# FACEDET3D: FACIAL EXPRESSIONS WITH 3D GEOMETRIC DETAIL HALLUCINATION

## -SUPPLEMENTARY-

**Anonymous authors**
Paper under double-blind review

## CONTENTS



## 1 EXPRESSION ANIMATION RESULTS

We show results of animating facial expressions with our method using AUs and expressions parameters extracted from videos from the MUG Aifanti et al. (2010) dataset. Videos titled `{ID}_{EXPR}.avi` contains the video of subject with ID `ID` animated in the input view with expression `EXPR`. In the first row, the first column is the input image, the second is the output rendering with facial details, the third is the output rendering without facial details, the fourth is the shaded geometry with details and the fifth is the shaded geometry without details. In the second row, a subset of hallucinated details, both on the detailed shaded geometry and the rendering, are marked with green rectangles and zoomed-in. Beside the zoomed-in details are plotted the zoom-ins of the same regions of the face without the details (marked with red rectangles) of both the non-detailed shading and the non-detailed rendering.

The animated mesh with the hallucinated details is smoothed via Savitsky-Golay smoothing applied to the vertices. As can be seen in the videos, the appearance of the details (both on the geometry and the final render) is consistent with the target expression and its intensity. The details are most prominent at the peak of the expression and disappear when the expression activation is minimal.

## 2 DETAILS HALLUCINATION TRAINING LOSSES

In this section we give a full exposition of the losses used to train the details hallucination network. A plausible detail map, $\widetilde{\mathcal{D}}(\mathbf{I_y})$, that's consistent with the given target expression $\mathbf{y}$ is hallucinated as follows:

$$\widetilde{\mathcal{D}}(\mathbf{I_y}) = \mathcal{DetH}(\mathcal{D}(\mathbf{I_x}), \mathbf{x}, \mathbf{y}, \text{Age}(\mathbf{I_x}), \text{FaceID}(\mathbf{I_x})) \ , \tag{1}$$

where $\mathbf{x}$ is the input AU, $\text{Age}(\mathbf{I_x})$ are features extracted from an age prediction network and $\text{FaceID}(\mathbf{I_x})$ is the facial embedding of $\mathbf{I_x}$ extracted using Schroff et al. (2015).

**Expression Adversarial Loss.** In order to ensure the hallucinated facial geometric details, $\widetilde{\mathcal{D}}(\mathbf{I_y})$, are consistent with the target expression $\mathbf{y}$, as encoded by AUs, we use an expression discriminator $\text{D}_{\text{Exp}}$. Given $\mathcal{D}(\mathbf{I_x})$ of some image $\mathbf{I_x}$ manifesting expression $\mathbf{x}$, $\text{D}_{\text{Exp}}$, outputs the following

$$\text{D}_{\text{Exp}}(\mathcal{D}(\mathbf{I}_x)) = \{r, \hat{\mathbf{x}}\} \ , \tag{2}$$

where $r$ is a realism score and $\hat{\mathbf{x}}$ is the predicted AU. For brevity, we will use $\text{D}_{\text{Exp}}(\mathcal{D}(\mathbf{I_x}))$ and $\text{D}_{\text{Exp}}(\mathcal{D})$ interchangeably. We use the Non-Saturating adversarial loss Goodfellow et al. (2014) along with the R1 gradient penalty Mescheder et al. (2018) to train $\text{D}_{\text{Exp}}$. We use a UNet based discriminator Schonfeld et al. (2020) in order to discriminate on pixel level. In addition, $\text{D}_{\text{Exp}}$ is trained to minimize the error of the predicted AU

$$\mathcal{L}_{\text{AU}}^{\text{D}_{\text{Exp}}} = \mathbb{E}_{\mathcal{D}\sim\mathcal{P}_{\mathcal{D}}} \left[ ||[\text{D}_{\text{Exp}}^{\text{AU}}(\mathcal{D}) - \mathbf{x}||_2^2 \right] \ , \tag{3}$$

where $\mathrm{D}_{\mathrm{Exp}}^{\mathrm{AU}}$ is the AU output head of $\mathrm{D}_{\mathrm{Exp}}$. The Details Hallucination Network, $\mathcal{DetH}$, in addition to be trained to minimize adversarial loss, is also trained to minimize the expression loss:

$$\mathcal{L}_{\mathrm{AU}}^{\mathcal{DetH}} = \mathop{\mathbb{E}}_{\mathbf{I_x},\{\mathbf{y}\}} ||\mathrm{D}_{\mathrm{Exp}}^{\mathrm{AU}}(\mathcal{DetH}(\cdot)) - \mathbf{y}||_2^2 \tag{4}$$

where $\mathcal{DetH}(\cdot)$ is to be read as in Eq. (1) and $\mathbf{y}$ is the target AU.

**Cycle Consistency Loss.** In order to ensure the input detail map is changed as little as possible, we enforce a cycle consistency loss on $\mathcal{DetH}$, as follows:

$$\begin{aligned} \widetilde{\mathcal{D}}(\mathbf{I_x}) &= \mathcal{DetH}(\widetilde{\mathcal{D}}(\mathbf{I_y}), \mathbf{y}, \mathbf{x}, \mathrm{Age}(\mathbf{I_x}), \mathrm{FaceID}(\mathbf{I_x})) \\ \mathcal{L}_{\mathrm{Cyc}}^{\mathcal{DetH}} &= \mathrm{LapLoss}\left(\widetilde{\mathcal{D}}(\mathbf{I_x}), \mathcal{D}(\mathbf{I_x})\right) \end{aligned} \tag{5}$$

where LapLoss is the Laplacian Loss Ling & Okada (2006); Bojanowski et al. (2018).

**Regression Loss.** In order to speed up training, we use a small amount of video data from MUG Aifanti et al. (2010) and ADFES Van Der Schalk et al. (2011) to directly regress the details map estimated by FDS Chen et al. (2019) as

$$\begin{aligned} \widetilde{\mathcal{D}}(\mathbf{I}_{\mathbf{y}}^k) &= \mathcal{DetH}\left(\mathcal{D}(\mathbf{I}_{\mathbf{x}}^m), \mathbf{x}, \mathbf{y}, \mathrm{Age}(\mathbf{I}_{\mathbf{x}}^m), \mathrm{FaceID}(\mathbf{I}_{\mathbf{x}}^m)\right) \\ \mathcal{L}_{\mathrm{Regress}}^{\mathcal{DetH}} &= \mathrm{LapLoss}\left(\widetilde{\mathcal{D}}(\mathbf{I}_{\mathbf{y}}^k), \mathcal{D}(\mathbf{I}_{\mathbf{y}}^k)\right) \end{aligned} \tag{6}$$

where, $\mathcal{D}(\mathbf{I}_{\mathbf{y}}^k)$ and $\mathcal{D}(\mathbf{I}_{\mathbf{x}}^m)$ are the detail map of $k$-th frame $\mathbf{I}_{\mathbf{y}}^k$ and $m$-th frame $\mathbf{I}_{\mathbf{x}}^m$ respectively. Training solely on video data is not possible due to the significant bias the dataset has towards younger subjects.

**Superresolution Losses.** The detail maps generated by FDS Chen et al. (2019) are of resolution $4096 \times 4096$ and thus cannot be used directly for training due to GPU-memory constraints. To get around this, we train $\mathcal{DetH}$ on detail maps downsampled to $256 \times 256$. Simultaneously, we finetune a superresolution network, RCAN Zhang et al. (2018), to super-resolve downsampled $256 \times 256$ patches of $\mathcal{D}(\mathbf{I_x})$ by a factor of 4

$$\mathcal{L}_{\mathrm{SR}}^{\mathrm{RCAN}} = \mathrm{L1}\left(\mathrm{RCAN}\left(\mathcal{D}(\mathbf{I_x})_{256}^P\right), \mathcal{D}(\mathbf{I_x})_{1024}^P\right) \ , \tag{7}$$

where $\mathcal{D}(\mathbf{I_x})_{1024}^P$ is a randomly sampled patch of resolution $1024 \times 1024$ from the full-resolution detail map $\mathcal{D}(\mathbf{I_x})$ and $\mathcal{D}(\mathbf{I_x})_{256}^P$ is its downsampled version. During inference, we use RCAN twice on the hallucinated detail map $\widetilde{\mathcal{D}}(\mathbf{I_y})$ to upsample it to $4096 \times 4096$:

$$\widetilde{\mathcal{D}}(\mathbf{I_y})^{HR} = \mathrm{RCAN}\left(\mathrm{RCAN}\left(\widetilde{\mathcal{D}}(\mathbf{I_y})\right)\right) \ . \tag{8}$$

In the interest of brevity, we will use $\widetilde{\mathcal{D}}(\mathbf{I_y})$ in lieu of $\widetilde{\mathcal{D}}(\mathbf{I_y})^{HR}$ in the remainder of this text.

## 3 Rendering Network Training Losses

We now give a full exposition of the training losses used to train the rendering network $\mathcal{R}(\cdot)$. The detailed face geometry is rendered as follows:

$$\tilde{\mathbf{I}}_{\mathbf{y}} = \mathcal{R}\left(\mathcal{T}(\mathbf{I_x}), \widetilde{\mathcal{D}}(\mathbf{I_y}), \boldsymbol{\alpha}_{\mathbf{s}}, \hat{\boldsymbol{\alpha}}_{\mathbf{e}}, \mathbf{y}, \mathbf{c}, l, \boldsymbol{\gamma}\right) \ , \tag{9}$$

where $\mathcal{T}(\mathbf{I_x})$ is the texture map extracted using FDS Chen et al. (2019), $\hat{\boldsymbol{\alpha}}_{\mathbf{e}}$ are the target expression parameters, $\boldsymbol{\alpha}_{\mathbf{s}}$ are the shape parameters, $\mathbf{y}$ is the target AU, $\mathbf{c}$ are the desired camera parameters, $\boldsymbol{\gamma}$ is the albedo PCA-space parameters of BFM Gerig et al. (2018), and $l$ are the lighting parameters. The rendering network is trained with the following losses:

**Photometric Loss.** The Photometric Loss ensures the rendered images are realistic by re-rendering a given image $\mathbf{I_x}$, producing $\hat{\mathbf{I}}_{\mathbf{x}}$ and comparing it to the ground truth. More specifically, given the texture map $\mathcal{T}(\mathbf{I_x})$, detailed geometry $G_{\mathcal{D}} = \{\mathcal{D}(\mathbf{I_x}), \boldsymbol{\alpha}_{\mathbf{s}}, \boldsymbol{\alpha}_{\mathbf{e}}\}$ and action unit $\mathbf{x}$, $\mathbf{I_x}$ is re-rendered using $\mathcal{R}$ as follows

$$\hat{\mathbf{I}}_{\mathbf{x}} = \mathcal{R}(\mathcal{T}(\mathbf{I_x}), G_{\mathcal{D}}, \mathbf{x}, \mathbf{c}, l, \boldsymbol{\gamma}) \tag{10}$$

where $c$ and $l$ are the camera and lighting parameters of $\mathbf{I_x}$. The re-rendered image $\hat{\mathbf{I}}_\mathbf{x}$ is then compared to $\mathbf{I_x}$

$$\begin{aligned}\mathcal{L}_{\text{Photo}} &= \text{MSE}(\hat{\mathbf{I}}_\mathbf{x}, \mathbf{I_x}) + \text{L1}(\hat{\mathbf{I}}_\mathbf{x}, \mathbf{I_x}) \\ &+ \text{LapLoss}(\hat{\mathbf{I}}_\mathbf{x}, \mathbf{I_x}) + \text{PerceptualLoss}(\hat{\mathbf{I}_\mathbf{x}}, \mathbf{I_x})\end{aligned} \tag{11}$$

where LapLoss is the Laplacian Loss Ling & Okada (2006); Bojanowski et al. (2018) and PerceptualLoss is the perceptual loss Johnson et al. (2016). In order to ensure the low-res rendering captures the image textures that are invariant to facial details as much as possible, the photometric loss is also applied to the low-res output of $g_\theta$ i.e $\hat{\mathbf{I}^{LR}_\mathbf{y}}$

$$\begin{aligned}\mathcal{L}^{\text{LR}}_{\text{Photo}} &= \text{MSE}(\hat{\mathbf{I}}^{LR}_\mathbf{x}, \mathbf{I_x}) + \text{L1}(\hat{\mathbf{I}}^{LR}_\mathbf{x}, \mathbf{I_x}) \\ &+ \text{LapLoss}(\hat{\mathbf{I}}^{LR}_\mathbf{x}, \mathbf{I_x}) + \text{PerceptualLoss}(\hat{\mathbf{I}}^{LR}_\mathbf{x}, \mathbf{I_x})\end{aligned} \tag{12}$$

**Augmented Wrinkle Loss (AugW).** In order to enforce the rendering of geometric details onto the rendered image we add 'fake' wrinkles to an image $\mathbf{I_x}$ and force $\mathcal{R}$ to generate the same. Given the the detailed geometry of $\mathbf{I_x}$, $G_\mathcal{D} = \{\mathcal{D}(\mathbf{I_x}), \boldsymbol{\alpha_s}, \boldsymbol{\alpha_e}\}$, a geometry with 'fake' details $G^*_\mathcal{D} = \{\mathcal{D}(\mathbf{I^*_z}), \boldsymbol{\alpha_s}, \boldsymbol{\alpha_e}\}$ using the geometric details from some random image $\mathbf{I^*_z}$ and the lighting $l$ of $\mathbf{I_x}$, the 'fake' wrinkles are added as follows:

$$\begin{aligned}&\texttt{Shading}(\mathbf{I}_\text{x}) = L_{\text{Sph}}(G_\mathcal{D}, l); \texttt{Shading}^*(\mathbf{I}_\text{x}) = L_{\text{Sph}}(G^*_\mathcal{D}, l) \\ &\mathbf{I}^*_\text{x} = \texttt{Shading}^*(\mathbf{I}_\text{x}) \times \left(\frac{\mathbf{I}_\text{x}}{\texttt{Shading}(\mathbf{I}_\text{x})}\right)\end{aligned} \tag{13}$$

where $L_{\text{Sph}}$ is the spherical harmonic lighting function and $l$ are the coefficients of the first 9 spherical harmonics. The artificially wrinkled image $\mathbf{I}^*_\text{x}$ is now re-rendered using $\mathcal{R}$

$$\begin{aligned}\hat{\mathbf{I}}^*_\mathbf{x} &= \mathcal{R}(\mathcal{T}(\mathbf{I_x}), G^*_\mathcal{D}, \mathbf{x}, \mathbf{c}, l) \\ \mathcal{L}_{\text{AugW}} &= \text{LapLoss}(\hat{\mathbf{I}}^*_\mathbf{x}, \mathbf{I}^*_\text{x}) \ .\end{aligned} \tag{14}$$

where, LapLoss is the Laplacian Loss Ling & Okada (2006). In order to faithfully reconstruct $\mathbf{I}^*_\text{x}$, $\mathcal{R}$ is forced to rely on the detailed geometry $G^*_\mathcal{D}$, since the input texture map $\mathcal{T}(\mathbf{I_x})$, and consequently the neural texture map, *contain no* information about the 'fake' wrinkles.

**Detailed Shading Loss (DSL).** In addition to the Augmented Wrinkle Loss, we also try to predict the shading of the detailed facial geometry from the output rendering $\hat{\mathbf{I}}_\mathbf{x}$

$$\begin{aligned}&\hat{\texttt{Shading}}(\hat{\mathbf{I}}_\mathbf{x}) = f_\theta(\hat{\mathbf{I}}_\mathbf{x}) \\ &\mathcal{L}_{\text{DSL}} = \text{LapLoss}(\hat{\texttt{Shading}}(\hat{\mathbf{I}}_\mathbf{x}), \texttt{Shading}^*(\mathbf{I}_\text{x})) \ ,\end{aligned} \tag{15}$$

where $f_\theta$ is a small convolutional network (CNN) with only two layers and the shading $\texttt{Shading}^*(\mathbf{I}_\text{x})$ is calculated as in Eq. (13). We calculate this loss only over the skin region. Since, $f_\theta$ is a small CNN with limited representational capacity, the details must be quite visible on the rendered image $\hat{\mathbf{I}}_\mathbf{x}$ in order for them to be picked up by $f_\theta$ to generate an accurate shading $\hat{\texttt{Shading}}(\hat{\mathbf{I}}_\mathbf{x})$.

**Expression Adversarial Loss.** In order to ensure that the rendered output conforms to the target expression we use an expression adversarial loss. Given a rendered image, $\hat{\mathbf{I}}_\mathbf{x} = \mathcal{R}(\mathcal{T}(\mathbf{I_x}), G_\mathcal{D}, \mathbf{x}, c, l)$, manifesting the expression encoded by AU $\mathbf{x}$ an expression discriminator, $\text{D}^{\text{RGB}}_{\text{Exp}}$, outputs

$$\text{D}^{\text{RGB}}_{\text{Exp}}(\hat{\mathbf{I}}_\mathbf{x}) = \{r, \hat{\mathbf{x}}\} \ , \tag{16}$$

where $r$ is a realism score and $\hat{\mathbf{x}}$ is the predicted AU. We use the Non-Saturating adversarial loss Goodfellow et al. (2014) along with the R1 gradient penalty Mescheder et al. (2018) to train $\text{D}_{\text{Exp}}$. In addition, $\text{D}^{\text{RGB}}_{\text{Exp}}$ is trained to minimize the predicted AU error

$$\mathcal{L}^{\text{D}^{\text{RGB}}_{\text{Exp}}}_{\text{AU}} = \mathbb{E}_{\mathbf{I_x} \sim \mathcal{P}_\mathbf{I}} \left[ ||[\text{D}^{\text{RGB,AU}}_{\text{Exp}}(\mathbf{I_x}) - \mathbf{x}||^2_2 \right] \ , \tag{17}$$

where $\text{D}^{\text{RGB,AU}}_{\text{Exp}}$ is the AU output head of $\text{D}^{\text{RGB}}_{\text{Exp}}$. The Rendering Network, $\mathcal{R}$, in addition to be trained to minimize adversarial loss, is also trained to minimize the AU loss

$$\mathcal{L}^{\mathcal{R}}_{\text{AU}} = \mathbb{E}_{\mathbf{I_x}} ||\text{D}^{\text{RGB,AU}}_{\text{Exp}}(\mathcal{R}(\cdot)) - \mathbf{y}||^2_2 \ , \tag{18}$$

where $\mathcal{R}(\cdot)$ is to be read as in Eq. (9).

## REFERENCES

Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. *WIAMIS*, pp. 1–4, 2010. 1, 2

Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. 2018. 2, 3

Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. Photo-realistic facial details synthesis from single image. In *ICCV*, 2019. 2

Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *IEEE FG*, 2018. 2

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 3

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

Haibin Ling and Kazunori Okada. Diffusion distance for histogram comparison. In *CVPR*, 2006. 2, 3

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? 2018. 1, 3

Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *CVPR*, 2020. 1

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1

Job Van Der Schalk, Skyler T Hawk, Agneta H Fischer, and Bertjan Doosje. Moving faces, looking places: validation of the amsterdam dynamic facial expression set (adfes). *Emotion*, 11(4):907, 2011. 2

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 2