

Supplementary Material for: 3D Face Morphable Models “In-the-Wild”

1. Gauss-Newton Jacobians

Herein, we derive the Jacobians used in Sec. 3.2 (Eqs. 18, 19) of the main paper. Let us first define some basic derivatives which are shared among all Jacobians.

Shape generation function. As explained in Sec. 2.1 (Eq. 2) of the main paper, the shape generation function is defined as

$$\begin{aligned} [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N]^T &\equiv \mathcal{S}(\mathbf{p}) = \\ &= \bar{\mathbf{s}} + \mathbf{U}_s \mathbf{p} \end{aligned} \quad (1)$$

Perspective camera function. As mentioned in Sec. 2.2 of the main paper, we define the perspective projection function as $\mathcal{P}(\mathbf{s}, \mathbf{c}) : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{2N}$ which, given a 3D shape instance (Eq. 1) it applies the following transformations

$$\begin{aligned} [x'_1, y'_1, x'_2, y'_2, \dots, x'_N, y'_N]^T &\equiv \mathcal{P}(\mathcal{S}(\mathbf{p}), \mathbf{c}) = \\ &= \left[f \frac{v_1^x}{v_1^z} + c_x, f \frac{v_1^y}{v_1^z} + c_y, f \frac{v_2^x}{v_2^z} + c_x, f \frac{v_2^y}{v_2^z} + c_y, \dots, f \frac{v_N^x}{v_N^z} + c_x, f \frac{v_N^y}{v_N^z} + c_y \right]^T \end{aligned} \quad (2)$$

with

$$\begin{bmatrix} v_1^x & v_2^x & \dots & v_N^x \\ v_1^y & v_2^y & \dots & v_N^y \\ v_1^z & v_2^z & \dots & v_N^z \end{bmatrix} = 2 \underbrace{\begin{bmatrix} \frac{1}{2} - q_2^2 - q_3^2 & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & \frac{1}{2} - q_1^2 - q_3^2 & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & \frac{1}{2} - q_1^2 - q_2^2 \end{bmatrix}}_{\mathbf{R}_v} \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ z_1 & z_2 & \dots & z_N \end{bmatrix} + \underbrace{\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}}_{\mathbf{t}_v} \quad (3)$$

Note that Eqs. 2 and 3 are redefining Eqs. 3 and 4 of the main paper for a mesh with N points.

Spatial derivative of feature-based image on projected image coordinates. Using the notation of Eq. 2, this derivative is expressed as

$$\left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathcal{P}(\mathcal{S}(\mathbf{p}), \mathbf{c})} \equiv \nabla \mathbf{F} = \left[\frac{\partial \mathbf{F}}{\partial x'_1}, \frac{\partial \mathbf{F}}{\partial y'_1}, \frac{\partial \mathbf{F}}{\partial x'_2}, \frac{\partial \mathbf{F}}{\partial y'_2}, \dots, \frac{\partial \mathbf{F}}{\partial x'_N}, \frac{\partial \mathbf{F}}{\partial y'_N} \right] \quad (4)$$

where each $\left[\frac{\partial \mathbf{F}}{\partial x'_i}, \frac{\partial \mathbf{F}}{\partial y'_i} \right], \forall i = 1, \dots, N$ has size $CN \times 2$. This derivative denotes the spatial gradient of the feature-based image \mathbf{F} with respect to the x and y coordinates of the projected image space and it has size $CN \times 2N$.

Spatial derivative of perspective projection function on 3D coordinates. By denoting the coordinates of a 3D shape (Eq. 1) as $\mathbf{x}_i = [x_i, y_i, z_i]^T, \forall i = 1, \dots, N$ and the coordinates of its 2D projection (Eq. 2) as $\mathbf{x}'_i = [x'_i, y'_i]^T, \forall i = 1, \dots, N$,

then this derivative is expressed as the block diagonal matrix

$$\frac{\partial \mathcal{P}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathcal{S}(\mathbf{p})} = \left[\frac{\partial \mathcal{P}}{\partial \mathbf{x}_1}, \frac{\partial \mathcal{P}}{\partial \mathbf{x}_2}, \dots, \frac{\partial \mathcal{P}}{\partial \mathbf{x}_N} \right] = \begin{bmatrix} \frac{\partial \mathbf{x}'_1}{\partial \mathbf{x}_1} & \mathbf{0}_{2 \times 3} & \cdots & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{2 \times 3} & \frac{\partial \mathbf{x}'_2}{\partial \mathbf{x}_2} & \cdots & \mathbf{0}_{2 \times 3} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \cdots & \frac{\partial \mathbf{x}'_N}{\partial \mathbf{x}_N} \end{bmatrix} \quad (5)$$

with size $2N \times 3N$. Each 2×3 derivative $\frac{\partial \mathbf{x}'_i}{\partial \mathbf{x}_i}$, $\forall i = 1, \dots, N$ is defined as

$$\frac{\partial \mathbf{x}'_i}{\partial \mathbf{x}_i} = \begin{bmatrix} \frac{\partial v_i^x}{\partial \mathbf{x}_i} v_i^z - v_i^x \frac{\partial v_i^z}{\partial \mathbf{x}_i} \\ f \frac{v_i^{z^2}}{v_i^{z^2}} \\ \frac{\partial v_i^y}{\partial \mathbf{x}_i} v_i^z - v_i^y \frac{\partial v_i^z}{\partial \mathbf{x}_i} \\ f \frac{v_i^{z^2}}{v_i^{z^2}} \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} \frac{\partial v_i^x}{\partial \mathbf{x}_i} &= [1 - 2q_2^2 - 2q_3^2, 2q_1q_2 - 2q_0q_3, 2q_1q_3 + 2q_0q_2] \\ \frac{\partial v_i^y}{\partial \mathbf{x}_i} &= [2q_1q_2 + 2q_0q_3, 1 - 2q_1^2 - 2q_3^2, 2q_2q_3 - 2q_0q_1] \\ \frac{\partial v_i^z}{\partial \mathbf{x}_i} &= [2q_1q_3 - 2q_0q_2, 2q_2q_3 + 2q_0q_1, 1 - 2q_1^2 - 2q_2^2] \end{aligned} \quad (7)$$

Derivative of perspective projection function with respect to camera parameters. From the definition of the perspective camera function in Eq. 2, this derivative has the form

$$\frac{\partial \mathcal{P}}{\partial \mathbf{c}} \Big|_{\mathbf{c}=\mathbf{c}} = \left[\frac{\partial \mathcal{P}}{\partial f}, \frac{\partial \mathcal{P}}{\partial q_1}, \frac{\partial \mathcal{P}}{\partial q_2}, \frac{\partial \mathcal{P}}{\partial q_3}, \frac{\partial \mathcal{P}}{\partial t_x}, \frac{\partial \mathcal{P}}{\partial t_y}, \frac{\partial \mathcal{P}}{\partial t_z} \right] \quad (8)$$

with size $2N \times n_c$.

The derivative with respect to the focal length is

$$\frac{\partial \mathcal{P}}{\partial f} = \left[\frac{v_1^x}{v_1^z}, \frac{v_1^y}{v_1^z}, \frac{v_2^x}{v_2^z}, \frac{v_2^y}{v_2^z}, \dots, \frac{v_N^x}{v_N^z}, \frac{v_N^y}{v_N^z} \right]^T \quad (9)$$

with size $2N \times 1$.

The derivatives with respect to the quaternion parameters q_1, q_2, q_3 are

$$\frac{\partial \mathcal{P}}{\partial q_1} = f \left[\frac{\frac{\partial v_1^x}{\partial q_1} v_1^z - v_1^x \frac{\partial v_1^z}{\partial q_1}}{v_1^{z^2}}, \frac{\frac{\partial v_1^y}{\partial q_1} v_1^z - v_1^y \frac{\partial v_1^z}{\partial q_1}}{v_1^{z^2}}, \dots, \frac{\frac{\partial v_N^x}{\partial q_1} v_N^z - v_N^x \frac{\partial v_N^z}{\partial q_1}}{v_N^{z^2}}, \frac{\frac{\partial v_N^y}{\partial q_1} v_N^z - v_N^y \frac{\partial v_N^z}{\partial q_1}}{v_N^{z^2}} \right]^T \quad (10)$$

$$\frac{\partial \mathcal{P}}{\partial q_2} = f \left[\frac{\frac{\partial v_1^x}{\partial q_2} v_1^z - v_1^x \frac{\partial v_1^z}{\partial q_2}}{v_1^{z^2}}, \frac{\frac{\partial v_1^y}{\partial q_2} v_1^z - v_1^y \frac{\partial v_1^z}{\partial q_2}}{v_1^{z^2}}, \dots, \frac{\frac{\partial v_N^x}{\partial q_2} v_N^z - v_N^x \frac{\partial v_N^z}{\partial q_2}}{v_N^{z^2}}, \frac{\frac{\partial v_N^y}{\partial q_2} v_N^z - v_N^y \frac{\partial v_N^z}{\partial q_2}}{v_N^{z^2}} \right]^T \quad (11)$$

$$\frac{\partial \mathcal{P}}{\partial q_3} = f \left[\frac{\frac{\partial v_1^x}{\partial q_3} v_1^z - v_1^x \frac{\partial v_1^z}{\partial q_3}}{v_1^{z^2}}, \frac{\frac{\partial v_1^y}{\partial q_3} v_1^z - v_1^y \frac{\partial v_1^z}{\partial q_3}}{v_1^{z^2}}, \dots, \frac{\frac{\partial v_N^x}{\partial q_3} v_N^z - v_N^x \frac{\partial v_N^z}{\partial q_3}}{v_N^{z^2}}, \frac{\frac{\partial v_N^y}{\partial q_3} v_N^z - v_N^y \frac{\partial v_N^z}{\partial q_3}}{v_N^{z^2}} \right]^\top \quad (12)$$

with sizes $2N \times 1$, where

$$\begin{aligned} \left[\frac{\partial v_i^x}{\partial q_1}, \frac{\partial v_i^x}{\partial q_2}, \frac{\partial v_i^x}{\partial q_3} \right] &= 2 [q_2 y_i + q_3 z_i, -2q_2 x_i + q_1 y_i + q_0 z_i, -2q_3 x_i - q_0 y_i + q_1 z_i] \\ \left[\frac{\partial v_i^y}{\partial q_1}, \frac{\partial v_i^y}{\partial q_2}, \frac{\partial v_i^y}{\partial q_3} \right] &= 2 [q_2 x_i - 2q_1 y_i - q_0 z_i, q_1 x_i + q_3 z_i, q_0 x_i - 2q_3 y_i + q_2 z_i] \\ \left[\frac{\partial v_i^z}{\partial q_1}, \frac{\partial v_i^z}{\partial q_2}, \frac{\partial v_i^z}{\partial q_3} \right] &= 2 [q_3 x_i + q_0 y_i - 2q_1 z_i, -q_0 x_i + q_3 y_i - 2q_2 z_i, q_1 x_i + q_2 y_i] \end{aligned} \quad (13)$$

$\forall i = 1, \dots, N$

The derivatives with respect to the translation vector $\mathbf{t}_v = [t_x, t_y, t_z]^\top$ are

$$\begin{aligned} \frac{\partial \mathcal{P}}{\partial t_x} &= f \left[\frac{1}{v_1^z}, 0, \frac{1}{v_2^z}, 0, \dots, \frac{1}{v_N^z}, 0 \right]^\top \\ \frac{\partial \mathcal{P}}{\partial t_y} &= f \left[0, \frac{1}{v_1^z}, 0, \frac{1}{v_2^z}, \dots, 0, \frac{1}{v_N^z} \right]^\top \\ \frac{\partial \mathcal{P}}{\partial t_z} &= -f \left[\frac{v_1^x}{v_1^{z^3}}, \frac{v_1^y}{v_1^{z^3}}, \frac{v_2^x}{v_2^{z^3}}, \frac{v_2^y}{v_2^{z^3}}, \dots, \frac{v_N^x}{v_N^{z^3}}, \frac{v_N^y}{v_N^{z^3}} \right]^\top \end{aligned} \quad (14)$$

with size $2N \times 1$ each.

Derivative of 3D shape generation function with respect to shape parameters. This is the derivative of Eq. 1 with respect to the shape parameters, which is simply

$$\left. \frac{\partial \mathcal{S}}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}} = \mathbf{U}_s \quad (15)$$

with size $3N \times n_s$.

1.1. Image Jacobians

Herein, we define the image Jacobians that are used in Sec. 3.2 (Eq. 18) of the main paper. Specifically, using the chain rule, the image Jacobian with respect to the shape parameters takes the form

$$\mathbf{J}_{\mathbf{F}, \mathbf{p}} = \left. \frac{\partial \mathbf{F}(\mathcal{W}(\mathbf{p}, \mathbf{c}))}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathcal{P}(\mathcal{S}(\mathbf{p}), \mathbf{c})} \left. \frac{\partial \mathcal{P}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathcal{S}(\mathbf{p})} \left. \frac{\partial \mathcal{S}}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}} \quad (16)$$

where the three components are given from Eqs. 4, 5 and 15, respectively. The Jacobian has size $CN \times n_s$.

Similarly, using the chain rule, the image Jacobian with respect to the camera parameters takes the form

$$\mathbf{J}_{\mathbf{F}, \mathbf{c}} = \left. \frac{\partial \mathbf{F}(\mathcal{W}(\mathbf{p}, \mathbf{c}))}{\partial \mathbf{c}} \right|_{\mathbf{c}=\mathbf{c}} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathcal{P}(\mathcal{S}(\mathbf{p}), \mathbf{c})} \left. \frac{\partial \mathcal{P}}{\partial \mathbf{c}} \right|_{\mathbf{c}=\mathbf{c}} \quad (17)$$

where the two components are given from Eqs. 4, and 8, respectively. The Jacobian has size $CN \times n_c$.

1.2. Landmarks Jacobians

Herein we define the Jacobians of the landmarks term that are used in Sec. 3.2 (Eq. 19) of the main paper. First, let us define the 3D landmarks generation function $\mathcal{S}_l(\mathbf{p}) : \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{3L}$ which generates a 3D instance of L sparse landmarks by slicing the linear combination defined in Eq. 1. Then the perspective camera function is defined as $\mathcal{W}_l(\mathbf{p}, \mathbf{c}) = \mathcal{P}(\mathcal{S}_l(\mathbf{p}), \mathbf{c})$ and it has the same formulation as in Eq. 2.

Specifically, using the chain rule, the landmarks Jacobian with respect to the shape parameters takes the form

$$\mathbf{J}_{\mathcal{W}_l, \mathbf{p}} = \left. \frac{\partial \mathcal{W}_l(\mathbf{p}, \mathbf{c})}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}} = \left. \frac{\partial \mathcal{P}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathcal{S}_l(\mathbf{p})} \left. \frac{\partial \mathcal{S}_l}{\partial \mathbf{p}} \right|_{\mathbf{p}=\mathbf{p}} \quad (18)$$

where the two components are given from Eqs. 5 and 15 for the L sparse landmarks. The Jacobian has size $2L \times n_s$.

Similarly, using the chain rule, the landmarks Jacobian with respect to the camera parameters takes the form

$$\mathbf{J}_{\mathcal{W}_l, \mathbf{c}} = \left. \frac{\partial \mathcal{W}_l(\mathbf{p}, \mathbf{c})}{\partial \mathbf{c}} \right|_{\mathbf{c}=\mathbf{c}} = \left. \frac{\partial \mathcal{P}}{\partial \mathbf{c}} \right|_{\mathbf{c}=\mathbf{c}} \quad (19)$$

which is given by Eq. 8 for the L sparse landmarks. It has size $2L \times n_c$.

2. Qualitative 3DMM fit

In the main paper, Fig. 4 shows a qualitative evaluation of three 3D Morphable Models on our new KF-ITW dataset. In Fig. 1 in this document, we present an example image from this experiment, along with the ground truth scan from Kinect Fusion, and the resulting fits from the three models under test.

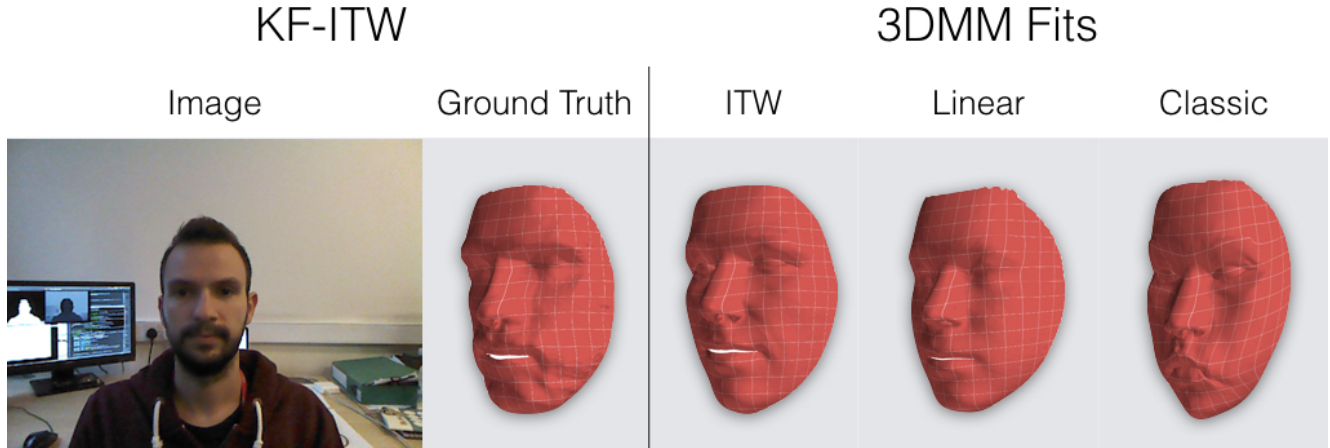


Figure 1. *Left*: An example RGB image and the complementary ground-truth 3D Kinect Fusion from our *KF-ITW* dataset. *Right*: The 3D shape recovered using the three methods under test.

3. Additional In-The-Wild texture extraction examples

In the main paper, Fig. 3 shows one example of how we extracted textures from “in-the-wild” images in order to form our robust texture basis. In Fig. 2 in this document, we present five more example texture extractions. For each, we show:

- The source “in-the-wild” image (left)
- The sampled texture from the image with mask (center)
- The final reconstructed texture used after completion with Robust PCA with Missing Values (right)

For clarification, we note the following:

- Our texture model is per-vertex — the “unwrapped” nature of the textures in these figures is purely a way to visualise these per-vertex texture models.
- The central column shows the colour per-vertex as extracted from the image when self occlusion is not taken into account. The transparent red overly shows the masked region as calculated by ray casting between each vertex and the camera.
- We actually used SIFT features for our texture model as explained in the main paper, but these RGB visualizations are much easier to interpret.

