

# Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation

Matan Sela      Elad Richardson      Ron Kimmel  
 Department of Computer Science, Technion - Israel Institute of Technology  
 {matansel, eladrich, ron}@cs.technion.ac.il



Figure 1: Results of the proposed method. Reconstructed geometries are shown next to the corresponding input images.

## Abstract

*It has been recently shown that neural networks can recover the geometric structure of a face from a single given image. A common denominator of most existing face geometry reconstruction methods is the restriction of the solution space to some low-dimensional subspace. While such a model significantly simplifies the reconstruction problem, it is inherently limited in its expressiveness. As an alternative, we propose an Image-to-Image translation network that maps the input image to a depth image and a facial correspondence map. This explicit pixel-based mapping can then be utilized to provide high quality reconstructions of diverse faces under extreme expressions. In the spirit of recent approaches, the network is trained only with synthetic data, and is then evaluated on “in-the-wild” facial images. Both qualitative and quantitative analyses demonstrate the accuracy and the robustness of our approach. As an additional analysis of the proposed network, we show that it can be used as a geometric constraint for facial image translation tasks.*

## 1. Introduction

Recovering the geometric structure of a face is a fundamental task in computer vision with numerous applications. For example, facial characteristics of actors in realistic movies can be manually edited with facial rigs that are carefully designed for manipulating the expression [39]. While producing animation movies, tracking the geometry

of an actor across multiple frames allows transferring the expression to an animated avatar [12, 7, 6]. Image-based face recognition methods deform the recovered geometry for producing a neutralized and frontal version of the input face in a given image, reducing the variations between images of the same subject [46, 17]. As for medical applications, acquiring the structure of a face allows for fine planning of aesthetic operations and plastic surgeries, designing of personalized masks and even bio-printing facial organs.

Here, we focus on the recovery of the geometric structure of a face from a single facial image under a wide range of expressions and poses. This problem has been investigated for decades and most existing solutions involve one or more of the following components.

- Facial landmarks [23, 43, 30, 44] - a set of automatically detected key points on the face such as the tip of the nose and the corners of the eyes, which can then guide the reconstruction process [46, 24, 1, 10, 27].
- A reference facial model - a typical neutral face that is used as an initialization of optical flow or shape from shading procedures [17, 24].
- A three-dimensional morphable model - a prior low-dimensional linear subspace of plausible facial geometries which allows an efficient, yet rough, recovery of a facial structure [3, 5, 46, 34, 21, 31, 40],

While using these components can simplify the reconstruction problem, they introduce some inherent limitations.

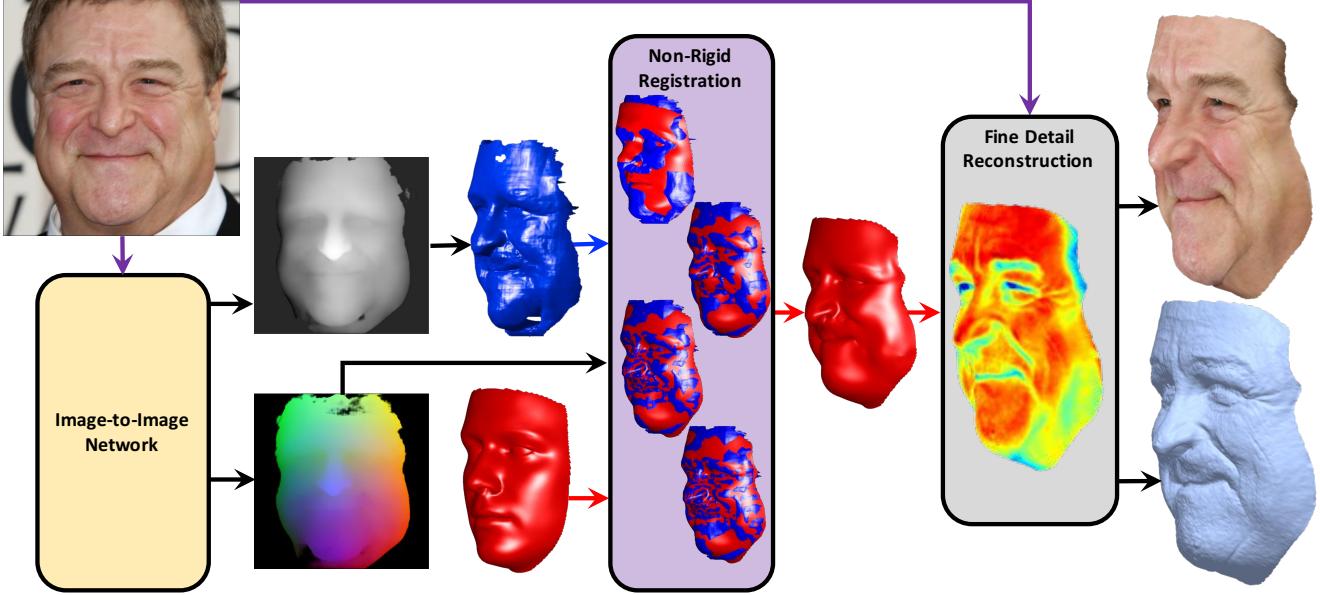


Figure 2: The algorithmic reconstruction pipeline.

Methods that rely only on landmarks are limited to a sparse set of constrained points. Classical techniques that use a reference facial model might fail to recover extreme expressions and non-frontal poses, as optical flows restrict the deformation to the image plane. The morphable model, while providing some robustness, limits the reconstruction as it can express only coarse geometries. Integrating some of these components together could mitigate the problems, yet, the underlying limitations are still manifested in the final reconstruction.

Alternatively, we propose an unrestricted approach which involves a fully convolutional network that learns to translate an input facial image to a representation containing two maps. The first map is an estimation of a depth image, while the second is an embedding of a facial template mesh in the image domain. Similar to previous approaches, we train the network with synthetic images of a wide range of facial identities, poses, expressions, lighting conditions, backgrounds and material parameters. Surprisingly, even though the network is trained with faces that are drawn from a limited generative model, it can generalize and produce structures far and beyond the limited scope of the model. Based on the resulting dense correspondence and depth images, an iterative facial deformation technique generates high-quality meshes with wrinkles and details at a mesoscopic-level. To further analyze the potential of the proposed representation of the network, we devise an application for translating images from one domain to another. As a case study, we transform synthetic facial images into realistic ones. Enforcing our network as a loss function successfully preserves the geometry throughout the cross do-

main mapping.

While using a neural network for face reconstruction was proposed in the past [31, 32, 40, 45], previous methods were still limited by the expressiveness of the linear model. In [32], a second network was proposed to refine the coarse facial reconstruction, yet, it could not compensate for large geometric variations beyond the given subspace. For example, the structure of the nose was still limited by the span of a facial morphable model. By learning the unconstrained geometry directly in the image domain, we overcome this limitation, as demonstrated by both quantitative and qualitative experimental results.

The contributions of this paper are:

- A network that predicts a geometric representation of a face from a single image, which is not restricted by any model.
- A deformation procedure that produces a high quality facial reconstruction based on the representation provided by the network and the given image.
- A novel application of the proposed network which allows translating synthetic facial images into realistic ones, while keeping the geometric structure intact.

## 2. Overview

The algorithmic pipeline is presented in Figure 2. The input of the network is a facial image, and the network produces two outputs: The first is an estimated depth map aligned with the input image. The second output is a dense

map from each pixel to a corresponding vertex on a reference facial mesh. To bring the results into full vertex correspondence and complete occluded parts of the face, we warp a template mesh in the three-dimensional space by an iterative non-rigid deformation procedure. Finally, a fine detail reconstruction algorithm guided by the input image recovers the subtle geometric structure of the face. Code for evaluation is available at <https://github.com/matansel/pix2vertex>.

### 3. Learning the Geometric Representation

There are several design choices to consider when working with neural networks. First and foremost is the training data, including the input channels, their labels, and how to gather the samples. Second is the choice of the architecture. A common approach is to start from an existing architecture [25, 36, 37, 18] and to adapt it to the problem at hand. Finally, there is the choice of the training process, including the loss criteria and the optimization technique. Next, we describe our choices for each of these elements.

#### 3.1. The Data and its Representation

The purpose of the suggested network is to regress a geometric representation from a given facial image. This representation is composed of the following two components:

**Depth Image** A depth profile of the facial geometry. Indeed, for many facial reconstruction tasks providing only the depth profile is sufficient [16, 24].

**Correspondence Map** An embedding which allows mapping image pixels to points on a template facial model, given as a triangulated mesh. To compute this signature for any facial geometry, we paint each vertex with the  $x$ ,  $y$ , and  $z$  coordinates of the corresponding point on a normalized canonical face. Then, we paint each pixel in the map with the color value of the corresponding projected vertex, see Figure 3. This feature map is a deformation agnostic representation, which is useful for applications such as facial motion capture [41], face normalization [46] and texture mapping [47]. While a similar representation was used

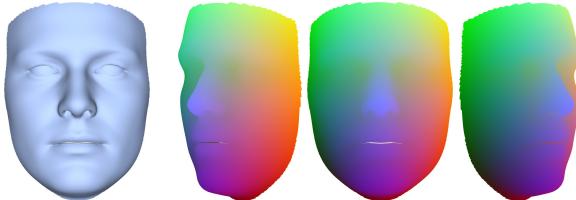


Figure 3: A reference template face presented alongside the dense correspondence signature from different viewpoints.



Figure 4: Training data samples alongside their representations.

in [32, 45] as feedback channel for an iterative network, the facial recovery was still restricted to the span of a facial morphable model.

For training the network, we adopt the same synthetic data generation procedure proposed in [31]. Each random face is generated by drawing random mesh coordinates  $S$  and texture  $T$  from a facial morphable model. In practice, we draw a pair of Gaussian random vectors,  $\alpha_g$  and  $\alpha_t$ , and recover the synthetic face as follows

$$\begin{aligned} S &= \mu_g + A_g \alpha_g \\ T &= \mu_t + A_t \alpha_t. \end{aligned}$$

where  $\mu_g$  and  $\mu_t$  are the stacked average facial geometry and texture of the model, respectively.  $A_g$  and  $A_t$  are matrices whose columns are the bases of low-dimensional linear subspaces spanning plausible facial geometries and textures, respectively. Next, we render the random textured meshes under various illumination conditions and poses. As the ground-truth geometry is known for each synthetic image, one readily has the matching depth and correspondence maps. Some examples of input images alongside their desired outputs are shown in Figure 4.

While working with synthetic data can still present some gaps when generalizing to “in-the-wild” images [8, 31], it provides flexibility in the generation process and has a deterministic connection from an image to its label. Other methods proposed to generate training data by employing existing reconstruction algorithms. For example, Güler *et al.* [14], used the framework introduced in [22] for creating dense correspondence maps to a reference face based on a sparse set of landmarks. However, while avoiding the over-fitting problem, such approaches might be limited to the performance of the algorithms involved in the data generation process.

### 3.2. Image to Geometry Translation

Pixel-wise prediction requires a proper network architecture [28, 15]. The proposed structure is inspired by the recent Image-to-Image translation framework proposed in [20], where a network was trained to map the input image to output images of various types. The architecture used there is based on the U-net [33] layout, where skip connections are used between corresponding layers in the encoder and the decoder.

While in [20] a combination of  $L_1$  and adversarial loss functions were used, in the proposed framework, we chose to omit the adversarial loss. That is because unlike the problems explored in [20], our setup includes less ambiguity in the mapping. Hence, a distributional loss function is redundant, and mainly introduces artifacts. Still, since the basic  $L_1$  loss function favors sparse errors in the depth prediction and does not account for differences between pixel neighborhoods, it is insufficient for producing fine geometric structures, see Figure 5b. Hence, we propose to augment the loss function with the following additional  $L_1$  term, which penalizes the discrepancy between the normals of the reconstructed depth and ground truth.

$$L(\hat{z}, z) = \|\vec{n}(\hat{z}) - \vec{n}(z)\|_1, \quad (1)$$

where  $\hat{z}$  is the recovered depth, and  $z$  denotes the ground-truth depth image. Figure 5 demonstrates the contribution of the latter loss function to the quality of the reconstruction provided by the network. Additional considerations as to the network implementation details and training parameters are given in the supplementary material.

## 4. From Representations to a Mesh

Based on the resulting depth and correspondence we introduce an approach to translate the 2.5D representation to a 3D facial mesh. The procedure is composed of an iterative elastic deformation algorithm (4.1) followed by a fine detail recovery step driven by the input image (B.2). The resulting output is an accurate reconstructed facial mesh with a full vertex correspondence to a template mesh with fixed triangulation. This type of data is helpful for various dynamic

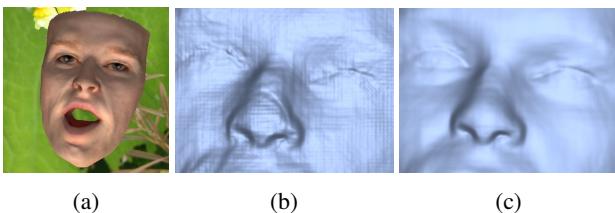


Figure 5: (a) the input image, (b) the result with only the  $L_1$  loss function and (c) the result with the additional normals loss function. Note the artifacts in (b).

facial processing applications, such as facial rigs, which allows creating and editing photo-realistic animations of actors. As a byproduct, this process also corrects the prediction of the network by completing domains in the face which are mistakenly classified as part of the background.

### 4.1. Non-Rigid Registration

Next, we describe the iterative deformation-based registration pipeline. First, we turn the depth map from the network into a mesh, by connecting neighboring pixels. Based on the correspondence map from the network, we compute the affine transformation from a template face to the mesh. This operation is done by minimizing the squared Euclidean distances between corresponding vertex pairs. Next, similar to [26], an iterative non-rigid registration process deforms the transformed template, aligning it with the mesh. Note that throughout the registration, only the template is warped, while the target mesh remains fixed. Each iteration involves the following four steps.

1. Each vertex in the template mesh,  $v_i \in \mathcal{V}$ , is associated with a vertex,  $c_i$ , on the target mesh, by evaluating the nearest neighbor in the embedding space. This step is different from the method described in [26], which computes the nearest neighbor in the Euclidean space. As a result, the proposed step allows registering a single template face to different facial identities with arbitrary expressions.
2. Pairs,  $(v_i, c_i)$ , which are physically distant and those whose normal directions disagree are detected and ignored in the next step.
3. The template mesh is deformed by minimizing the following energy

$$\begin{aligned} E(V, C) = & \alpha_{p2point} \sum_{(v_i, c_i) \in \mathcal{J}} \|v_i - c_i\|_2^2 \\ & + \alpha_{p2plane} \sum_{(v_i, c_i) \in \mathcal{J}} |\vec{n}(c_i)(v_i - c_i)|^2 \\ & + \alpha_{memb} \sum_{i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}(v_i)} w_{i,j} \|v_i - v_j\|_2^2, \end{aligned} \quad (2)$$

where,  $w_{i,j}$  is the weight corresponding to the biharmonic Laplacian operator (see [19, 4]),  $\vec{n}(c_i)$  is the normal of the corresponding vertex at the target mesh  $c_i$ ,  $\mathcal{J}$  is the set of the remaining associated vertex pairs  $(v_i, c_i)$ , and  $\mathcal{N}(v_i)$  is the set 1-ring neighboring vertices about the vertex  $v_i$ . Notice that the first term above is the sum of squared Euclidean distances between matches. The second term is the distance from the point  $v_i$  to the tangent plane at the corresponding point of the target mesh. The third term quantifies the stiffness of the mesh.

- If the motion of the template mesh between the current iteration and the previous one is below a fixed threshold, we divide the weight  $\alpha_{memb}$  by two. This relaxes the stiffness term and allows a greater deformation in the next iteration.

This iterative process terminates when the stiffness weight is below a given threshold. Further implementation information and parameters of the registration process are provided in the supplementary material. The resulting output of this phase is a deformed template with fixed triangulation, which contains the overall facial structure recovered by the network, yet, is smoother and complete, see the third column of Figure 9.

## 4.2. Fine Detail Reconstruction

Although the network already recovers fine geometric details, such as wrinkles and moles, across parts of the face, a geometric approach can reconstruct details at a finer level, on the entire face, independently of the resolution. Here, we propose an approach motivated by the passive-stereo facial reconstruction method suggested in [2]. The underlying assumption here is that subtle geometric structures can be explained by local variations in the image domain. For some skin tissues, such as nevi, this assumption is inaccurate as the intensity variation results from the albedo. In such cases, the geometric structure would be wrongly modified. Still, for most parts of the face, the reconstructed details are consistent with the actual variations in depth.

The method begins from an interpolated version of the deformed template. Each vertex  $v \in \mathcal{V}_D$  is painted with the intensity value of the nearest pixel in the image plane. Since we are interested in recovering small details, only the high spatial frequencies,  $\mu(v)$ , of the texture,  $\tau(v)$ , are taken into consideration in this phase. For computing this frequency band, we subtract the synthesized low frequencies from the original intensity values. This low-pass filtered part can be computed by convolving the texture with a spatially varying Gaussian kernel in the image domain, as originally proposed. In contrast, since this convolution is equivalent to computing the heat distribution upon the shape after time  $dt$ , where the initial heat profile is the original texture, we propose to compute  $\mu(v)$  as

$$\mu(v) = \tau(v) - (I - dt \cdot \Delta_g)^{-1} \tau(v), \quad (3)$$

where  $I$  is the identity matrix,  $\Delta_g$  is the cotangent weight discrete Laplacian operator for triangulated meshes [29], and  $dt$  is a scalar proportional to the cut-off frequency of the filter.

Next, we displace each vertex along its normal direction such that  $v' = v + \delta(v) \vec{n}(v)$ . The step size of the displacement,  $\delta(v)$ , is a combination of a data-driven term,  $\delta_\mu(v)$ , and a regularization one,  $\delta_s(v)$ . The data-driven term is

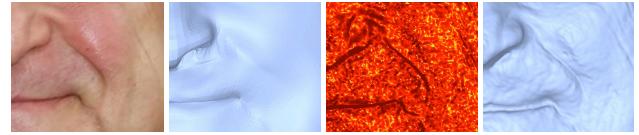


Figure 6: Mesoscopic displacement. From left to right: an input image, the shape after the iterative registration, the high-frequency part of the texture -  $\mu(v)$ , and the final shape.

guided by the high-pass filtered part of the texture,  $\mu(v)$ . In practice, we require the local differences in the geometry to be proportional to the local variation in the high frequency band of the texture. For each vertex  $v$ , with a normal  $\vec{n}(v)$ , and a neighboring vertex  $v_i$ , the data-driven term is given by

$$\delta_\mu(v) = \frac{\sum_{v_i \in \mathcal{N}(v)} \alpha_{(v,v_i)} (\mu(v) - \mu(v_i)) \left(1 - \frac{|\langle v - v_i, \vec{n}(v) \rangle|}{\|v - v_i\|}\right)}{\sum_{v_i \in \mathcal{N}(v)} \alpha_{(v,v_i)}}, \quad (4)$$

where  $\alpha_{(v,v_i)} = \exp(-\|v - v_i\|)$ . For further explanation of Equation 13, we refer the reader to the supplementary material of this paper or the implementation details of [2].

Since we move each vertex along the normal direction, triangles could intersect each other, particularly in domains of high curvature. To reduce the probability of such collisions, a regularizing displacement field,  $\delta_s(v)$ , is added. This term is proportional to the mean curvature of the original surface, and is equivalent to a single explicit mesh fairing step [9]. The final surface modification is given by

$$v' = v + (\eta \delta_\mu(v) + (1 - \eta) \delta_s(v)) \cdot \vec{n}(v), \quad (5)$$

for some constant  $\eta \in [0, 1]$ . A demonstration of the results before and after this step is presented in Figure 6

## 5. Experiments

Next, we present evaluations on both the proposed network and the pipeline as a whole, and comparison to different prominent methods of single image based facial reconstruction [24, 46, 32].

### 5.1. Qualitative Evaluation

The first component of our algorithm is an Image-to-Image network. In Figure 7, we show samples of output maps produced by the proposed network. Although the network was trained with synthetic data, with simple random backgrounds (see Figure 4), it successfully learns to separate the hair and background from the face itself and learn the corresponding representations. To qualitatively assess the accuracy of the correspondence, we present

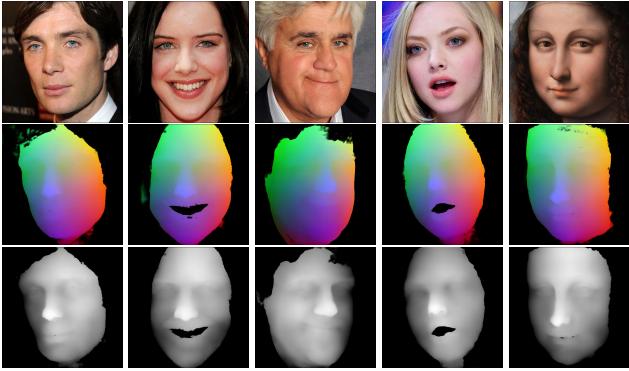


Figure 7: Network Output.

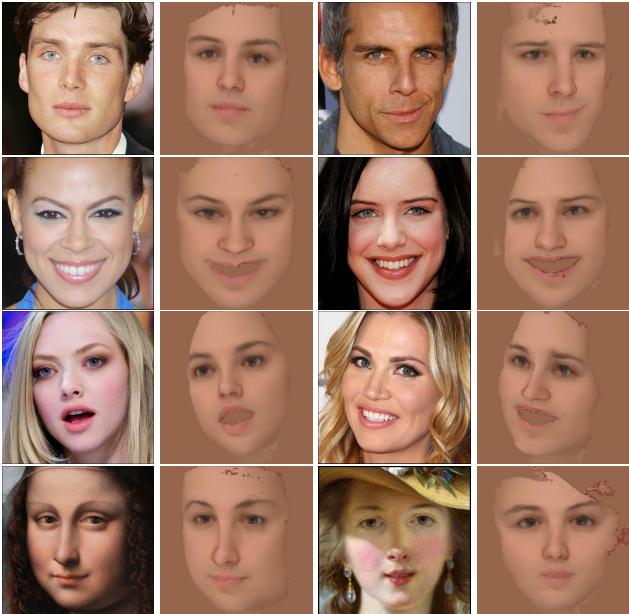


Figure 8: Texture mapping via the embedding.

a visualization where an average facial texture is mapped to the image plane via the predicted embedding, see Figure 8. Next, in Figure 9 we show the reconstruction of the network, alongside the registered template and the final shape. Note that the network successfully captures the overall facial structure including some subtle details. Figure 10 shows a qualitative comparison between the proposed method and others. Note how our method better matches the global structure, as well as the facial details. To better perceive these differences, see Figure 11. For additional results see Figure 1 and Figure 13.

## 5.2. Quantitative Evaluation

For a quantitative comparison, we used the first 200 subjects from the BU-3DFE dataset [42], which contains facial images aligned with ground truth depth images. Each

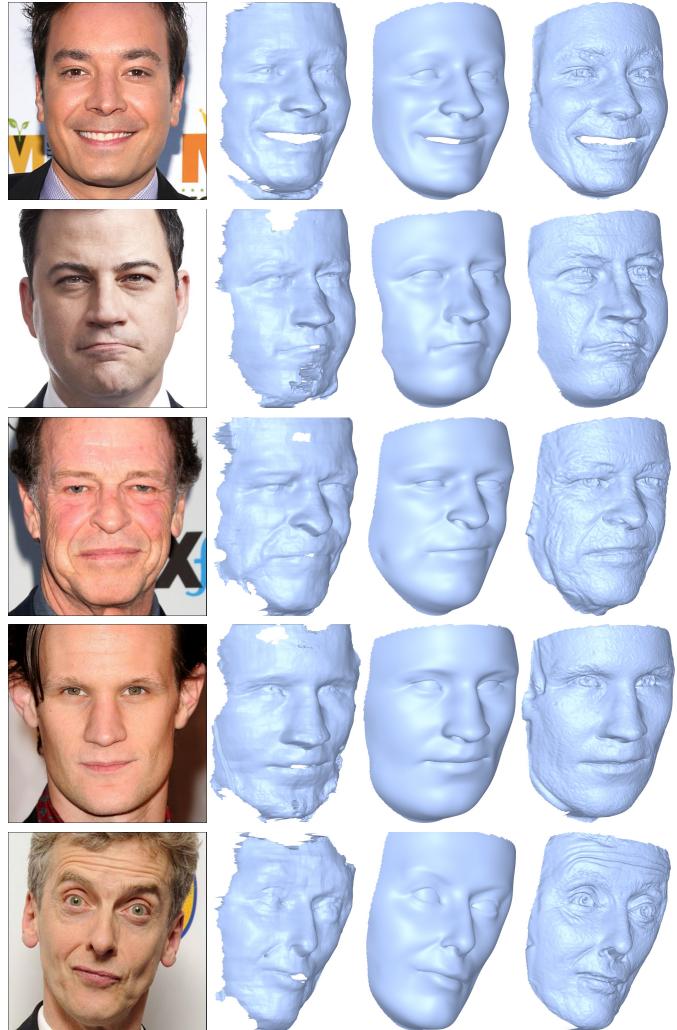


Figure 9: The reconstruction stages. From left to right: the input image, the reconstruction of the network, the registered template and the final shape.

method provides its own estimation for the depth image alongside a binary mask, representing the valid pixels to be taken into account in the evaluation. Obviously, since the problem of reconstructing depth from a single image is ill-posed, the estimation needs to be judged up to global scaling and transition along the depth axis. Thus, we compute these parameters using the Random Sample Consensus (RANSAC) approach [11], for normalizing the estimation according to the ground truth depth. This significantly reduces the absolute error of each method as the global parameter estimation is robust to outliers. Note that the parameters of the RANSAC were identical for all the methods and samples. The results of this comparison are given in Table 1, where the units are given in terms of the percentile of the ground-truth depth range. As a further analysis of the

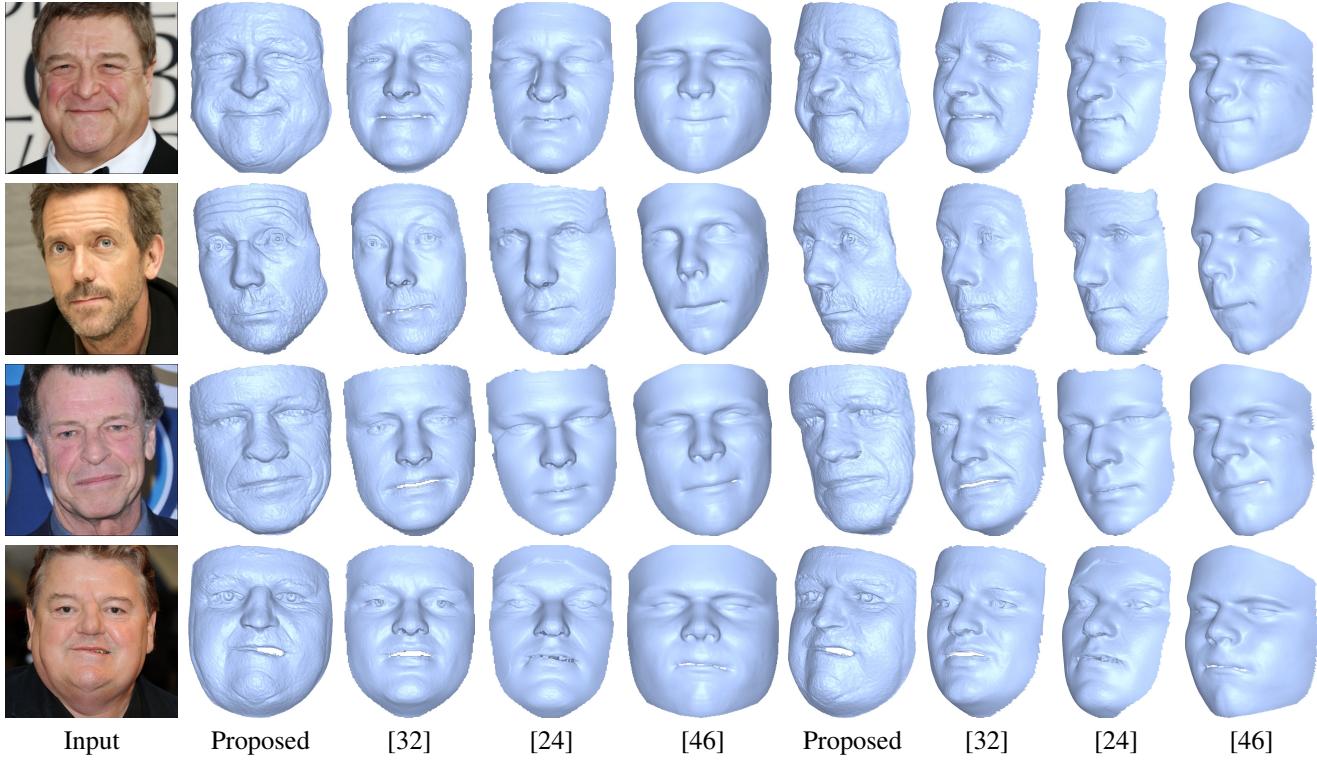


Figure 10: Qualitative comparison. Input images are presented alongside the reconstructions of the different methods.

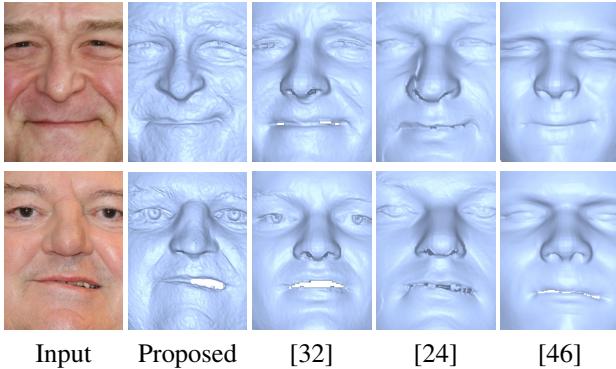


Figure 11: Zoomed qualitative result of first and fourth subjects from Figure 10.

reconstruction accuracy, we computed the mean absolute error of each method based on expressions, see Table 2.

### 5.3. The Network as a Geometric Constraint

As demonstrated by the results, the proposed network successfully learns both the depth and the embedding representations for a variety of images. This representation is the key part behind the reconstruction pipeline. However, it can also be helpful for other face-related tasks. As an example, we show that the network can be used as a ge-

|      | Mean Err.   | Std Err.    | Median Err. | 90% Err.    |
|------|-------------|-------------|-------------|-------------|
| [24] | 3.89        | 4.14        | 2.94        | 7.34        |
| [46] | 3.85        | 3.23        | 2.93        | 7.91        |
| [32] | 3.61        | 2.99        | 2.72        | 6.82        |
| Ours | <b>3.51</b> | <b>2.69</b> | <b>2.65</b> | <b>6.59</b> |

Table 1: Quantitative evaluation on the BU-3DFE Dataset. From left to right: the absolute depth errors evaluated by mean, standard deviation, median and the average ninety percent largest error.

|      | AN          | DI          | FE          | HA          | NE          | SA          | SU          |
|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| [24] | 3.47        | 4.03        | 3.94        | 4.30        | 3.43        | 3.52        | 4.19        |
| [46] | 4.00        | 3.93        | 3.91        | 3.70        | 3.76        | 3.61        | <b>3.96</b> |
| [32] | <b>3.42</b> | 3.46        | 3.64        | 3.41        | 4.22        | 3.59        | 4.00        |
| Ours | 3.67        | <b>3.34</b> | <b>3.36</b> | <b>3.01</b> | <b>3.17</b> | <b>3.37</b> | 4.41        |

Table 2: The mean error by expression. From left to right: Anger, Disgust, Fear, Happy, Neutral, Sad, Surprise.

ometric constraint for facial image manipulations, such as transforming synthetic images into realistic ones. This idea is based on recent advances in applying Generative Adversarial Networks (GAN) [13] for domain adaption tasks [38].

In the basic GAN framework, a Generator Network (G)

learns to map from the source domain,  $\mathcal{D}_S$ , to the target domain  $\mathcal{D}_T$ , where a Discriminator Network (D) tries to distinguish between generated images and samples from the target domain, by optimizing the following objective

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{y \sim \mathcal{D}_T} [\log D(y)] + \mathbb{E}_{x \sim \mathcal{D}_S} [\log (1 - D(G(x)))] . \quad (6)$$

Theoretically, this framework could also translate images from the synthetic domain into the realistic one. However, it does not guarantee that the underlying geometry of the synthetic data is persevered throughout that transformation. That is, the generated image might look realistic, but have a completely different facial structure from the synthetic input. To solve that potential inconsistency, we suggest to involve the proposed network as an additional loss function on the output of the generator.

$$L_{Geom}(x) = \|Net(x) - Net(G(x))\|_1 , \quad (7)$$

where  $Net(\cdot)$  represents the operation of the introduced network. Note that this is feasible, thanks to the fact that the proposed network is fully differentiable. The additional geometric fidelity term forces the generator to learn a mapping that makes a synthetic image more realistic while keeping the underlying geometry intact. This translation process could potentially be useful for data generation procedures, similarly to [35]. Some results of this process are visualized in Figure 12. Notice that the network implicitly learns to add facial hair and teeth, and modify the texture and shading, without changing the facial structure. As demonstrated by this analysis, the proposed network learns a strong representation that has merit not only for reconstruction, but for other tasks as well.

## 6. Discussion

One of the core ideas of this work was a model-free approach, where the solution space is not restricted to a low dimensional subspace. Instead, the Image-to-Image network represents the solution in the extremely high-dimensional image domain. However, lifting the subspace constraint makes the reconstruction problem more challenging, requiring the network to learn the representation space by itself, without any prior structure. In our framework, this structure is learned from synthetic examples, and shown to generalize to “in-the-wild” images. Still, the performance of such a framework is governed by the quality of the synthetic data. For example, the network has a tendency to classify thick beards as part of the background, as facial hair did not appear in the training data. In order to achieve higher performance, one could generate more realistic facial images with depth and correspondence maps.

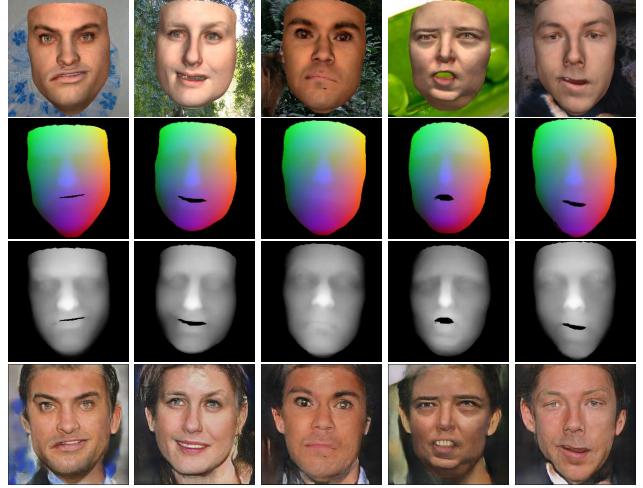


Figure 12: Translation results. From top to bottom: synthetic input images, the correspondence and the depth maps recovered by the network, and the transformed result. Notice how in the rightmost example the white fur at the background is transformed into a collar.

## 7. Conclusion

We presented an unrestricted approach for recovering the geometric structure of a face from a single image. Our algorithm employs an Image-to-Image network which maps the input image to a pixel-based geometric representation followed by an iterative deformation-based registration process. The network is trained only by synthetic facial images, yet, is capable of reconstructing real faces. Using the network as a loss function, we propose a framework for translating synthetic facial images into realistic ones while preserving the geometric structure.



Figure 13: Additional reconstruction results.

## References

- [1] O. Aldrian and W. A. Smith. A linear approach of 3D face shape and texture recovery using a 3d morphable model. In *Proceedings of the British Machine Vision Conference*, pages, pages 75–1, 2010.
- [2] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH ’10, pages 40:1–40:9, New York, NY, USA, 2010. ACM.
- [3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.
- [4] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, Jan 2008.
- [5] P. Breuer, K.-I. Kim, W. Kienzle, B. Scholkopf, and V. Blanz. Automatic 3D face reconstruction from single images or video. In *Automatic Face & Gesture Recognition, 2008. FG’08. 8th IEEE International Conference on*, pages 1–8. IEEE, 2008.
- [6] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (TOG)*, 34(4):46, 2015.
- [7] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):41, 2013.
- [8] W. Chen, H. Wang, Y. Li, H. Su, D. Lischinsk, D. Cohen-Or, B. Chen, et al. Synthesizing training images for boosting human 3D pose estimation. *arXiv preprint arXiv:1604.02703*, 2016.
- [9] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’99, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [10] P. Dou, Y. Wu, S. K. Shah, and I. A. Kakadiaris. Robust 3D face shape reconstruction from single images via two-fold coupled structure learning. In *Proc. British Machine Vision Conference*, pages 1–13, 2014.
- [11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [12] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3D face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):28, 2016.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [14] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. *arXiv preprint arXiv:1612.01202*, 2016.
- [15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015.
- [16] T. Hassner. Viewing real-world faces in 3d. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3607–3614, 2013.
- [17] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015.
- [18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [19] B. T. Helenbrook. Mesh deformation using the biharmonic operator. *International journal for numerical methods in engineering*, 56(7):1007–1021, 2003.
- [20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.
- [21] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu. 3d face reconstruction with geometry details from a single image. *arXiv preprint arXiv:1702.05619*, 2017.
- [22] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3D model fitting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [23] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.
- [24] I. Kemelmacher-Shlizerman and R. Basri. 3D face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):394–405, 2011.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [26] H. Li. *Animation Reconstruction of Deformable Surfaces*. PhD thesis, ETH Zurich, November 2010.
- [27] F. Liu, D. Zeng, J. Li, and Q. Zhao. Cascaded regressor based 3D face reconstruction from a single arbitrary view image. *arXiv preprint arXiv:1509.06161*, 2015.
- [28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [29] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, et al. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3(2):52–58, 2002.
- [30] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014.

- [31] E. Richardson, M. Sela, and R. Kimmel. 3D face reconstruction by learning from synthetic data. In *3D Vision (3DV), 2016 International Conference on*, pages 460–469. IEEE, 2016.
- [32] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. *arXiv preprint arXiv:1611.05053*, 2016.
- [33] O. Ronneberger, P. Fischer, and T. 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.
- [34] S. Saito, L. Wei, L. Hu, K. Nagano, and H. Li. Photorealistic facial texture inference using deep neural networks. *arXiv preprint arXiv:1612.00523*, 2016.
- [35] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016.
- [36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [38] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.
- [39] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.
- [40] A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *arXiv preprint arXiv:1612.04904*, 2016.
- [41] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM Transactions on Graphics (TOG)*, volume 30, page 77. ACM, 2011.
- [42] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*, pages 211–216. IEEE, 2006.
- [43] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.
- [44] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 386–391, 2013.
- [45] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 146–155, 2016.
- [46] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796, 2015.
- [47] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.

## Supplementary Material

### A. Additional Network Details

Here, we summarize additional considerations concerning the network and its training procedure.

- The proposed architecture is based on the one introduced in [20]. For allowing further refinement of the results, three additional convolution layers with a kernel of size  $1 \times 1$  were concatenated at the end. Following the notations of [20], the encoder architecture is given as

$$C64 - C128 - C256 - C512 - C512 - C512 - C512 - C512,$$

while the decoder is given by

$$CD512 - CD512 - CD512 - C512 - C512 - C256 - C128 - C64 - C^*64 - C^*32 - C^*4,$$

where  $C^*$  represents a  $1 \times 1$  convolution with stride 1.

- The resolution of the input and output training images was  $512 \times 512$  pixels. While this is a relatively large input size for training, the Image-to-Image architecture was able to process it successfully, and provided accurate results. Although, one could train a network on smaller resolutions and then evaluate it on larger images, as shown in [20], we found that our network did not successfully scale up for unseen resolutions.
- The weights during training were set to  $\lambda_{L1} = 100$  and  $\lambda_N = 10$ , where  $\lambda_{L1}$  is the standard  $L_1$  loss, and  $\lambda_N$  is the proposed normals loss.
- While a single network was successfully trained to retrieve both depth and correspondence representations, our experiments show that training separated networks to recover the representations is preferable. Note that the architectures of both networks were identical. This can be justified by the observation that during training, a network allocates its resources for a specific translation task and the representation maps we used have different characteristics.
- A necessary parameter for the registration step is the scale of the face with respect to the image dimensions. While this can be estimated based on global features, such as the distance between the eyes, we opted to retrieve it directly by training the network to predict the  $x$  and  $y$  coordinates of each pixel in the image alongside the  $z$  coordinate.

### B. Additional Registration and Refinement Details

Next, we provide a detailed version of the iterative deformation-based registration phase, including implementation details of the fine detail reconstruction.

#### B.1. Non-Rigid Registration

First, we turn the  $x, y$  and  $z$  maps from the network into a mesh, by connecting four neighboring pixels, for which the coordinates are known, with a couple of triangles. This step yields a target mesh that might have holes but has dense map to our template model. Based on the correspondence given by the network, we compute the affine transformation from a template face to the mesh. This operation is done by minimizing the squared Euclidean distances between corresponding vertex pairs. To handle outliers, a RANSAC approach is used [11] with 1,000 iterations and a threshold of 3 millimeters for detecting inliers. Next, similar to [26], an iterative non-rigid registration process deforms the transformed template, aligning it with the mesh. Note, that throughout the registration, only the template is warped, while the target mesh remains fixed. Each iteration involves the following four steps.

1. Each vertex in the template mesh,  $v_i \in \mathcal{V}$ , is associated with a vertex,  $c_i$ , on the target mesh, by evaluating the nearest neighbor in the embedding space. This step is different from the method described in [26], which computes the nearest neighbor in the Euclidean space. As a result, the proposed step allows registering a single template face to different facial identities with arbitrary expressions.
2. Pairs,  $(v_i, c_i)$ , which are physically distant by more than 1 millimeter and those with normal direction disagreement of more than 5 degrees are detected and ignored in the next step.
3. The template mesh is deformed by minimizing the following energy

$$\begin{aligned}
E(V, C) = & \alpha_{p2point} \sum_{(v_i, c_i) \in \mathcal{J}} \|v_i - c_i\|_2^2 \\
& + \alpha_{p2plane} \sum_{(v_i, c_i) \in \mathcal{J}} |\vec{n}(c_i)(v_i - c_i)|^2 \\
& + \alpha_{memb} \sum_{i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}(v_i)} w_{i,j} \|v_i - v_j\|_2^2,
\end{aligned} \tag{8}$$

where,  $w_{i,j}$  is the weight corresponding to the biharmonic Laplacian operator (see [19, 4]),  $\vec{n}(c_i)$  is the normal of the corresponding vertex at the target mesh  $c_i$ ,  $\mathcal{J}$  is the set of the remaining associated vertex pairs  $(v_i, c_i)$ , and  $\mathcal{N}(v_i)$  is the set 1-ring neighboring vertices about the vertex  $v_i$ . Notice that the first term above is the sum of squared Euclidean distances between matches and its weight  $\alpha_{p2point}$  is set to 0.1. The second term is the distance from the point  $v_i$  to the tangent plane at the corresponding point on the target mesh, and its weight  $\alpha_{p2plane}$  is set to 1. The third term quantifies the stiffness of the mesh and its weight  $\alpha_{memb}$  is initialized to  $10^8$ . In practice, the energy term given in Equation 8 is minimized iteratively by an inner loop which contains a linear system of equations. We run this loop until the norm of the difference between the vertex positions of the current iteration and the previous one is below 0.01.

4. If the motion of the template mesh between the current outer iteration and the previous one is below 0.1, we divide the weight  $\alpha_{memb}$  by two. This relaxes the stiffness term and allows a greater deformation in the next outer iteration. In addition, we evaluate the difference between the number of remaining pairwise matches in the current iteration versus the previous one. If the difference is below 500, we modify the vertex association step to estimate the physical nearest neighbor vertex, instead of the nearest neighbor in the space of the embedding given by the network.

This iterative process terminates when the stiffness weight  $\alpha_{memb}$  is below  $10^6$ . The resulting output of this phase is a deformed template with fixed triangulation, which contains the overall facial structure recovered by the network, yet, is smoother and complete.

## B.2. Fine Detail Reconstruction

Although the network already recovers fine geometric details, such as wrinkles and moles, across parts of the face, a geometric approach can reconstruct details at a finer level, on the entire face, independently of the resolution. Here, we propose an approach motivated by the passive-stereo facial reconstruction method suggested in [2]. The underlying assumption here is that subtle geometric structures can be explained by local variations in the image domain. For some skin tissues, such as nevi, this assumption is inaccurate as the intensity variation results from the albedo. In such cases, the geometric structure would be wrongly modified. Still, for most parts of the face, the reconstructed details are consistent with the actual variations in depth.

The method begins from an interpolated version of the deformed template, provided by a surface subdivision technique. Each vertex  $v \in \mathcal{V}_D$  is painted with the intensity value of the nearest pixel in the image plane. Since we are interested in recovering small details, only the high spatial frequencies,  $\mu(v)$ , of the texture,  $\tau(v)$ , are taken into consideration in this phase. For computing this frequency band, we subtract the synthesized low frequencies from the original intensity values. This low-pass filtered part can be computed by convolving the texture with a spatially varying Gaussian kernel in the image domain, as originally proposed. In contrast, since this convolution is equivalent to computing the heat distribution upon the shape after time  $dt$ , where the initial heat profile is the original texture, we propose to compute  $\mu(v)$  as

$$\mu(v) = \tau(v) - (I - dt \cdot \Delta_g)^{-1} \tau(v), \tag{9}$$

where  $I$  is the identity matrix,  $\Delta_g$  is the cotangent weight discrete Laplacian operator for triangulated meshes [29], and  $dt = 0.2$  is a scalar proportional to the cut-off frequency of the filter.

Next, we displace each vertex along its normal direction such that  $v' = v + \delta(v)\vec{n}(v)$ . The step size of the displacement,  $\delta(v)$ , is a combination of a data-driven term,  $\delta_\mu(v)$ , and a regularization one,  $\delta_s(v)$ . The data-driven term is guided by the high-pass filtered part of the texture,  $\mu(v)$ . In practice, we require the local differences in the geometry to be proportional to the local variation in the high frequency band of the texture. That is for each vertex  $v$ , with a normal  $\vec{n}(v)$ , and a neighboring vertex  $v_i$ , the data-driven term is given by

$$(\mu(v) - \mu(v_i)) = \langle v + \delta_\mu(v)\vec{n}(v) - v_i, \vec{n}(v) \rangle. \quad (10)$$

Thus, the step size assuming a single neighboring vertex can be calculated by

$$\delta_\mu(v) = \gamma(\mu(v) - \mu(v_i)) - \langle v - v_i, \vec{n}(v) \rangle. \quad (11)$$

In the presence of any number of neighboring vertices of  $v$ , we compute the weighted average of its 1-ring neighborhood

$$\delta_\mu(v) = \frac{\sum_{v_i \in \mathcal{N}(v)} \alpha(v, v_i) \gamma [(\mu(v) - \mu(v_i)) - \langle v - v_i, \vec{n}(v) \rangle]}{\sum_{v_i \in \mathcal{N}(v)} \alpha(v, v_i)}, \quad (12)$$

An alternative term can spatially attenuate the contribution of the data-driven term in curved regions for regularizing the reconstruction by

$$\delta_\mu(v) = \frac{\sum_{v_i \in \mathcal{N}(v)} \alpha_{(v, v_i)} (\mu(v) - \mu(v_i)) \left(1 - \frac{|\langle v - v_i, \vec{n}(v) \rangle|}{\|v - v_i\|}\right)}{\sum_{v_i \in \mathcal{N}(v)} \alpha_{(v, v_i)}}, \quad (13)$$

where  $\alpha_{(v, v_i)} = \exp(-\|v - v_i\|)$ . where  $\mathcal{N}(v)$  is the set 1-ring neighboring vertices about the vertex  $v$ , and  $\vec{n}(v)$  is the unit normal at the vertex  $v$ .

Since we move each vertex along the normal direction, triangles could intersect each other, particularly in regions with high curvature. To reduce the probability of such collisions, a regularizing displacement field,  $\delta_s(v)$ , is added. This term is proportional to the mean curvature of the original surface, and is equivalent to a single explicit mesh fairing step [9]. The final surface modification is given by

$$v' = v + (\eta \delta_\mu(v) + (1 - \eta) \delta_s(v)) \cdot \vec{n}(v), \quad (14)$$

for a constant  $\eta = 0.2$ .

### C. Additional Experimental Results

We present additional qualitative results of our method. Figure 14 shows the output representations of the proposed network for a variety of different faces. Notice the failure cases presented in the last couple of rows. One can see that the network generalizes well, but is still limited by the synthetic data. Specifically, the network might fail in presence of occlusions, facial hair or extreme poses. Additional reconstruction results of our method are presented in Figure 15 and Figure 16. For analyzing the distribution of the error along the face, we present an additional comparison in Figure 17, where the absolute error, given in percents of the ground truth depth, is shown for several facial images.

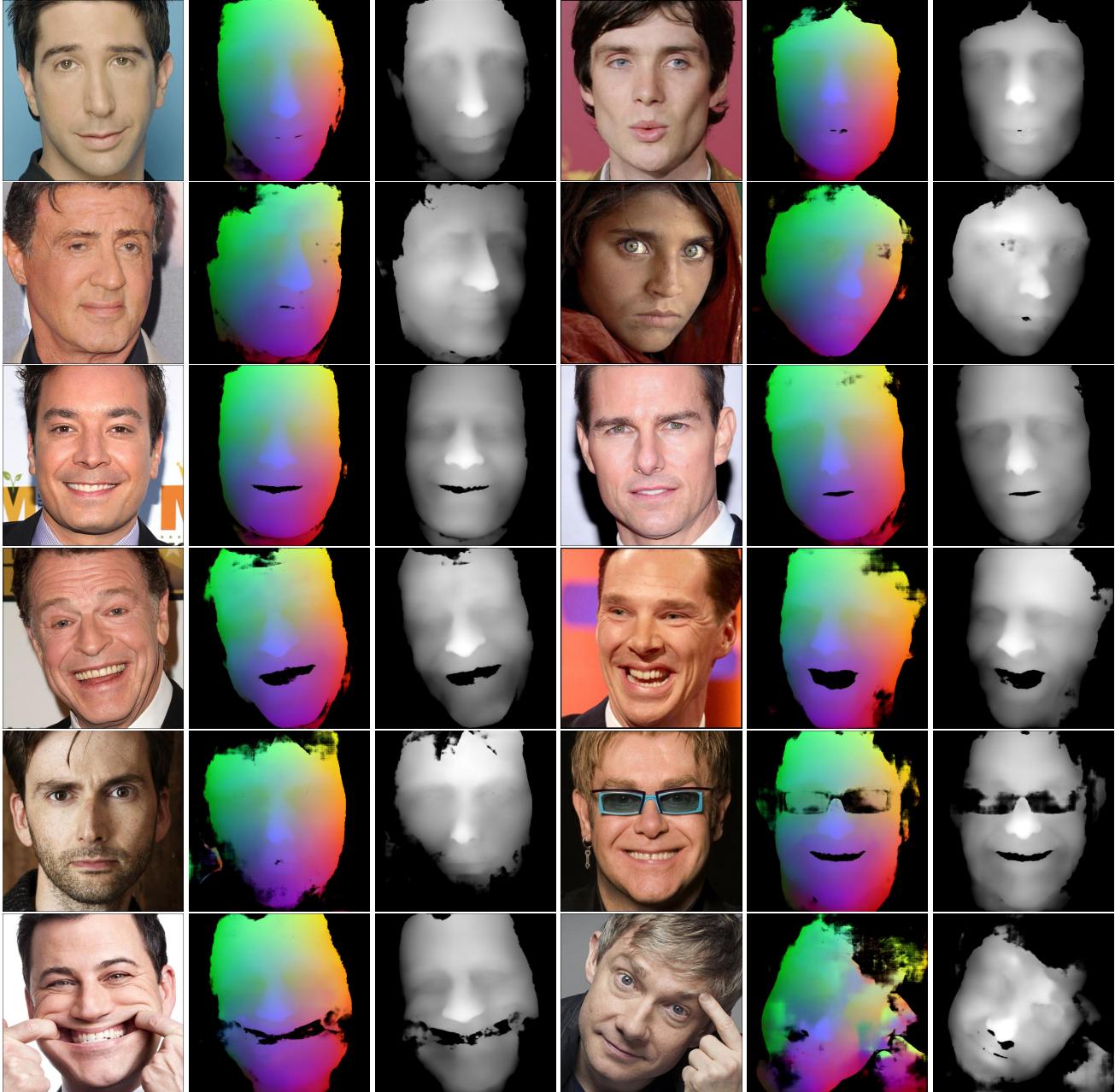


Figure 14: Network Output.



Figure 15: Additional reconstruction results.

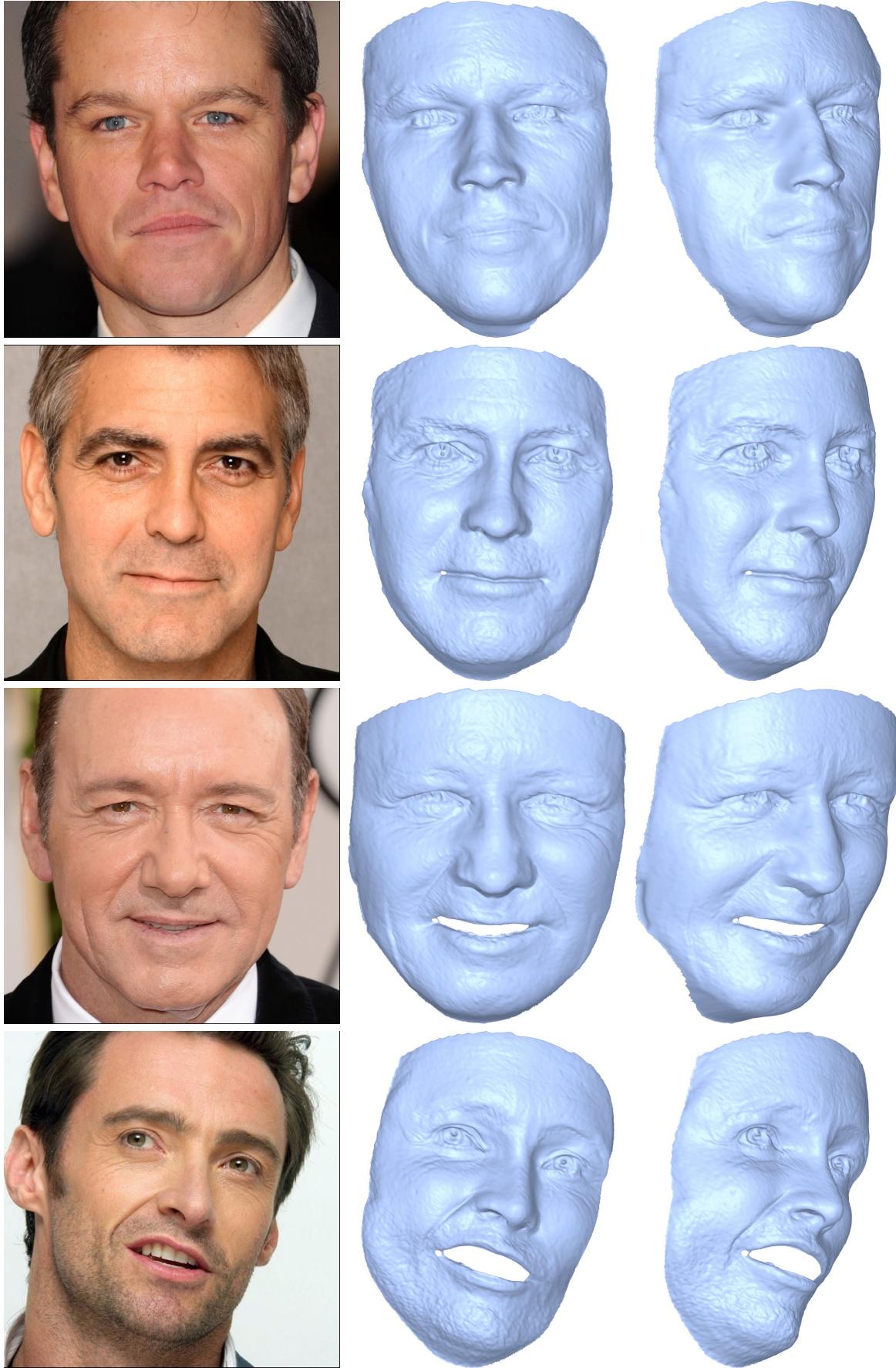


Figure 16: Additional reconstruction results.

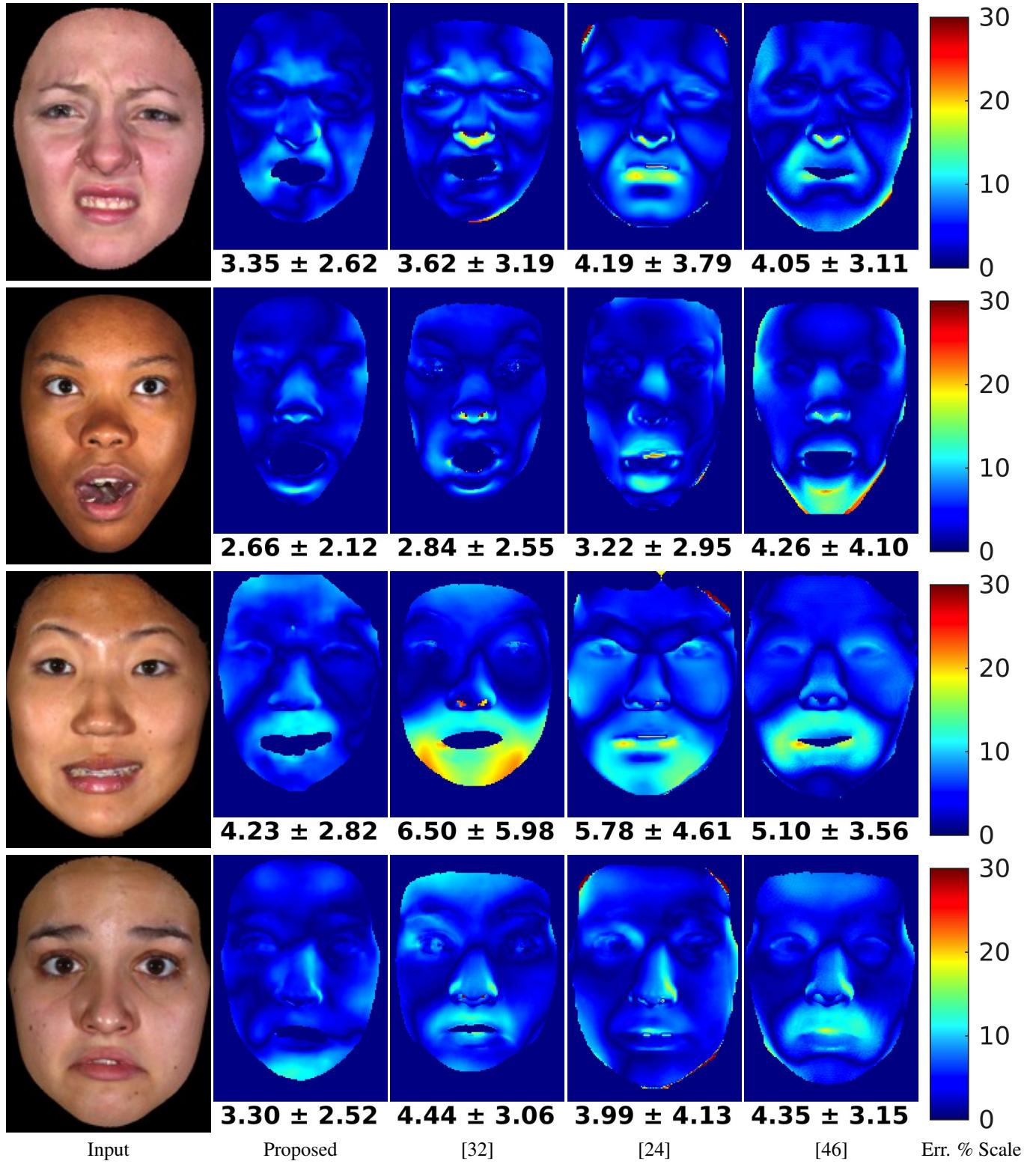


Figure 17: Error heat maps in percentile of ground truth depth.