NormalFusion: Real-Time Acquisition of Surface Normals for High-Resolution RGB-D Scanning Supplemental Document

Hyunho Ha[†] Joo Ho Lee[*] Andreas Meuleman[†] Min H. Kim[†]

[†] KAIST [*] University of Tuebingen

1. Hierarchical Nonlinear Optimization

Estimating two unknowns, normal and albedo, from reflected irradiance is a severely ill-posed problem in shape-from-shading (SfS) methods [4, 5]. In order to solve the nonlinear optimization problem of inverse rendering we formulate a total energy function (Equation (3) in the main paper) that seeks optimal depth and albedo $\mathbf{x} = \{\hat{D}, \mathbf{a}\}$ using the Gauss-Newton optimization [3] as follows:

$$E(\mathbf{x}) = \sum\nolimits_{\alpha=1}^{M} r_\alpha(\mathbf{x})^2, \tag{1}$$

where $r_\alpha(\mathbf{x})$ is a residual function of each energy term, and M is the total number of our energy term computed for each valid pixel. Note that given depth value $\hat{D}$ at pixel (i, j), we can calculate a 3D point $\mathbf{p}$ in the world space as follows:

$$\mathbf{p}(i,j) = \left[\frac{(i - c_x)}{f_x}, \frac{(j - c_y)}{f_y}, 1\right]^\intercal \cdot \hat{D}(i,j), \tag{2}$$

where f_x, f_y, and c_x, c_y are horizontal/vertical parameters of the focal length and the principal point of the depth camera, respectively. An unnormalized normal vector $\mathbf{n}'$ can be drived from the cross product of the horizontal/vertical partial derivatives $\mathbf{n}'(i,j)=(\mathbf{p}(i,j-1)-\mathbf{p}(i,j+1))\times(\mathbf{p}(i-1,j) - \mathbf{p}(i+1,j))$. Finally, we compute a unit normal $\mathbf{n}(i,j) = \frac{\mathbf{n}'(i,j)}{||\mathbf{n}'(i,j)||}$ from the normal vector $\mathbf{n}'$.

This energy function can be written in the matrix form as:

$$E(\mathbf{x}) = ||\mathbf{F}(\mathbf{x})||^2, \tag{3}$$

where $\mathbf{F}(\mathbf{x}) = [r_1(\mathbf{x}), \cdots, r_M(\mathbf{x})]^\intercal$. The optimal solution $\tilde{\mathbf{x}}$ can be obtained as follows:

$$\tilde{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} ||\mathbf{F}(\mathbf{x})||^2. \tag{4}$$

We can approximate the vector field $\mathbf{F}(\mathbf{x})$ around $\mathbf{x}_{\beta+1}$ using the first order of Taylor expansion:

$$\mathbf{F}(\mathbf{x}_{\beta+1}) \approx \mathbf{F}(\mathbf{x}_\beta) + \mathbf{J}(\mathbf{x}_\beta)\Delta\mathbf{x}_\beta, \tag{5}$$

where $\mathbf{J}$ is the Jacobian matrix of the residuals, and $\Delta\mathbf{x}_\beta = \mathbf{x}_{\beta+1} - \mathbf{x}_\beta$. Substituting $\mathbf{F}(\mathbf{x})$ in Equation (4) with in the optimization problem with Equation (5), the Taylor expansion, our final optimization becomes a linear least-squares problem.

$$\Delta\tilde{\mathbf{x}}_\beta = \operatorname*{arg\,min}_{\Delta\mathbf{x}} ||\mathbf{F}(\mathbf{x}_\beta) + \mathbf{J}(\mathbf{x}_\beta)\Delta\mathbf{x}_\beta||^2 \tag{6}$$

Optimal $\Delta\tilde{\mathbf{x}}_\beta$ can be calculated by solving the following linear system:

$$\mathbf{J}(\mathbf{x}_\beta)^T\mathbf{J}(\mathbf{x}_\beta)\Delta\tilde{\mathbf{x}}_\beta = -\mathbf{J}(\mathbf{x}_\beta)^T\mathbf{F}(\mathbf{x}_\beta). \tag{7}$$

Since this linear system is very large, it needs to be solved by an iterative method. We implement a GPU-friendly version of the preconditioned conjugate gradient method [3] with two sparse matrix-vector multiplication kernels [6] for efficiently solving of the system.

2. Normal/Albedo Blending

Once we know the spatially-varying warp function $\mathcal{W}^t$, we are ready to blend normals $\hat{N}^t$ and albedos $\hat{A}^t$ with the transferred normals $\dot{N}^t$ and albedos $\dot{A}^t$ at the current frame t to the canonical texture space $\bar{N}^t$ and $\bar{A}^t$, respectively. For each texel of a 3D point $\mathbf{p}$ in the canonical space of TSDFs, we evaluate the spatial resolution and registration certainty of each image pixel by computing blending weights following the current methods [2, 1] as follows:

$$\begin{aligned} \bar{N}^t(\mathbf{p}) &= \frac{\Psi^{t-1}(\mathbf{p})\dot{N}^t(\mathbf{p}) + \psi(\mathbf{p})\hat{N}^t(\tilde{\mathbf{u}})}{\Psi^{t-1}(\mathbf{p}) + \psi(\mathbf{p})}, \\ \bar{A}^t(\mathbf{p}) &= \frac{\Psi^{t-1}(\mathbf{p})\dot{A}^t(\mathbf{p}) + \psi(\mathbf{p})\hat{A}^t(\tilde{\mathbf{u}})}{\Psi^{t-1}(\mathbf{p}) + \psi(\mathbf{p})}, \end{aligned} \tag{8}$$

where $\tilde{\mathbf{u}}$ is a pixel that corresponds to point $\mathbf{p}$ via the warping function $\mathbf{W}$ at the current frame t. In addition, weight Ψ is the accumulated weight for normal/albedo blending at the current frame: $\Psi^t(\mathbf{p}) = \min(\Psi^{t-1}(\mathbf{p}) + \psi(\mathbf{p}), \psi_{\text{max}})$, where ψ_{max} is a predefined parameter that controls the upper bound of the blending weight, $\psi(\mathbf{p})$ is the blending weight for a given camera pose. The current blending weight $\psi(\mathbf{p})$ can be computed by accounting for the area size, the camera angle, and occlusion:

$$\psi(\mathbf{p}) = \psi_{\text{area}}(\mathbf{p}) \cdot \psi_{\text{angle}}(\mathbf{p}) \cdot \psi_{\text{occ}}(\mathbf{p}), \tag{9}$$

following the weight formulae defined in [2].

We compute the area ψ_{area} and the angle weight ψ_{angle} as:

$$\begin{aligned} \psi_{\text{area}}(\mathbf{p}) &= \exp(-((1 - (\frac{z_{\text{min}}}{z})^2 \mathbf{n} \cdot \mathbf{o})/\sigma_{\text{area}})^2), \\ \psi_{\text{angle}}(\mathbf{p}) &= \exp(-((1 - \mathbf{n} \cdot \mathbf{o})/\sigma_{\text{angle}})^2), \end{aligned} \tag{10}$$

where z is the depth value of the $\mathbf{p}$, z_{min} is the minimum depth of the reconstructed scene, $\mathbf{o} = (\mathbf{p} - \mathbf{c})/||\mathbf{p} - \mathbf{c}||$ is the camera view vector, σ_{area} and σ_{angle} are hyperparameters that controls the smoothness of Gaussian weights, they are set to 1.0 and 0.5, respectively, and ψ_{occ} is a soft-occlusion weight factor, following [2, 1].

References

[1] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 425–432, 2001.

[2] Joo Ho Lee, Hyunho Ha, Yue Dong, Xin Tong, and Min H Kim. Texturefusion: High-quality texture acquisition for real-time rgb-d scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1272–1280, 2020.

[3] Daniel Weber, Jan Bender, Markus Schnoes, André Stork, and Dieter Fellner. Efficient gpu data structures and methods to solve sparse linear systems in dynamics applications. In *Computer Graphics Forum*, volume 32, pages 16–26. Wiley Online Library, 2013.

[4] Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics (ToG)*, 33(6):1–10, 2014.

[5] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG)*, 34(4):1–14, 2015.

[6] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):1–12, 2014.