Mesh-Tension Driven Expression-Based Wrinkles for Synthetic Faces

Raman, Chirag; Hewitt, Charlie; Wood, Erroll ; Baltrusaitis, Tadas

DOI
10.1109/WACV56688.2023.00351

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

Citation (APA)
Raman, C., Hewitt, C., Wood, E., & Baltrusaitis, T. (2023). Mesh-Tension Driven Expression-Based Wrinkles for Synthetic Faces. In L. O'Conner (Ed.), *Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 3504-3514). IEEE.
https://doi.org/10.1109/WACV56688.2023.00351

2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-9346-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WACV56688.2023.00351

Mesh-Tension Driven Expression-Based Wrinkles for Synthetic Faces

Chirag Raman
Delft University of Technology
chiragraman.com

Charlie Hewitt
Microsoft
chewitt.me

Erroll Wood
Microsoft
errollw.com

Tadas Baltrušaitis
Microsoft
tabaltru@microsoft.com

Abstract

Recent advances in synthesizing realistic faces have shown that synthetic training data can replace real data for various face-related computer vision tasks. A question arises: how important is realism? Is the pursuit of photorealism excessive? In this work, we show otherwise. We boost the realism of our synthetic faces by introducing dynamic skin wrinkles in response to facial expressions, and observe significant performance improvements in downstream computer vision tasks. Previous approaches for producing such wrinkles either required prohibitive artist effort to scale across identities and expressions, or were not capable of reconstructing high-frequency skin details with sufficient fidelity. Our key contribution is an approach that produces realistic wrinkles across a large and diverse population of digital humans. Concretely, we formalize the concept of mesh-tension and use it to aggregate possible wrinkles from high-quality expression scans into albedo and displacement texture maps. At synthesis, we use these maps to produce wrinkles even for expressions not represented in the source scans. Additionally, to provide a more nuanced indicator of model performance under deformations resulting from compressed expressions, we introduce the 300W-winks evaluation subset and the Pexels dataset of closed eyes and winks.

1. Introduction

Synthetic data has been commonly employed for a variety of computer vision tasks including object recognition [2–5], scene understanding [6–9], eye tracking [10, 11], hand tracking [12, 13], and full body analysis [14–16]. However, the complexity of modeling the human head has largely precluded the generation of full-face synthetics for face-related machine learning. While realistic digital humans have been created for movies and video games, they usually entail significant artist effort per character [17, 18]. Consequently in literature, the synthesis of facial training data has been accompanied by simplifications, or a focus on parts of the face such as the eye region [19, 20] or the *hockey mask* [21–24]. This has resulted in a *domain gap*—a difference in distributions between real and synthetic facial data that makes generalization challenging. Efforts towards bridging this domain gap have mainly utilized domain adaptation to refine synthesized images [25] or domain-adversarial training where models are encouraged to ignore domain differences [26]. As such, generating realistic face data has been considered so challenging that it is assumed that synthetic data cannot fully replace real data for in-the-wild tasks [25].

Figure 1: **Final renders for a diverse set of synthetic identities and expressions.** For each identity we illustrate renders using the base method of Wood et al. [1] (left), and our added technique for generating expression-based wrinkling effects (right). For the same expression parameters, our method produces varied wrinkling effects across distinct identities (middle and bottom row).

2642-9381/23/$31.00 ©2023 IEEE
DOI 10.1109/WACV56688.2023.00351

To directly address the challenge, Wood et al. [1] attempted to minimize the domain gap at the source, by generating synthetic faces with unprecedented realism. Their method procedurally combines a parametric 3D face model with a comprehensive library of high-quality artist-created assets including textures, hair, and clothing. In doing so, the method overcomes a key bottleneck in techniques employed by the Visual Effects (VFX) industry for synthesizing realistic humans—that of scale. The procedural sampling can randomly create and render novel 3D faces without manual intervention. Machine learning systems trained on the synthesized data for landmark localization and face parsing achieved performance comparable with the state-of-the-art without using a single real image.

However, one limitation of the method proposed by Wood et al. [1] is the lack of dynamic, expression dependent wrinkles. The method generates textures using only the neutral-expression scans, which remain static for all deformations of the underlying face mesh resulting from expression changes. In this work we propose a simple yet effective method for incorporating expression-based wrinkles. Our central idea is to capture complex wrinkling effects for an identity from high-resolution scans of their posed expressions. We store all these possible wrinkles into albedo and displacement textures we refer to as *wrinkle maps*. At synthesis, for any arbitrary expression beyond those represented in the source scans, we blend between the neutral and wrinkle textures using a notion of the *tension* in the face mesh to obtain dynamic wrinkling effects. Figure 1 contrasts the results of our method against the current state-of-the-art (SOTA) approach for face synthetics. We also include an animated sequence in the Supplementary Material.

The term *wrinkle maps* was first used by early VFX approaches to refer to artist-defined bump or normal maps for simulating animated wrinkles [27–30]. However, these approaches suffer from three drawbacks. First, the bump and normal maps only *simulate* underlying geometry changes; the silhouette and shadows which are of relevance for face related tasks such as landmark localization remain unaffected. Second, the methods do not affect the albedo or diffuse textures. Finally, the most crucial drawback is scale. The methods entail manual definition of wrinkle maps and masks for every blendshape for every character. In contrast, our automatic mesh-tension driven method naturally scales with the number of identities and expressions, while incorporating real wrinkles for both albedo and displacement textures from scans. Furthermore, we also handle identities without expression scans, transferring plausible wrinkles from the most similar neutral textures.

To advance the development of synthetics for face-related tasks, we make the following concrete contributions:

- A system for dynamic, expression-based wrinkles that scales easily with increasing identities and expressions.
- A demonstration of empirical qualitative and quantitative improvement over the SOTA synthetics system on face-keypoint localization and surface-normal estimation.
- Novel evaluation data and metrics for keypoint localization in the eye region where wrinkles are especially relevant for learning tasks.

2. Background: Synthesizing Faces

We build upon the work of Wood et al. [1] for synthesizing face images for downstream machine learning tasks. Their method involved sampling from a generative 3D blendshape-based face model learned from 3D scans of 511 individuals with neutral expression. The sampled face is then *dressed up* with samples from a large collection of hair, clothing, and accessory assets. For each synthesized face, the authors employ three textures that remain fixed across all expressions: one albedo map for skin color; one coarse displacement map to encode scan geometry not captured by the sparsity of the vertex-level identity model; and one meso-displacement map to approximate skin-pore level detail built by high-pass filtering the albedo texture. In contrast, we automatically compute an additional sets of albedo and displacement wrinkle textures from expression scans to support dynamic wrinkling effects.

3. Related Work

Wrinkle Maps. Oat [27] proposed using a pair of bump maps to render animated wrinkles on virtual characters. These bump maps—called *wrinkle maps*—store surface normals for an expanded (or stretched) and compressed (or *scrunched-up*) expression, typically obtained from artist sculpted high-resolution meshes. A base normal map stores fine surface details such as pores. In order to achieve independently controlled wrinkles, the face is divided into multiple regions. Each region is specified by an artist-defined mask stored in a texture map. An animated scalar wrinkle weight in the range $[-1, 1]$ then interpolates between the two wrinkle maps for each masked region: at either end of the range one of the wrinkle maps is at its full influence, with a weight of 0 corresponding to no influence on the base normal map. A similar method was later independently proposed by Duque Reis et al. [31] using a single wrinkle map. Jimenez et al. [29] expanded on the scheme proposed by Oat [27], allowing for the use of any number of wrinkle maps, with a weight in the range of $[0, 1]$ defining the influence of each map. Subsequent improvements to make the technique amenable in real-time or performance driven settings involved the dynamic generation of either the region masks [30] or the wrinkle weights [28]. Both approaches relied on using a *skinned* mesh attached to bones. Dutreve et al. [30] proposed generating dynamic region masks by using

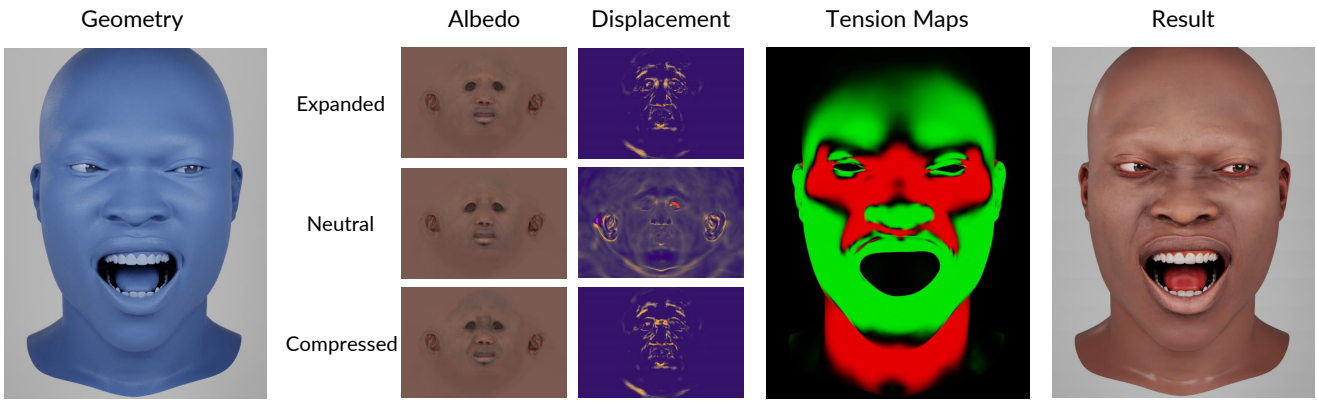

Figure 2: **Method Overview.** The state-of-the-art method for face synthetics [1] generates albedo and displacement textures using only the neutral-expression scan for an identity (middle row, also see Figure 1). In contrast, we automatically compute expanded and compressed texture maps to aggregate wrinkling effects in the face and neck regions across available posed-expression scans for the identity. At synthesis, for a given set of arbitrary expression parameters we compute the local tension at every vertex in the corresponding face mesh: we depict expansion in green and compression in red. This mesh tension serves as weights to dynamically blend between the neutral, expanded, and compressed texture maps to synthesize the wrinkling effect at that vertex. Note that our method can thereby generate wrinkles for expressions even beyond those represented in the source scans.

the bone influence weights from a set of artist defined reference poses. Oat [28] proposed generating dynamic wrinkle weights by comparing each mesh triangle's area before and after skinning, a technique derived from Microsoft's DirectX 10 Sparse Morph Targets demo [32]. While the term *wrinkle maps* in literature has been alternatively used to refer to bump or normal maps, in this work we use the term to collectively refer to the textures used for synthesizing wrinkles: the albedo and displacement maps corresponding to the expanded and compressed textures.

Simulation Based Approaches. While the use of wrinkle maps is the most common methodology when artistic control is of importance, several alternate techniques have been proposed for simulating wrinkles on 3D surfaces. These methods can broadly be grouped into physical and geometric simulation of wrinkles. An early physical simulation based approach employed a biomechanical perspective, considering the skin as an elastic membrane and modeling the deformations using linear plastic model [33]. Boissieux et al. [34] extended the elastic membrane perspective by modeling the skin as a volumetric substance comprising layers of different materials and using a finite element method for computing deformations. Finite element modeling was also employed in subsequent works to simulate forearm skin wrinkling [35], and skin aging [36]. Wang et al. [37] and Venkataraman et al. [38] proposed energy based approaches. Here, wrinkle deformations are produced by minimizing an energy function indicating flexure properties of a governing curve on a surface. To produce wrinkles on dynamic meshes such as simulated cloth, Müller and Chentanez [39] proposed attaching a higher resolution wrinkle mesh to the coarse base mesh and determining the deviations of the wrinkle mesh vertices using a static solver [40]. Geometric simulation based approaches typically involve expressing the wrinkles using some geometric primitives. Bando et al. [41] represented wrinkles using a cubic Bezier curve, generating their furrows from a sequence of starting points along a user specified direction field. Other proposed techniques involved the use of length preserving constraints on planar curves along with artist placed features at locations on an animated mesh where wrinkling is desired [42, 43]. Ilie et al. [44] employed a Hermite spline interpolation along with a modified Rayleigh distribution function to simulate wrinkling activity in facial animations. Subsequent methods extracted wrinkle curves automatically from images [45, 46]. Finally, Gui et al. [47] used both a muscle model and a geometric wrinkle shape function to simulate 3D facial wrinkles.

Machine Learning Approaches. More recently, several methods for expression and texture synthesis, and facial performance capture have addressed the synthesis of wrinkles. As part of their performance capture system, Cao et al. [48] trained regressors for mapping local image appearance to wrinkle displacements to augment a coarse face mesh tracked in real-time. Zeng et al. [24] and Richardson et al. [22] proposed convolutional networks based refinement architectures to reconstruct detailed facial geometry from a single image. Nagano et al. [49] proposed a

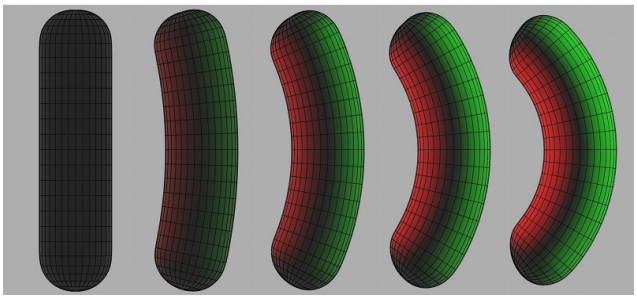

Figure 3: **Mesh Tension.** We illustrate our computation of mesh tension for various deformations of a simple cylinder. Expansion is depicted in green and compression in red. Black shading corresponds to zero tension.

Figure 4: **Data—High-resolution 3D Scans.** For each identity, we illustrate: the raw neutral scan (top-left), the manually-cleaned neutral scan to remove sensor noise and hair (top-right), and two raw expression scans (bottom).

conditional generative adversarial network architecture for the synthesis of image-based dynamic 3D avatars. Given a single neutral-face input image, their system can generate novel photo-real expressions from alternate viewpoints, including variable details such as wrinkles. More directly, Deng et al. [50] proposed a variational autoencoder architecture to synthesize plausible fine-scale wrinkles on a variety of coarse-scale 3D faces.

4. Synthethizing Expression-Based Wrinkles

Figure 2 illustrates an overview of our approach. The underlying idea is that wrinkles can by synthesized additively over the neutral-expression textures. We formalize the concept of mesh tension and use it to automatically aggregate wrinkling effects in a data-driven manner across all expression scans of an identity. We store these possible wrinkles corresponding to the expansion and compression deformations of the face in separate albedo and displacement textures, which we collectively refer to as wrinkle maps in this work. Note that displacement maps modify the underlying geometry unlike bump or normal maps that simply simulate the geometry changes. At synthesis, we sample a face mesh from a generative face model [1] and randomly select a set of neutral and wrinkle textures corresponding to an identity from the available scans. We then compute the tension in the face mesh to drive the blending between the neutral and wrinkle maps to obtain dynamic wrinkling effects. In contrast with previous learning-based wrinkling methods [22–24, 50], we do not build a generative model for the textures since such models struggle to reconstruct high frequency details such as wrinkles compared to directly extracting them from scans.

4.1. Mesh Tension

We formalize mesh tension to capture the amount of compression or expansion at each vertex of a 3D polygon mesh resulting from a deformation. More concretely, we express mesh tension as a function of the mean change in the length of the edges connected to a vertex as a result of the deformation. Consider an undeformed mesh $\overline{\mathbf{X}} = (\overline{V}, \overline{E})$ with a sequence of vertices $\overline{V}$ and sequence of edges $\overline{E}$, that undergoes a deformation to result in the mesh $\mathbf{X} = (V, E)$. We only consider deformations such that $\overline{\mathbf{X}}$ and $\mathbf{X}$ possess the same topology. For vertex $v_i \in V$, let $(e_1, \ldots, e_K)$ denote the sequence of K edges connected to v_i, with $(\bar{e}_1, \ldots, \bar{e}_K)$ denoting the corresponding edges in $\overline{X}$ connected to $\bar{v}_i$. We then define the mesh tension at v_i as

$$t_{v_i} := 1 - \frac{1}{K} \sum_{k \in [K]} \frac{\|e_k\|}{\|\bar{e}_k\|}, \tag{1}$$

where $[K] = \{1, \ldots, K\}$, and $\|.\|$ denotes edge length. Note that we subtract from 1 so that positive values of t_{v_i} indicate compression, negative values indicate expansion, and a value of 0 indicates no change.

In practice, for finer manual control we introduce the parameters of strength s to scale the tension, and bias b to artificially favor expansion or compression, computing the weighted tension at v_i as $t'_{v_i} = s \cdot t_{v_i} + b$. Further, we allow for artificial propagation of expansion and compression effects through the mesh. For each effect we introduce a parameter denoting the number of iterations for a morphological dilation (positive values) or erosion (negative values) operation. The propagation of each effect is first performed independently over the mesh, and the resulting tension values are added for vertices that end up with both expansion and compression. Figure 3 illustrates these effects for a simple cylindrical mesh. See Appendix A for additional illustrations of the effect of the tension parameters. Code as a Blender [51] add-on is available at `https://github.com/chiragraman/mesh-tension`

4.2. Data and Preprocessing

We start with a set of high-quality commercially available 3D scans of 208 individuals. All 208 identities contain scans with neutral expressions, while 52 contain additional

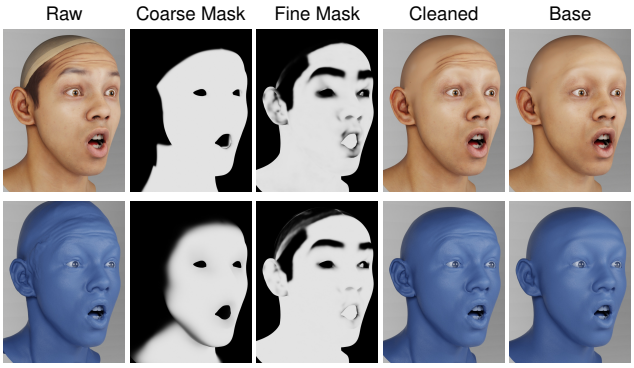

Figure 5: **Cleaning Raw Textures.** We illustrate the cleanup of albedo (top) and displacement (bottom) textures on the *surprise* expression. We automatically remove the hair and sensor noise artifacts in the raw textures around the head, neck, and cheeks while preserving the desired wrinkles in the nose, forehead, and mouth regions (compared to the base mesh, with neutral albedo and without displacement respectively, for the same expression).

scans for posed expressions. The neutral scans were manually cleaned for removing noise and hair artifacts, and registered to the topology of the 3D face model proposed by Wood et al. [1], resulting in a mesh of $7,667$ vertices and $7,414$ polygons. Figure 4 illustrates the scans.

Automatic Cleaning of Expression Scans. The manual cleaning of scans is a labor-intensive process. To automate the process of masking the noise and hair artifacts from the expression scans, we utilize the difference between the raw and manually-cleaned neutral scans. Concretely, we employ a two-stage masking procedure illustrated in Figure 5. First, we apply an identity-agnostic coarse mask to filter most artifacts outside of the hockey-mask and neck regions where expression-based wrinkling occurs. Next, to capture the manual changes made by the artists in the cleaning of each neutral scan, we employ a Gaussian Mixture Model-based background subtraction technique [52]. Treating the clean neutral textures as background and the raw original ones as foreground, we obtain an identity-specific mask of the noise and hair artifacts for every identity. We apply this fine mask to clean the textures from the corresponding expression scans for each identity.

4.3. Data-Driven Wrinkle Maps

Tension-Weighted Wrinkle Maps. Figure 6 illustrates our method for generating wrinkle maps from the face scans. Our underlying idea is to use the tension at each vertex as weights in a linear combination of the cleaned textures across expressions, with zero tension corresponding to the

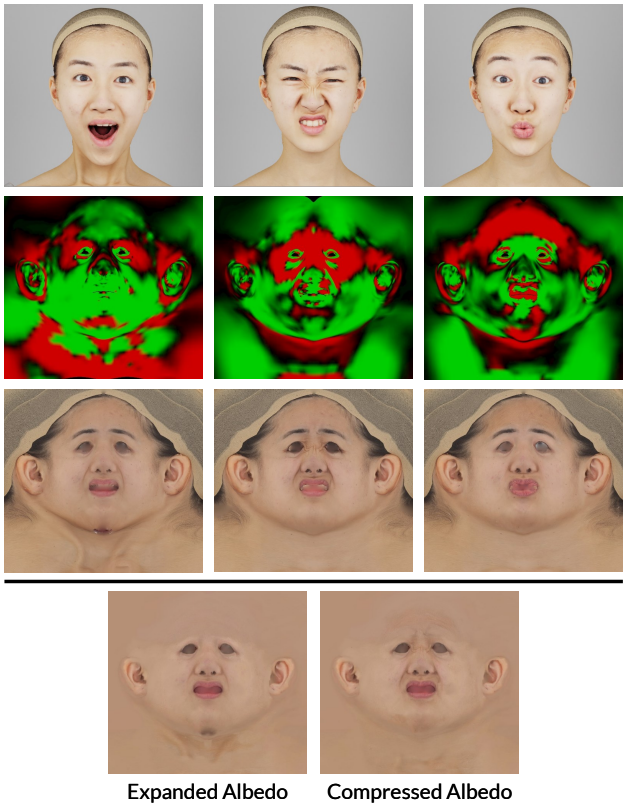

Figure 6: **Generating Wrinkle Maps from Scans.** We illustrate the computation of albedo wrinkle maps with three raw expression scans (top). We compute the tension maps corresponding to the scans (middle), depicting expansion in green and compression in red. Finally, the expression albedo textures (bottom) are linearly combined using the normalized tension as weights to obtain the expanded and compressed albedo wrinkle maps. A similar procedure is applied to obtain the displacement wrinkle maps.

neutral textures. (Figure 6 depicts raw textures for easier visual correspondence with the scans.) We begin by fitting the generative face model from Wood et al. [1] to the raw scans and compute the tension maps from the resulting meshes. The individual expansion and compression maps are then normalized using the $\mathrm{softmax}$ function. Finally, we linearly combine expression textures using the normalized tension as weights to obtain the expanded and compressed wrinkle maps. The same procedure is applied to obtain both the albedo and displacement wrinkle maps.

Identities With Missing Expression Scans. How do we compute wrinkle maps for the identities without posed expression scans? We employ a simple wrinkle-grafting procedure. For a target identity without wrinkle maps, we find the source identity with wrinkle maps that has the most sim-

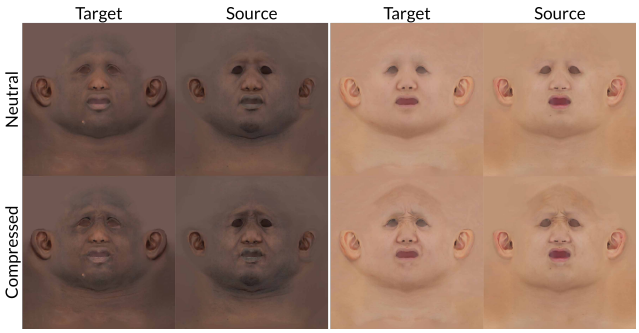

Figure 7: **Grafting Wrinkles.** For an identity with missing expression scans (target), we find the identity from among those with expression scans that has the most similar neutral albedo map (source). We then graft the wrinkles from the source's wrinkle map onto the target's neutral texture to obtain the target wrinkle maps (here illustrating the compressed albedo).

Table 1: **Landmark Localization on 300W.** We normalize mean error using interocular distance. Lower is better.

Method	Common NME	Challenging NME	Private $FR_{10\%}$
Trained on Real Data			
LAB [53]	2.98	5.19	0.83
AWING [54]	**2.72**	**4.52**	0.33
ODN [55]	3.56	6.67	-
3FabRec [56]	3.36	5.74	**0.17**
LUVLi [57]	2.76	5.16	-
Trained on Synthetic Data			
No wrinkles [1]	3.11	4.84	0.33
Ours (wrinkles)	3.10	4.83	**0.17**

ilar neutral albedo map, measured by mean squared error in pixel color. For the source identity, we compute the wrinkling effects as the difference between the neutral and wrinkle maps (for both albedo and displacement). We then add this difference to the neutral textures for the target identity to obtain the target wrinkle maps. We illustrate the grafting procedure for the compressed albedo maps in Figure 7, and final example renders with grafted wrinkles in Figure 8.

5. Experiments and Results

We evaluate our proposed mesh-tension driven wrinkles both quantitatively and qualitatively on two face analysis tasks: landmark detection (Section 5.1) and normal estimation (Section 5.2). We compare adding mesh-tension to the existing SOTA method for full-face synthetics, and compare the performance of models trained on the resulting data against SOTA approaches in the field for these tasks.

Figure 8: **Final Renders for Some Identities with Grafted Wrinkles.** We computed the wrinkle maps for these identities by grafting wrinkles from identities with expression scans (see Figure 7). We illustrate two expressions for each identity, without (left) and with wrinkles (right).

5.1. Landmark Localization

Experimental Details. We use direct regression based facial landmark detection [58] with an off-the-shelf ResNet 101 [59]. We use a 256×256 px RGB image as input to predict 703 dense facial landmarks. We additionally employ label translation [1] to deal with systematic inconsistencies between our 703 predicted dense landmarks and the 68 sparse landmarks labeled as ground truth in our evaluation datasets (this is done only for Table 1).

As a training dataset we rendered $100k$ synthetic images, consisting of $20k$ identities with 5 frames for each identity (different view-points, expressions, and environments). We also generated ground-truth annotations of 703 dense 2D landmarks from the face-meshes to accompany each image. We train our models for 300 epochs using PyTorch Lightning, starting with a learning rate of $1e{-}3$ and halved every 100 epochs.

Evaluation Datasets and Metrics. We use the **300W** dataset [60] (with common, challenging and private subsets), and employ the standard normalized mean error (NME) and failure rate ($FR_{10\%}$) error metrics [60].

While the 300W dataset provides evaluation of overall

Table 2: **Landmark Localization - Eyes.** We report eye-opening errors for Pexels, and eyelid point-to-polyline errors for 300W and the *winks* subset. In all cases normalized by bounding-box diagonal. Lower is better.

Method	Pexels	300W	300W-winks
Trained on Real Data			
AWING [54]	1.06	0.62	**0.69**
3FabRec [56]	3.60	0.81	1.32
Trained on Synthetic Data			
No wrinkles [1]	0.97	0.51	0.86
Ours (wrinkles)	**0.86**	**0.48**	0.74

landmark detection performance, it is not sensitive enough to detect improvements in specific parts of the face or during particular expressions. We identify a small subset of 30 images from 300W that contain winks and compressed face expressions (**300W-winks**) to provide a more nuanced indication of performance under such deformations. We report errors for eyelid-landmarks by taking a point-to-line distance from every predicted eyelid landmark to the corresponding polyline defining an eyelid in ground truth. This metric allows us to better understand eye region error and to use different landmark definitions in training and evaluating models (e.g. from our 703 landmark model or from 98 landmark models [54]). See Appendix E for the list of images in 300W-winks.

We also introduce a **Pexels** dataset which contains 318 images of fully closed eyes (because of blinking, scrunching or compressing the face) and 105 images with only a single eye closed (winking). This allows us to asses model performance under such conditions which are rare in other datasets. To collect the data we used a stock photography website [1] using search terms *wink/blink/compress/scrunched* and similar image searches. We select only semi-frontal images with no or limited occlusion of the eyes to best evaluate performance in that region. The URLs of the images selected can be found in Appendix F. Knowing which images contain fully closed eyes or just a single eye closed allows us to measure eyelid accuracy without explicit landmark annotations. We define the eye opening error as the mean eye aperture of both eyes in the *eye-closed* case and eye aperture of closed eye in the *wink* case. See Appendix B for illustrations of the above two metrics.

Baselines. We compare against recent SOTA methods trained on images of real faces. For subsequent nuanced analysis on 300W-winks and Pexels we consider the methods of Wang et al. [54] and Browatzki and Wallraven [56] since they collectively yield the best performance on 300W.

Results. From Table 1 we see that our proposed mesh-

[1] https://www.pexels.com/

Figure 9: **Qualitative results for landmark localization on Pexels.** Training on synthetic faces with our expression-based wrinkles is crucial for localizing keypoints in compressed regions of the face.

Table 3: **Landmark Localization Ablation.** We report eye-opening errors for Pexels, and eyelid point-to-polyline errors for 300W and the *winks* subset. Lower is better.

Dataset	Base	Disp. Only	Albedo Only	Full
300W	0.51	0.51	0.50	**0.48**
300W-winks	0.86	0.76	0.80	**0.74**
Pexels	0.97	**0.86**	0.89	**0.86**

tension driven wrinkles provide a marginal improvement for landmark localization. However, when we look at specific eye region results on 300W, 300W-winks and Pexels in Table 2, we see that improvement is much larger for the eye region and our synthetic-only trained approaches outperform real-data based models. Also see Figure 9 and Appendix C.

Ablation. We further analyze the importance of the albedo and displacement wrinkling components for landmark detection. From Figure 10 and Table 3 we see that displacement plays a more important role than albedo in improving performance, but best results are achieved through a combination of both.

Figure 10: **Expression-Based Wrinkle Components.** We add wrinkles through two components: displacement and albedo. Here we show each in isolation. Displacement is critical for achieving realistic lighting of wrinkles. Especially note the forehead (zoomed) and neck regions.

Figure 11: **Qualitative Surface-Normals Predictions on Pexels.** The model trained on synthetic faces with wrinkles recovers significantly more high-frequency details.

5.2. Surface-Normals Prediction

Surface normals can be used to infer 3D information about a surface from 2D images, and have been used in several human-centered vision tasks such as clothing [61] and face-shape [62] reconstruction and relighting [63].

We train a U-Net [64] with a ResNet 18 [59] encoder to predict camera-space surface normals of the face. As input we use 256×256 px RGB images from a dataset of $50k$ synthetics images. The network is trained for 200 epochs using cosine similarity loss with a learning rate of $1e{-}3$. Camera-space surface normal images rendered as part of our synthetic data pipeline are used as ground-truth.

Results on real images are shown in Figure 11; the network trained on images synthesized with our method recovers more high-frequency detail on the face. As shown in Figure 12, we achieve comparable results to other recent methods for face surface-normals prediction [62, 63]. Further comparisons are provided in Appendix D.

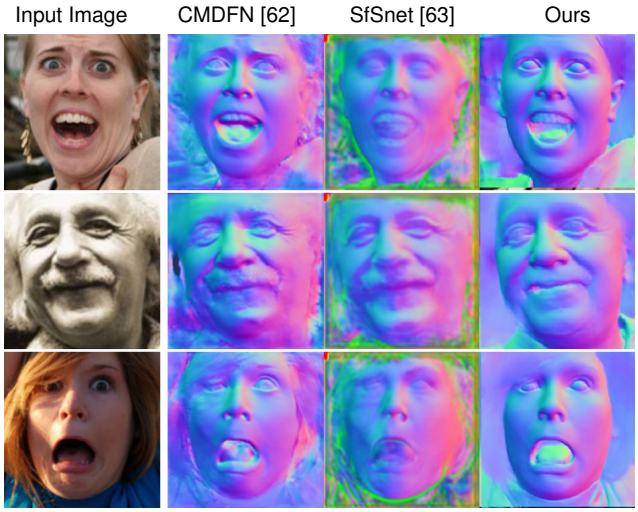

Figure 12: **Qualitative Comparison against SOTA**. Our synthetic data-only U-Net yields predictions comparable to SOTA while being less noisy and more robust to lighting.

6. Conclusion

We have presented a method for introducing dynamic expression-based wrinkles to synthetic faces that yields improved performance on the downstream tasks of landmark localization and surface-normals estimation, especially for regions of the face most deformed by expressions.

Our use of tension in the face mesh is key in the automatic scaling of our method with identities and expressions, which has been a bottleneck for past wrinkling approaches that rely on prohibitive artist effort. In addition, our data-driven approach also enables the capturing of real wrinkles from scans which doesn't require artistic judgment.

By boosting the realism of synthesized faces with dynamic wrinkles, we have made an explicit case for synthetic data: our method yields improved performance for models on downstream tasks. In addition, synthesizing data with diverse faces across races and genders involves significantly less effort than collecting representative datasets in the wild. Consequently, downstream real-life systems developed using such synthetic data are less likely to suffer from unfair biases along these sensitive variables.

Acknowledgments

Chirag would like to thank: Tom Cashman, Stephan Garbin, and Panagiotis Giannakopoulos for the insightful discussions; Sebastian Dziadzio for help with fitting the face model; and Sarah Roberts for being an infallible remover of obstacles.

References

[1] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake It Till You Make It: Face analysis in the wild using synthetic data alone. *arXiv:2109.15102 [cs]*, October 2021.

[2] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating Content Consistent Vehicle Datasets with Attribute Descent. *arXiv:1912.08855 [cs]*, July 2020.

[3] Tomas Hodan, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N. Sinha, and Brian Guenter. Photorealistic Image Synthesis for Object Instance Detection. In *arXiv:1902.03334 [Cs]*, pages 66–70, February 2019.

[4] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. UnrealCV: Virtual Worlds for Computer Vision. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, pages 1221–1224, New York, NY, USA, October 2017. Association for Computing Machinery. ISBN 978-1-4503-4906-2. doi: 10.1145/3123266.3129396.

[5] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On Rendering Synthetic Images for Training an Object Detector. *Computer Vision and Image Understanding*, 137:24–37, August 2015. ISSN 10773142. doi: 10.1016/j.cviu.2014.12.006.

[6] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-Sim: Learning to Generate Synthetic Datasets. *arXiv:1904.11621 [cs]*, April 2019.

[7] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. *arXiv:1605.06457 [cs, stat]*, May 2016.

[8] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. *arXiv:1608.02192 [cs]*, August 2016.

[9] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.352.

[10] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *arXiv:1505.05916 [cs]*, May 2015.

[11] Lech Świrski and Neil Dodgson. Rendering synthetic ground truth images for eye tracker evaluation. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 219–222, Safety Harbor Florida, March 2014. ACM. ISBN 978-1-4503-2751-0. doi: 10.1145/2578153.2578188.

[12] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated Hands for Real-time 3D Hand Tracking from Monocular RGB. *arXiv:1712.01057 [cs]*, December 2017.

[13] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. *arXiv:1704.07809 [cs]*, April 2017.

[14] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4627–4635, July 2017. doi: 10.1109/CVPR.2017.492.

[15] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, June 2011. doi: 10.1109/CVPR.2011.5995316.

[16] Huazhong Ning, Wei Xu, Yihong Gong, and Thomas Huang. Discriminative learning of visual words for 3D human pose estimation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, USA, June 2008. IEEE. ISBN 978-1-4244-2242-5. doi: 10.1109/CVPR.2008.4587534.

[17] Darren Hendler, Lucio Moser, Rishabh Battulwar, David Corral, Phil Cramer, Ron Miller, Rickey Cloudsdale, and Doug Roble. Avengers: Capturing thanos's complex face. In *ACM SIGGRAPH 2018 Talks*, pages 1–2, Vancouver British Columbia Canada, August 2018. ACM. ISBN 978-1-4503-5820-0. doi: 10.1145/3214745.3214766.

[18] Karis, Brian, Antoniades, Tameem, Caulkin, Steve, and Mastilovic, Vladimir. Digital Humans: Crossing the Uncanny Valley in UE4. In *Game Developers Conference*, 2016.

[19] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138, Charleston South Carolina, March 2016. ACM. ISBN 978-1-4503-4125-7. doi: 10.1145/2857491.2857492.

[20] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1821–1828, Columbus, OH, USA, June 2014. IEEE. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.235.

[21] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation. *arXiv:1703.10131 [cs]*, March 2017.

[22] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning Detailed Face Reconstruction from a Single Image. *arXiv:1611.05053 [cs]*, April 2017.

[23] Elad Richardson, Matan Sela, and Ron Kimmel. 3D Face Reconstruction by Learning from Synthetic Data. *arXiv:1609.04387 [cs]*, September 2016.

[24] Xiaoxing Zeng, Xiaojiang Peng, and Yu Qiao. DF2Net: A Dense-Fine-Finer Network for Detailed 3D Face Reconstruction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2315–2324, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00240.

[25] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. *arXiv:1612.07828 [cs]*, July 2017.

[26] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *arXiv:1505.07818 [cs, stat]*, May 2016.

[27] Christopher Oat. Animated wrinkle maps. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, pages 33–37, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 978-1-4503-1823-5. doi: 10.1145/1281500.1281667.

[28] Christopher Oat. Real-Time Wrinkles, 2007.

[29] Jorge Jimenez, Jose I. Echevarria, Christopher Oat, and Diego Gutierrez. Practical and Realistic Facial Wrinkles Animation. In Wolfgang Engel, editor, *GPU pro 2*, chapter Practical and Realistic Facial Wrinkles Animation. AK Peters Ltd., 2011.

[30] Ludovic Dutreve, Alexandre Meyer, and Saida Bouakaz. Real-Time Dynamic Wrinkles of Face for Animated Skinned Mesh. In *ISVC' 09: 5th International Symposium on Visual Computing*, Advances in Visual Computing, pages 25–34, Las Vegas, USA, United States, November 2009. Springer. doi: 10.1007/978-3-642-10520-3_3.

[31] Clausius Duque Reis, G Reis, José De Martino, and Harlen Batagelo. *Real-Time Simulation of Wrinkles*. February 2008.

[32] Microsoft DirectX 10 SDK Team. Sparse Morph Targets Sample, 2007.

[33] Yin Wu, Prem Kalra, and Nadia Magnenat Thalmann. Physically-based Wrinkle Simulation & Skin Rendering. In W. Hansmann, W. T. Hewitt, W. Purgathofer, Daniel Thalmann, and Michiel van de Panne, editors, *Computer Animation and Simulation '97*, pages 69–79. Springer Vienna, Vienna, 1997. ISBN 978-3-211-83048-2 978-3-7091-6874-5. doi: 10.1007/978-3-7091-6874-5_5.

[34] Laurence Boissieux, Gergo Kiss, Nadia Magnenat Thalmann, and Prem Kalra. Simulation of Skin Aging and Wrinkles with Cosmetics Insight. In W. Hansmann, W. Purgathofer, F. Sillion, Nadia Magnenat-Thalmann, Daniel Thalmann, and Bruno Arnaldi, editors, *Computer Animation and Simulation 2000*, pages 15–27. Springer Vienna, Vienna, 2000. ISBN 978-3-211-83549-4 978-3-7091-6344-3. doi: 10.1007/978-3-7091-6344-3_2.

[35] Cormac Flynn and B. A. O. McCormack. Finite element modelling of forearm skin wrinkling. *Skin Research and Technology*, 14, 2008.

[36] Cormac Flynn and Brendan A. O. McCormack. Simulating the wrinkling and aging of skin with a multi-layer finite element model. *Journal of Biomechanics*, 43(3):442–448, February 2010. ISSN 0021-9290. doi: 10.1016/j.jbiomech.2009.10.007.

[37] Yu Wang, Charlie CL Wang, and Matthew MF Yuen. Fast energy-based surface wrinkle modeling. *Computers & Graphics*, 30(1):111–125, 2006.

[38] Kartik Venkataraman, Suresh Lodha, and Raghu Raghavan. A kinematic-variational model for animating skin with wrinkles. *Computers & Graphics*, 29(5):756–770, October 2005. ISSN 0097-8493. doi: 10.1016/j.cag.2005.08.024.

[39] Matthias Müller and Nuttapong Chentanez. Wrinkle meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 85–92, 2010.

[40] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, April 2007. ISSN 1047-3203. doi: 10.1016/j.jvcir.2007.01.005.

[41] Y. Bando, T. Kuratate, and T. Nishita. A simple method for modeling wrinkles on human skin. In *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pages 166–175, Beijing, China, 2002. IEEE Comput. Soc. ISBN 978-0-7695-1784-1. doi: 10.1109/PCCGA.2002.1167852.

[42] Caroline Larboulette and Marie-paule Cani. Real-Time Dynamic Wrinkles, 2004.

[43] Ming Li, BaoCai Yin, DeHui Kong, and XiaoNan Luo. Modeling Expressive Wrinkles of Face For Animation. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 874–879, August 2007. doi: 10.1109/ICIG.2007.22.

[44] Mihai Daniel Ilie, Cristian Negrescu, and Dumitru Stanomir. A robust mathematical model for simulating wrinkle activity

in 3D facial animations. In *2012 10th International Symposium on Electronics and Telecommunications*, pages 271–274, November 2012. doi: 10.1109/ISETC.2012.6408082.

[45] Li Li, Fei Liu, Congbo Li, and Guoan Chen. Realistic wrinkle generation for 3D face modeling based on automatically extracted curves and improved shape control functions. *Computers & Graphics*, 35(1):175–184, February 2011. ISSN 00978493. doi: 10.1016/j.cag.2010.08.003.

[46] Ron Vanderfeesten and Jacco Bikker. Example-Based Skin Wrinkle Displacement Maps. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 212–219, Parana, October 2018. IEEE. ISBN 978-1-5386-9264-6. doi: 10.1109/SIBGRAPI.2018.00034.

[47] Jiamin Gui, Yue Zhang, and Shaobin Li. Realistic 3D Facial Wrinkles Simulation Based on Tessellation. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 250–254, December 2016. doi: 10.1109/ISCID.2016.1064.

[48] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics*, 34(4):1–9, July 2015. ISSN 0730-0301, 1557-7368. doi: 10.1145/2766943.

[49] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. paGAN: Real-time avatars using dynamic textures. *ACM Transactions on Graphics*, 37(6):1–12, January 2019. ISSN 0730-0301, 1557-7368. doi: 10.1145/3272127.3275075.

[50] Qixin Deng, Luming Ma, Aobo Jin, Huikun Bi, Binh Huy Le, and Zhigang Deng. Plausible 3D Face Wrinkle Generation Using Variational Autoencoders. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2021. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2021.3051251.

[51] Blender. `https://www.blender.org/`. Accessed: 2021-09-30.

[52] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7): 773–780, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.11.005.

[53] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2138, 2018.

[54] Xinyao Wang, Liefeng Bo, and Li Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6971–6981, 2019.

[55] Meilu Zhu, Daming Shi, Mingjie Zheng, and Muhammad Sadiq. Robust facial landmark detection via occlusion-adaptive deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3486–3496, 2019.

[56] Bjoern Browatzki and Christian Wallraven. 3fabrec: Fast few-shot face alignment by reconstruction. In *CVPR*, 2020.

[57] Abhinav Kumar, Tim K Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8236–8246, 2020.

[58] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljevic, Daniel Wilde, Stephan Garbin, Toby Sharp, Ivan Stojiljkovic, Tom Cashman, and Julien Valentin. 3d face reconstruction with dense landmarks, 2022.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[60] C Sagonas, E. Antonakos, G Tzimiropoulos, S Zafeiriou, and M Pantic. 300 faces In-the-wild challenge: Database and results. *Image and Vision Computing (IMAVIS)*, 2016.

[61] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019.

[62] Victoria Fernández Abrevaya, Adnane Boukhayma, Philip HS Torr, and Edmond Boyer. Cross-modal deep face normals with deactivable skip connections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4979–4989, 2020.

[63] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6296–6305, 2018.

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.