the comparisons.

Comparisons with TEMPEH. The recent TEMPEH [6], trained with a large amount of 4D head data, can directly regress dynamic topologic head models

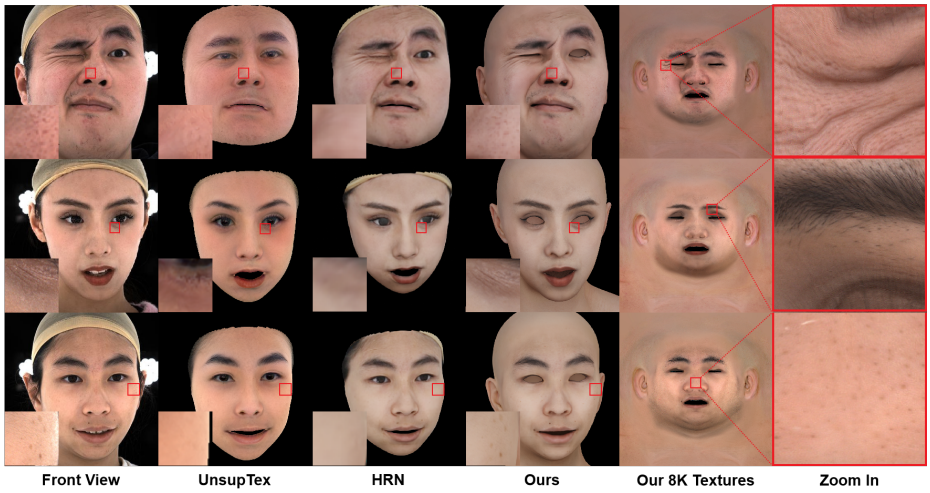

Fig. 3: Additional qualitative evaluation of the rendering results between our method, UnsupTex [13] and HRN [5]. We also provide the zoom-in renderings for better observation. Our generated 8K textures and pore-level zoom-in details are demonstrated in columns 5 and 6.

from multi-view videos, which shares a similar setting as our work. However, 2 major differences exist between TEMPEH and our Topo4D: **1)** TEMPEH is trained on its proposed dataset FaMoS, thus it is only applicable to the specific capture system. Hence, new data are required to train the model to utilize TEMPEH in new capture systems. Conversely, our Topo4D is suitable for more systems with calibrated cameras including FaMoS. **2)** TEMPEH can only generate meshes while our Topo4D can generate both meshes and high-quality textures.

For fair comparisons, we compare TEMPEH and our Topo4D both on our dataset and TEMPEH's FaMoS dataset. Note that, due to the lack of a large amount of registered data in our dataset as supervision, we are unable to train TEMPEH by ourselves. Therefore, we use its publicly released pre-trained models for face reconstruction. As is shown in Fig. 4, **1)** on our dataset, our method faithfully reconstructs facial geometry and generates 8K textures that can produce realistic rendering results. However, TEMPEH fails to reconstruct meaningful heads, revealing that the pre-trained TEMPEH cannot be directly applied to different capture systems. **2)** FaMoS consists of 16 gray-scale and 8 color images in each frame, including 2 posterolateral color images. In the experiments, we only use 6 color images containing the front face as the input of our method, while showing TEMPEH's best results with its publicly released pre-trained model on 16 gray-scale views. Even though, we achieve competitive geometric results with TEMPEH. Especially under some extreme conditions in the 4th and 6th columns in Fig. 4, we can even better restore asymmetrical eyebrow and pouting expressions. Additionally, our method can generate high-fidelity textures while TEMPEH cannot. **Overall**, our method is more general

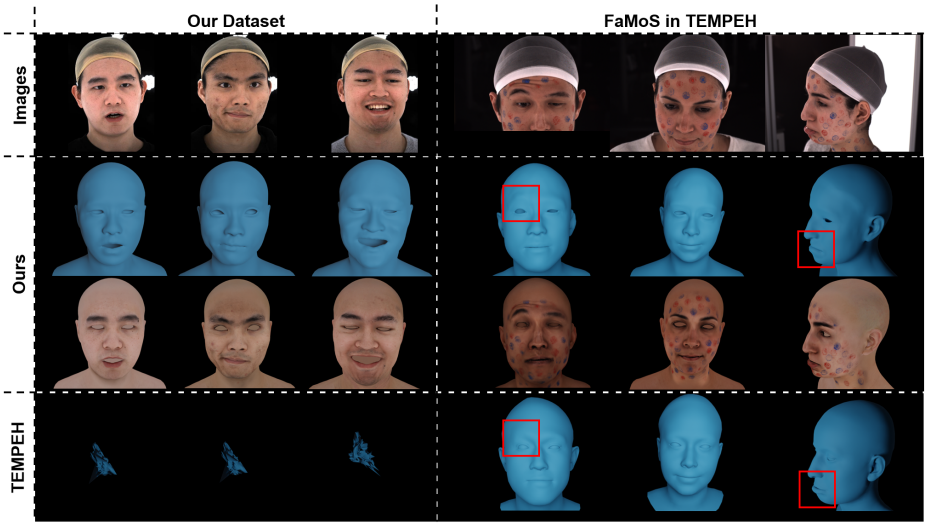

Fig. 4: Qualitative comparisons with TEMPEH [6] on our dataset and FaMoS. We realize high-quality face reconstruction with textures in both our dataset and FaMoS. TEMPEH achieves competitive geometry results as ours, but it fails to reconstruct facial models in our dataset. Moreover, TEMPEH cannot generate texture maps.

across capture systems than TEMPEH [6], and we can generate high-quality texture maps directly. Moreover, our method does not require a large amount of registered data for training.

Comparisons with ReFA. Similar to our Topo4D, ReFA [11] can generate face meshes from multi-view images, while it can generate 4K textures via super-resolution modules. However, since ReFA does not release its codes, pre-trained models, and test datasets, we borrow some results from its paper and website for qualitative comparisons with our Topo4D toward the quality of meshes and texture maps. In the geometric comparisons, we select some expressions similar to ReFA's results for better evaluation of extreme expressions. As shown in Fig. 5, our method can more faithfully reconstruct the detailed geometric meshes than ReFA, especially in the eyes regions, *e.g.*, 4th and 6th columns. As the comparisons on textures shown in Fig. 6, our Topo4D can faithfully generate 8K high-resolution dynamic textures, including pore-level details and individually discernible strands of hair. Instead, ReFA relies on a super-resolution module to obtain 4K textures, but it fails to faithfully restore the original details of the face and leads to artifacts in the texture. Overall, our Topo4D achieves better results than ReFA in both mesh fidelity and texture quality.

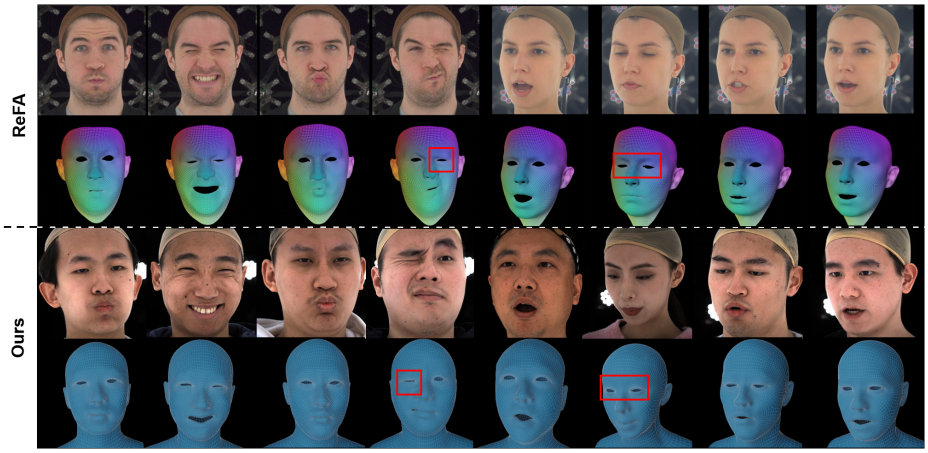

Fig. 5: Qualitative comparison of geometries with ReFA [11]. We show examples on our dataset similar to those expressions in ReFA's paper. Please **zoom-in** for detailed observation.

3.4 Performance on Multiface Dataset

To show our method's robustness against different capture systems and extreme expressions, we test our method on Multiface Dataset [15] and compare it with other methods, as is shown in Fig. 7 and Fig. 8.

We additionally compare our method with two landmark-based optimization methods: **(1)** Track + Wrap4D [2] guided by a commercial landmark detector, which is widely used in modern CG pipelines. **(2)** Smith et al. [14], the tracking pipeline used in Multiface. As is shown in Fig. 9, current landmark detection methods generally have notable errors under extreme expressions, resulting in wrong correspondence during tracking. Despite using a personalized detector, Smith et al. struggles to track dense areas like eyelids and lips. Due to significant bias in eyelid and lip keypoints, Warp4D produces serrated eyelids, interpenetrated mouth corners, and lips that registered to teeth. In contrast, our method is robust against extreme expressions and outperforms others while keeping stable dense correspondence without using landmarks.

3.5 Efficiency Comparisons with Optimization-based Methods

As is shown in Tab. 2, We compare our method with other optimization-based methods from a perspective of computational cost. It is noteworthy that most optimization-based algorithms are not open-sourced, thus we directly use the time cost claimed in their papers. The traditional pipeline (Metashape [1] + Wrap4D [2]) takes more than 5 minutes to reconstruct a mesh with texture, and requires significant more time for manual tweaking. The optical-flow based method Fyffe et al. [8] reconstructs meshes without MVS but is still more than

Fig. 6: Qualitative comparison of textures with ReFA [11]. We show examples of textures generated on our dataset. Please **zoom-in** for detailed observation.

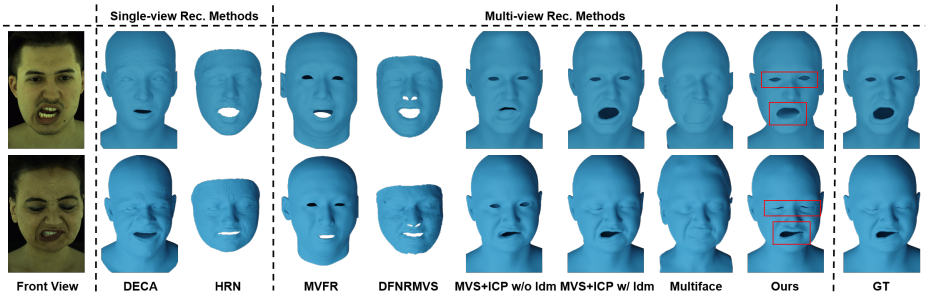

Fig. 7: Qualitative evaluation of meshes generated by our method and other topology-consistent reconstruction methods on Multiface [15] dataset. We use artist-manually registered head mesh as the ground truth. We frame areas that are difficult to reconstruct.

Table 2: Time required per frame of different optimization-based reconstruction methods.

Method	Mesh	Texture	MVS	Manual
MVS + ICP (Metashape [1]+Wrap4D [2])	≈4min	≈80s	✓	✓
Fyffe et al. [8]	≈25min	-	×	×
Ours	≈30s	≈30s	×	×

20 minutes slower than us when reconstructing a coarse mesh due to its time-consuming volumetric Laplacian solve. Our geometry tracking and 8K texture learning stage both takes only 30 seconds, which is fully automatic and not require any additional supervision such as scans, landmarks, optical flow, etc.

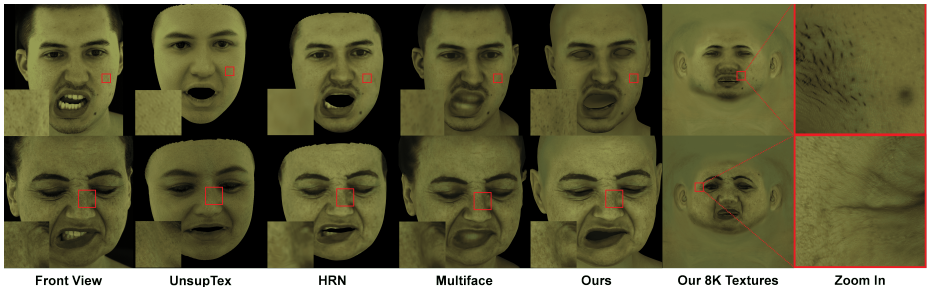

Fig. 8: Qualitative evaluation of textures generated by our method and other topology-consistent reconstruction methods on Multiface [15] dataset. Noting that the images in Multiface dataset are 2K resolution, our method still generates 8K textures.

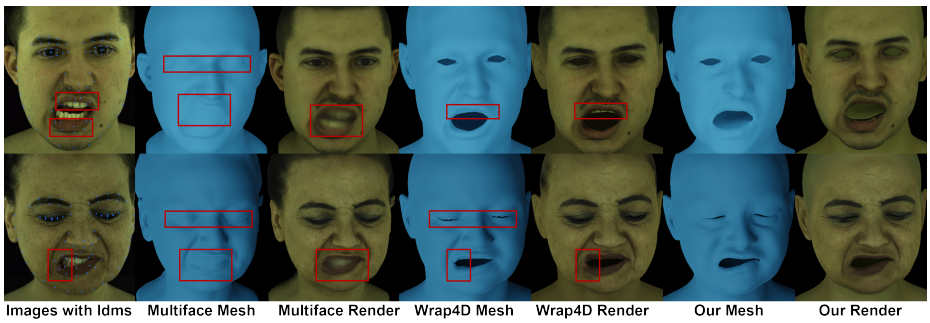

Fig. 9: Comparisons on Multiface Dataset. Challenging areas are highlighted in red boxes. Please **zoom-in**.

3.6 Gaussian Mesh Rendering Results

Fig. 10 (a) shows an example of the Gaussian rendering results of *Gaussian Mesh* and *Dense Gaussian Mesh* under the extreme expression. Our Gaussian Mesh contains only 8280 points for geometry optimization and is learned on 512×375 images. Although using such a small number of Gaussian points may cause some degree of blurring in the rendering results, it is sufficient to represent facial geometry and is photometric enough for tracking. Our Dense Gaussian Mesh contains around 5 million points and is learned on raw 4000×3000 images. A Gaussian point corresponds to about 5 pixels in the facial UV region, which is nearly enough for 8K texture. It can be observed from the rendering result that such a level of Gaussian density is sufficient to capture pore-level details.

4 Limitation Discussion

Topo4D is designed to achieve 4D facial registration without MVS reconstruction and artists' manual intervention. Although it can efficiently and automatically

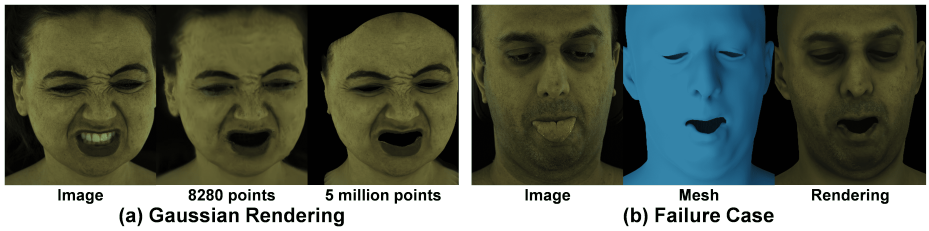

Fig. 10: (a) Rendering examples of Gaussian Mesh and Dense Gaussian Mesh. (b) An example of tracking failure caused by severe occlusion.

reconstruct 4D facial meshes with pore-level texture details, it still has some limitations. First, our method fails when heavy overlapping occurs due to the lose of tracking, such as sticking out the tongue as is shown in Fig. 10 (b), which is commonly solved by artists' manual operations [10]. Second, our method inevitably trade-offs between topology quality and surface accuracy, leading to smooth results. We will model detailed geometry by reconstructing displacement maps in future work. Last, due to limited camera angles and the absence of information on polarized light, our method primarily focuses on facial reconstruction and is limited to texture capture only. We aim to extend our method to PBR assets reconstruction in the future.

5 Ethics Discussion

In this work, all subjects have signed agreements authorizing us to use the collected data for scientific research on 4D facial reconstruction. We will make every effort to safeguard the original data from disclosure. Our method relies heavily on visual capture systems similar to the Light Stage for data collection, which can mitigate the risk of misuse to some extent. We are committed to privacy protection, preventing the misuse of 4D face reconstruction for criminal purposes.

References

1. Agisoft metashape: Agisoft metashape. https://www.agisoft.com/ 7, 8
2. Wrap4d - faceform. https://faceform.com/wrap4d/ 7, 8
3. Bai, Z., Cui, Z., Rahim, J.A., Liu, X., Tan, P.: Deep facial non-rigid multi-view stereo. In: CVPR. pp. 5850–5860 (2020) 3
4. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. pp. 586–606 (1992) 3
5. Biwen, L., Jianqiang, R., Mengyang, F., Miaomiao, C., Xuansong, X.: A hierarchical representation network for accurate and detailed face reconstruction from in-the-wild images. In: CVPR. pp. 394–403 (2023) 3, 4, 5
6. Bolkart, T., Li, T., Black, M.J.: Instant multi-view head capture through learnable registration. In: CVPR. pp. 768–779 (2023) 4, 6

7. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an animatable detailed 3D face model from in-the-wild images. In: TOG (2021) 3
8. Fyffe, G., Nagano, K., Huynh, L., Saito, S., Busch, J., Jones, A., Li, H., Debevec, P.: Multi-view stereo on consistent face topology. In: Computer Graphics Forum. pp. 295–309 (2017) 7, 8
9. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. TOG (2020) 3
10. Li, R., Bladin, K., Zhao, Y., Chinara, C., Ingraham, O., Xiang, P., Ren, X., Prasad, P., Kishore, B., Xing, J., et al.: Learning formation of physically-based face attributes. In: CVPR. pp. 3410–3419 (2020) 10
11. Liu, S., Cai, Y., Chen, H., Zhou, Y., Zhao, Y.: Rapid face asset acquisition with recurrent feature alignment. TOG pp. 1–17 (2022) 4, 6, 7, 8
12. Loop, C., Zhang, Z.: Computing rectifying homographies for stereo vision. In: CVPR. pp. 125–131 (1999) 3
13. Slossberg, R., Jubran, I., Kimmel, R.: Unsupervised high-fidelity facial texture generation and reconstruction. In: ECCV (2022) 4, 5
14. Smith, B., Wu, C., Wen, H., Peluse, P., Sheikh, Y., Hodgins, J.K., Shiratori, T.: Constraining dense hand surface tracking with elasticity. ACM Transactions on Graphics (ToG) **39**(6), 1–14 (2020) 7
15. Wuu, C.h., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Huang, X., et al.: Multiface: A dataset for neural face rendering. arXiv preprint arXiv:2207.11243 (2022) 7, 8, 9
16. Xiao, Y., Zhu, H., Yang, H., Diao, Z., Lu, X., Cao, X.: Detailed facial geometry recovery from multi-view images by learning an implicit function. In: AAAI (2022) 3