Merge2-3D: Combining Multiple Normal Maps with 3D Surfaces

Sema Berkiten
Princeton University
berkiten@princeton.edu

Xinyi Fan
Princeton University
xinyi@princeton.edu

Szymon Rusinkiewicz
Princeton University
smr@princeton.edu

***Abstract*—We propose an approach to enhance rough 3D geometry with fine details obtained from multiple normal maps. We begin with unaligned 2D normal maps and rough geometry, and automatically optimize the alignments through 2-step iterative registration algorithm. We then map the normals onto the surface, correcting and seamlessly blending them together. Finally, we optimize the geometry to produce high-quality 3D models that incorporate the high-frequency details from the normal maps. We demonstrate that our algorithm improves upon the results produced by some well-known algorithms: Poisson surface reconstruction [1] and the algorithm proposed by Nehab et al. [2].**

***Keywords*-2.5/3D alignment, mesh enhancement, surface normals.**

I. Introduction

Geometry acquisition has become increasingly popular in computer graphics and vision, with demand for high-quality models driven by advances in 3D printing, realistic real-time renderings of 3D self-avatars in video games, digital libraries for historical objects etc. Producing such detailed geometry requires not only precise acquisition devices, but also a pipeline of processing steps including registration of multiple views (since an entire 3D model typically cannot be acquired in one sweep) and merging of different scans. In this work we present a high-quality reconstruction pipeline that combines data from two different sources: a rough 3D model obtained with an active (laser-stripe or structured-light) scanner and a few 2D normal maps calculated by a photometric stereo algorithm.

The motivation for combining these two kinds of data is strong. Normal maps are easily obtained at high resolution using active shape from shading (photometric stereo). They can contain a wealth of information about the fine details of a surface, but suffer from low-frequency bias. 3D scans, in contrast, are difficult to obtain at extremely high resolutions, but they contain all of the information about the topology and overall coarse-scale shape of the surface.

Although direct reconstruction of a 3D surface from normal maps is possible, in practice the results are far from satisfactory because of bias present in low frequencies, noise, and insufficient constraints among disconnected patches. This problem was addressed by the work of Nehab et al. [2], which proposed combining the low-frequency components of the geometry with the high-frequency details of the normal maps. However, that work assumed perfect alignment between the 3D geometry and a single 2D normal map. Also, their optimization for full 3D models (as opposed to height fields) is an approximation that forces the surface to be locally planar, resulting in over-smoothing of high-frequency details.

In this paper, we present a complete system to enhance rough geometry with multiple detailed normal maps. The pipeline consists of four main steps:

- ***Acquisition*** of 3D rough geometry, as well as extraction of 2D normal maps from images under varying light directions with respect to the object (Section III);
- ***Alignment*** of the normal maps to the rough geometry without any initial alignment, (Section IV);
- ***Blending*** of multiple normal maps to produce a seamless normal field over the surface (Section V); and
- ***Optimization*** of the 3D model to incorporate high-frequency details from the normal maps (Section VI).

Our pipeline takes two types of inputs (multiple 2D normal maps and coarse 3D geometry), and produces an enhanced 3D model. The advantage of our system is that it requires neither initial alignment nor resolution/precision compatibility between different data types.

Our main contributions are four-fold:

- A new feature detector tailored for normal maps.
- An algorithm to align 2D normal maps to a 3D surface by iteratively minimizing dissimilarity of normals.
- Seamless mapping of the aligned 2D normal maps to the 3D geometry.
- A method for combining the original 3D positions with the mapped normals.

II. Related Work

Registration: Pairwise alignment have been common problems in many research areas such as computer vision/graphics, medical imaging, robotic vision, etc. The types of input signals can be varied, and can include 2D images (visible/infrared image, x-ray, magnetic resonance imaging, computed tomography , positron emission tomography etc.) and/or 3D geometry (point clouds, range maps, or surface meshes). Registration for many combinations of these input types has been studied in the literature, but there is little previous work addressing the problem of aligning normal maps to rough 3D surface geometry.

Surface normals are frequently used for 3D-to-3D registration, in work such as [3], [4], [5]. Related studies on invariant features and motion estimation for small nonrigid deformations between 3D models have been presented in various publications such as [6], [7], [8], [9], [10]. Wang et al.[11] proposed a method to register 3D face template to normal maps by selecting 8 points manually and deforming the generic model. All of these works showed that normal-guided registration is more robust than others.

The closest related problem to ours is that of aligning images to geometry, and Viola and Wells [12] and Corsini et al. [13] proposed methods based on mutual information. While these techniques could be adapted for normal maps, both of them assume that the details in the 3D model are similar to those in the images. In our case, however, we accept rough 3D geometry as input, and develop a method of reliably aligning it to (possibly warped) normal maps.

Blending: Pérez et al. [14] proposed using the Poisson equation to blend two images seamlessly, and Chuang et al. [15] extended it to blending multiple texture images on a 3D object using the Laplace-Beltrami operator. Li et al. [16] also used Poisson blending, after recovering reflectance and global illumination. We extend these kinds of methods to blending normals, observing that the low-frequency bias in normal maps should be corrected before blending.

Combining 3D positions and normals: Nehab et al. [2] proposed a linear formulation to combine the high-frequency details of normals and low-frequency shape from 3D positions. However, their acquisition setup used the same camera to capture both range scans and normal maps (using photometric stereo [17]). That is why they assumed that their different data types were perfectly aligned, and they did not take into account missing parts or holes in the 3D positions. However, the condition of perfect alignment between the two different types of data cannot always be satisfied, as when 3D and normals are acquired using completely independent devices. Building upon the work of Nehab et al., our system does not require or assume any alignment or similar resolution between the 3D positions and the normal maps.

Yu et al. [18] proposed an approach to improve the quality of normal maps and consequently the geometry acquired with MS Kinect. They try to enhance the quality of the normals which is calculated from 3D data (depth); while we do not focus on that problem, rather we assume that we have normal maps with higher quality than the geometry and use that extra information to improve the geometry.

A more general method for combining positions and normals is provided by Poisson surface reconstruction [1], [19]. This is a generic surface reconstruction algorithm that takes as input a point cloud with corresponding normals. However, if the accuracies and resolutions of the positions and normals are different, Poisson surface reconstruction often generates a smoothed result, as we show later.

III. Data Acquisition

In this work, we experiment with both synthetic and acquired data. The synthetic dataset begins with a ground-truth 3D model of the armadillo, and acquires the following:

- **2D normal maps:** We render normal maps from multiple views and save the viewing directions for evaluation.
- **Rough 3D model:** We smooth the original 3D model with Gaussian filter of $\sigma = 4\times$(average edge length).

Real data acquisition uses two separate capture stages:

- **2D normal maps:** To obtain each normal map we capture a series of D-SLR images of the object with the same acquisition setup in [20] ; and use a variant of the photometric stereo algorithm [17].
- **3D geometry acquisition:** Of the many sensors available on the market for 3D acquisition, we experiment with two very different ones. First, we use a relatively high-resolution NextEngine scanner, which is based on laser stripe triangulation. We follow the steps in [21] to reconstruct the surface. Secondly, to explore even lower-resolution geometric input, we have also experimented with Kinect [22], with the Skanect [23] software used to reconstruct the surface from the captured data.

IV. Alignment

Unlike other common alignment work-flows such as ICP [24], [25] which start with a user-provided rough initial alignment , we design a 2-step alignment algorithm to have a fully automatic system: i) estimation of an initial alignment via a feature based algorithm , ii) iterative refinement by minimizing the dissimilarity between normals.

A. Initial Alignment

Inspired by SIFT [26], we design an adapted feature (nSIFT), which detects scale-invariant feature points and computes feature descriptors on normal maps. Denoting the normal map by $N(x, y)$ and the x, y components of it by $N_x(x, y)$ and $N_y(x, y)$ (which are embedded in R and G channels), we first construct a Gaussian scale space for $N_x(x, y)$ and $N_y(x, y)$. In order to detect stable keypoint locations in the scale space, we approximate the scale-normalized Laplacian of Gaussian as follows:

$$\sigma G(x, y, \sigma) * \frac{\partial N_x(x, y)}{\partial x} + \sigma G(x, y, \sigma) * \frac{\partial N_y(x, y)}{\partial y} \quad (1)$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$. Similar to [26], keypoints are detected by looking for local extrema in the 3×3 neighborhood at the scale-space. The orientation of each keypoint is assigned by the x and y components of the normal at that point and the descriptor is obtained by computing a 3D spatial histogram of the normals in the neighborhood of each keypoint, rotated by its orientation. Since the real normal map has more details than the rendered normal map we first smooth the real normal map by a Gaussian kernel of $\sigma = 3$; and during feature matching, we

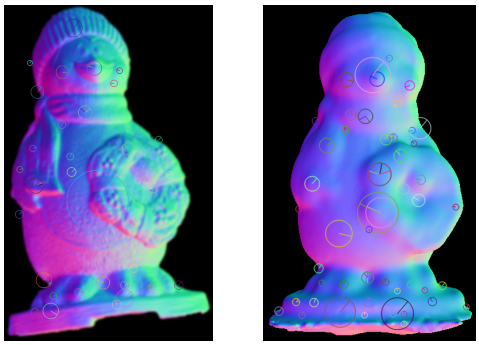


Figure 1. nSIFT feature points detected on pre-smoothed real normal map (left) and nSIFT feature points detected on rendered normal map (right). (See the supplemental material for more results.)

only use features detected in higher scales (with larger σ) of the real normal map to reduce false positive matches. After computing the matches among features using the nearest neighbor matcher, we use the RANSAC framework [27] to eliminate outliers. We also incorporate the location of feature points relative to the center of the model's bounding box to further eliminate outliers. The algorithm 1 summarizes our feature-based alignment.

Algorithm 1 Feature-based Rough Alignment

Input: 3D model M, real normal map N, view sphere V, sample rate r.

repeat

Sample V at rate r to obtain set of viewpoints V_r

$\forall$ viewpoint $v \in V_r$

N_v = rendered surface normals of M from v

Find N's nearest neighbor N_v^* by nSIFT

Feature matching in RANSAC framework

$V \leftarrow N_v^*$'s neighborhood, $\quad r \leftarrow 2r$

until $\parallel N$ - $N_v^* \parallel < \tau$

B. Refinement

Inspired by ICP [24], [25], we propose an iterative alignment algorithm that begins with a rough alignment between a 2D normal map and 3D model and improves the registration by minimizing an energy function.

To evaluate the energy function, we transform the 3D model given some candidate alignment, then render it into image space using the projection matrix of the normal map. The mesh is colored based on its normal map (transformed by the rotational portion of the alignment transform), so that the rendered image allows us to determine the mesh normal $\mathbf{N}_m$ that is mapped to each pixel (i, j). We then evaluate the dissimilarity between those mesh normals and the corresponding normals $\mathbf{N}_{img}$ from the normal map:

$$\mathcal{E} = \sum_{i=1}^{\#rows} \sum_{j=1}^{\#cols} f\big(\mathbf{N}_m(i,j), \mathbf{N}_{img}(i,j)\big). \qquad (2)$$

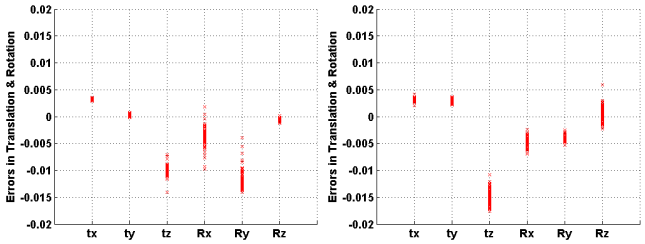


Figure 2. Alignment error for the armadillo dataset from the front (left) and the side (right). Error is measured for six parameters of rigid transformation: errors in translation (t_x, t_y, t_z) are measured as a fraction of the diameter of the bounding sphere of the model, while errors in rotation (R_x, R_y, R_z) are measured in radians.

The dissimilarity function f is defined as:

$$f(\mathbf{a}, \mathbf{b}) = \begin{cases} 0 & \text{if both } \mathbf{a} \text{ and } \mathbf{b} \text{ are masked,} \\ \pi & \text{if either } \mathbf{a} \text{ xor } \mathbf{b} \text{ is masked,} \\ |w \times cos^{-1}(\mathbf{a} \cdot \mathbf{b})| & \text{otherwise.} \end{cases} \qquad (3)$$

The masks hold the locations of background pixels, so that this function imposes a large penalty if a pixel is part of the background of either the model or normal map, but not both. The weight w is calculated as $w = 1 - (\mathbf{b} \cdot \mathbf{C})$, where $\mathbf{C}$ is the camera direction. So, if the normal is facing towards the camera it gets a higher weight, and vice versa.

To optimize this energy function we need to transform the model and render the normal map, making the problem non-linear. We therefore use the Nelder-Mead algorithm [28], which initializes a simplex in the 6-dimensional rotation-translation space and heuristically grows, moves, and shrinks the simplex. For added robustness we perform randomized restart, computing a new random simplex (with maximum extent half the size as at the previous restart) once the simplex becomes sufficiently small, or if the algorithm performs more than 8 "shrink" operations in a row. We also re-initialize the simplex if 100 energy function calculations have been performed since the last restart. We terminate the algorithm after 8 restarts. See the supplemental material for the the energy function over time.

To measure the accuracy of the alignment, we use two synthetic normal maps (front and side) obtained from the armadillo model. For each of 100 trials, we randomly perturb the ground-truth alignment by up to 6% of the radius of the model's bounding sphere to obtain an initial registration, and then run our iterative optimization to align the smoothed 3D mesh to the normal map. Scatter plots of the errors in translation (t_x, t_y, t_z) and rotation (R_x, R_y, R_z) are shown in Figure 2, where translation is measured as a fraction of the diameter of the model's bounding sphere and rotation is measured in radians.

Note that different components of the alignment are recovered with different accuracies, reflecting the screen-space perturbation resulting from changing each parameter. For example, for the front view (shown in Figure 2, left), it is

more difficult to recover the z component of translation, corresponding to moving along the camera's direction of view, than it is to find the x and y components.

V. Blending Normal Maps

After the normal maps are aligned to the coarse 3D geometry, we would like to assign a single normal to each vertex on the 3D surface (after isometrically re-meshing the surface to have the desired output resolution). If the normal maps were perfect, and perfectly aligned, we could assign to each surface point a normal from an arbitrary normal map that observed it (e.g., the one that observed it most head-on). In practice, however, both of these hypotheses are false. Normal maps are *not* perfect due to limitations with the photometric stereo algorithm, which imperfectly accounts for shadows, non-Lambertian materials, non-opaque surfaces, and inter-reflection. The alignment is *not* perfect, since we are aligning to coarse 3D meshes that sometimes do not contain enough detail to reliably constrain the alignment. So, if we only assigned a single normal to each surface point, the resulting normal field would be discontinuous. Averaging all the normals projecting to a given 3D surface point would not solve the problem, since we would still see seams where we switched between including, and not including, the contribution of each normal map.

A partial solution to this problem was proposed by Nehab et al.[2]. They observed that the dominant error in normal maps is low-frequency. In contrast, while scanned 3D models might have high-frequency noise or lack high-frequency detail, their low frequencies are reliable. It is therefore possible to *correct* the normal maps by replacing their low-frequency components by the low-frequency component of normals obtained from the 3D model. We adopt this strategy, including it in all of the results shown in this paper. Nevertheless, we find that this correction is not always sufficient for avoiding seams between data from different maps.

Inspired by Poisson image editing [14], we propose blending between the (already-corrected) normal maps in a way that preserves their gradients. To do this we optimize an energy function with two terms, a vertex term that pushes the result towards a consensus (weighted average) among samples, and an edge term that preserves differences between neighboring normals:

$$\mathcal{E} = \lambda \mathcal{E}_{vert} + (1-\lambda)\,\mathcal{E}_{edge}, \tag{4}$$

where λ controls the relative significance of the terms. The vertex energy seeks to minimize the weighted difference between the mesh normal $\boldsymbol{n_v}$ and the projected normal from each normal map $\boldsymbol{n_v^s}$:

$$\mathcal{E}_{vert} = \sum_{v=1}^{\#vert} \sum_{s=1}^{\#maps} w_v^s \left\| \boldsymbol{n_v} - \boldsymbol{n_v^s} \right\|^2, \tag{5}$$

where higher weight is given to normal maps that see the surface more directly: $w_v^s = (\boldsymbol{d_{viewer}} \cdot \boldsymbol{n_v^s})^p$.

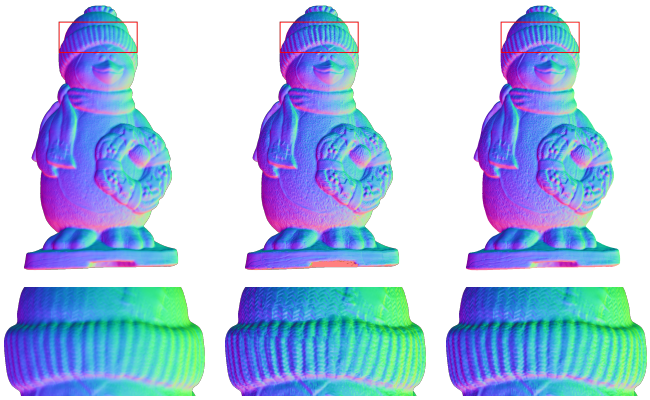


Figure 3. Comparison of different blending methods. ***Left:*** averaging all sample normals. **Center:** choosing the normal which is captured from the most direct view point. **Right:** our proposed blending method. Normals are false-colored by embedding in the RGB color space.

The edge energy ensures that local gradients among the samples and over the surface will be consistent:

$$\mathcal{E}_{edge} = \sum_{e=1}^{\#edge} \sum_{s=1}^{\#maps} \left[(\boldsymbol{n_{ev1}} - \boldsymbol{n_{ev2}}) - (\boldsymbol{n_{ev1}^s} - \boldsymbol{n_{ev2}^s}) \right]^2, \tag{6}$$

$\boldsymbol{n_{ev1}}$, $\boldsymbol{n_{ev2}}$ are the normals at the endpoints of edge e.

This energy function can be minimized with linear least squares, and it is sparse because there is only one coefficient for each vertex constraint and two coefficients for each edge constraint. We set the parameter $\lambda = 0.4$ to blend multiple normal maps for all results demonstrated in the paper.

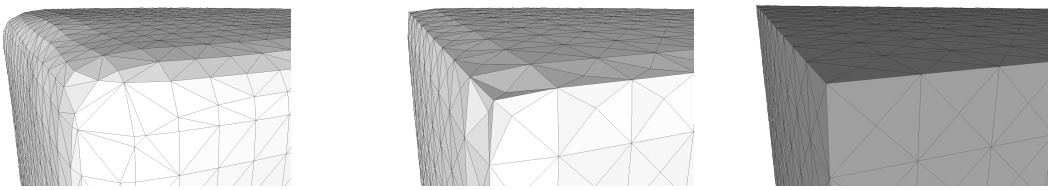
We compare our method to two alternative approaches: averaging samples, or just taking the normal captured from the most direct view. As shown in Figure 3, both of these methods create dramatic transitions, especially close to silhouettes. Our method successfully blends samples coming from different normal maps, and handles silhouettes well.

VI. Surface Enhancement

Given high-quality normals and rough vertex positions, the true surface can be approximated by Poisson surface reconstruction[1], or using the method of Nehab et al.[2]. However, one of the most common problems with these methods is loss of some high-frequency information. Even though the work of Nehab et al. was designed to compensate for this problem, their full-model optimization (as opposed to their method for height fields) sacrifices some high-frequency details to make the problem linear, by assuming that the surface is locally planar.

To overcome this drawback, we perform a non-linear optimization, minimizing an energy function consisting of two terms (similar to the previous work [2]) as follows:

$$\mathcal{E} = \lambda \mathcal{E}_p + (1-\lambda)\mathcal{E}_n, \tag{7}$$



Iterated Poisson surface reconstruction Iterated Nehab et al. Our Method

Figure 4. Comparison of results of Poisson surface reconstruction, Nehab et al., and our method on a cube. The ground-truth vertex positions and normals of the cube were used as input.

where $\mathcal{E}_p$ is the position term and $\mathcal{E}_n$ is the normal term (see [2] for notations). These are formulated as follows:

$$\mathcal{E}_p = \sum_{v=1}^{\#vert} \|\boldsymbol{p_v} - \boldsymbol{p_v^{mesh}}\|^2 \tag{8}$$

$$\mathcal{E}_n = \sum_{v=1}^{\#vert} -\|\boldsymbol{n_v} \cdot \boldsymbol{n_v^{measured}}\|^2, \tag{9}$$

where the optimized positions $\boldsymbol{p_v}$ are intended to match those coming from the mesh, and the optimized normals $\boldsymbol{n_v}$ are intended to match those coming from the normal maps (after alignment, low-frequency correction, and blending). We have also experimented with calculating $\mathcal{E}_n$ by summing over the faces rather than over the vertices, but found that this increased computational complexity with little benefit visible in the final result.

We minimize this energy function with the conjugate gradient method, using analytically-computed Jacobians and a line search at each iteration (see the supplemental material for details). We use the parameter $\lambda = 0.4$ for the enhancement in both our method and the algorithm of Nehab et al. [2] for comparison.

Figure 5 shows a quantitative comparison of our method to Poisson surface reconstruction [1] and the method of Nehab et al. [2]. We found that the latter two methods exhibit less smoothing if the surface positions are close to the correct ones, and so we actually run these methods in an *iterative* manner. We start with a smoothed armadillo model and normal maps obtained from the original mesh. We then repeatedly assign the original normals onto the enhanced model from the previous iteration (using ground-truth alignment), and reconstruct/enhance the surface, until each algorithm converges.

The visualizations in the figure are color-coded by perpendicular distance from the original model to the enhanced surface, with the color mapping shown at the bottom of Figure 5. It is obvious that Poisson reconstruction smooths out high frequency details, and flattens parts with high curvature such as the tips of the ears, hands, noise, and feet. Note, however, that our scenario was not really the one for which this algorithm was designed. Note also that we do not use the more recent Screend Poisson Surface Reconstruction algorithm [19] because it gives more weight to the vertex positions, which would result in even worse results.

The algorithm by Nehab et al., which *was* designed for this specific problem, produces better results. However, our method, which performs the correct non-linear optimization, loses fewer high-frequency details and produces results very close to the true surface.

In Figure 4, we used a cube to test an extreme case. We used the original positions and normals of the cube as input, in order to observe the behavior of the three algorithms in a controlled setting. The previous algorithms are unable to preserve the sharp corners of the cube, introducing significant smoothing. In contrast, our method neither introduces any artifacts nor smooths over the corners.

VII. Results and Discussion

Figure 6 demonstrates results for the three algorithms described above, using source data from two different types of 3D acquisition techniques: moderate-resolution models acquired with the NextEngine laser scanner and coarse models obtained with MS Kinect. The leftmost column in Figure 6 shows the results and close-up images for the iterative Poisson reconstruction; the middle column shows the results of the algorithm proposed by Nehab et al. [2]; and finally the rightmost column shows our results. All of these used our proposed alignment and blending algorithms.

Our method preserves more of the high-frequency information from the normal maps, as compared to the other two methods. This is particularly visible in the close-ups in the last two rows of Figure 6, which used the comparatively coarse Kinect reconstructions as input. Also note the additional detail in the fingers of the soldier (fourth row) preserved by our method. Of course, in some cases this may also mean that our method preserves more noise, since it does not smooth the normal maps — this effect is especially visible in the third row of Figure 6. We include high-resolution renderings and time analysis in the materials.

VIII. Conclusion and Future Work

In this work, we proposed a complete pipeline for combining coarse 3D models with multiple unaligned 2D normal maps, including acquisition, fully automatic alignment, blending the normal maps, and enhancing the model to incorporate high-frequency normals. We demonstrated that our algorithm outperforms two well-known methods: Poisson surface reconstruction from point clouds with normals and the work proposed by Nehab et al.

We believe that there are several directions for future work. First, we could use more information from the normal

maps, such as extracting the object silhouettes from the normal maps and constraining the 3D object to match those silhouettes. We could also use the available normal maps to synthesize plausible surface detail (texture) in locations where the normal maps did not cover the object. Finally, we wish to explore large-scale applications, such as combining normal maps of a building with 3D point clouds from Google Street View or enhancing only a part of a big object.

IX. Acknowledgment

This work was partially supported by the ISTC-VC and NSF grants CCF-1012147 and CCF-1027962.

References

[1] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Symposium on Geometry Processing*, 2006, pp. 61–70. 1, 2, 4, 5

[2] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3D geometry," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 536–543, 2005. 1, 2, 4, 5

[3] C. Lara, L. Romero, and F. Calderón, "A robust iterative closest point algorithm with augmented features," in *MICAI 2008: Advances in Artificial Intelligence, LNCS 5317*. Springer, 2008, pp. 605–614. 2

[4] D. Münch, B. Combès, and S. Prima, "A modified ICP algorithm for normal-guided surface registration," in *Proc. SPIE 7623, Medical Imaging 2010: Image Processing*, 2010. 2

[5] H. Mohammadzade and D. Hatzinakos, "Iterative closest normal point for 3d face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 381–397, Feb 2013. 2

[6] C. Kambhamettu, D. B. Goldgof, and M. He, "On a study of invariant features in nonrigid transformations," in *Proc. Qualitative Vision*, 1993, pp. 118–127. 2

[7] C. Kambhamettu, D. Goldgof, and M. He, "Determination of motion parameters and estimation of point correspondences in small nonrigid deformations," in *Computer Vision and Pattern Recognition*, Jun 1994, pp. 943–946. 2

[8] P. Laskov and C. Kambhamettu, "Comparison of 3D algorithms for non-rigid motion and correspondence estimation," in *Proc. British Machine Vision Conference*, 2001. 2

[9] P. Laskov and C. Kambhamettu, "Curvature-based algorithms for nonrigid motion and correspondence estimation," *PAMI*, pp. 1349–1354, 2003. 2

[10] M. Li, C. Kambhamettu, and M. Stone, "Nonrigid motion recovery for 3D surfaces," *Image and Vision Computing*, vol. 25, no. 3, pp. 250–261, 2007. 2

[11] Z. Wang, M. Grochulla, T. Thormahlen, and H. P. Seidel, "3D face template registration using normal maps," in *Proc. International Conference on 3D Vision*, 2013. 2

[12] P. Viola and W. Wells, "Alignment by maximization of mutual information," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, Jun 1995, pp. 16–23. 2

[13] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno, "Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1755–1764, 2009. 2

[14] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003. 2, 4

[15] M. Chuang, L. Luo, B. J. Brown, S. Rusinkiewicz, and M. Kazhdan, "Estimating the Laplace-Beltrami operator by restricting 3D functions," in *Proceedings of the Symposium on Geometry Processing*, vol. 28, 2009, pp. 1475–1484. 2

[16] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev, "3D self-portraits," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–9, 2013. 2

[17] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, vol. 19, no. 1, pp. 139–144, 1980. 2

[18] L.-F. Yu, S.-K. Yeung, Y.-W. Tai, and S. Lin, "Shading-based shape refinement of rgb-d images," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1415–1422. 2

[19] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013. 2, 5

[20] C. Toler-Franklin, A. Finkelstein, and S. Rusinkiewicz, "Illustration of complex real-world objects using images with normals," in *Proc. NPAR*, 2007, pp. 111–119. 2

[21] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doumas, S. Rusinkiewicz, T. Weyrich, and T. Weyrich, "A system for high-volume acquisition and matching of fresco fragments: reassembling Theran wall paintings," *ACM Trans. Graph.*, vol. 27, no. 3, p. 84, 2008. 2

[22] Microsoft, Inc., "Kinect," www.xbox.com/en-US/kinect, 2010. 2

[23] Occipital, Inc., "Skanect," skanect.manctl.com, 2013. 2

[24] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. PAMI*, vol. 14, no. 2, 1992. 2, 3

[25] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. Robotics and Automation*, 1991, pp. 2724–2729. 2, 3

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 2

[27] 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*, 1981. 3

[28] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, Jan 1965. 3

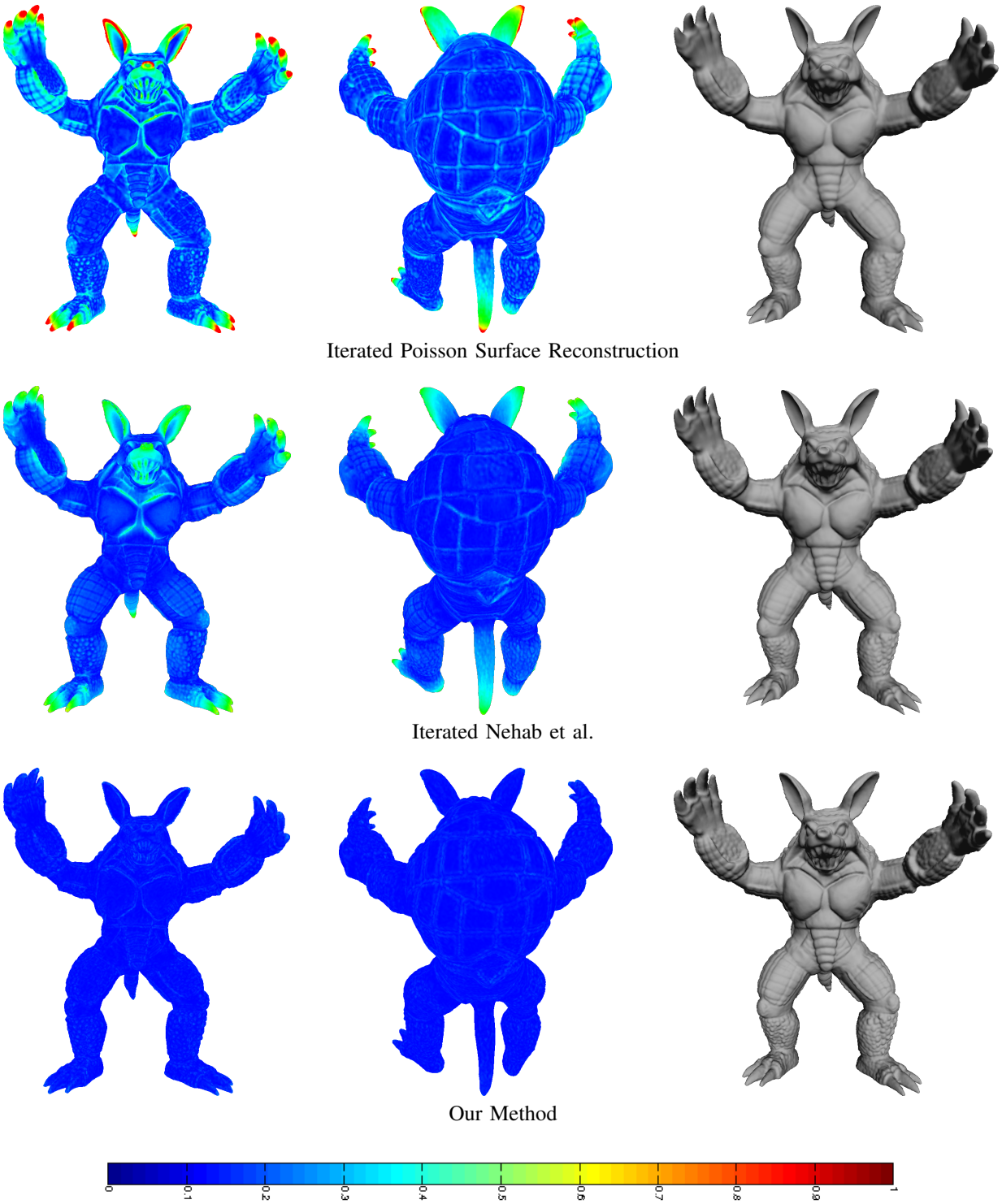


Figure 5. Comparisons of our method to Poisson surface reconstruction and Nehab et al. Blue represents zero distance between the reconstructed surface and ground truth, while red is greater than or equal to 0.001 of the radius of the bounding sphere.

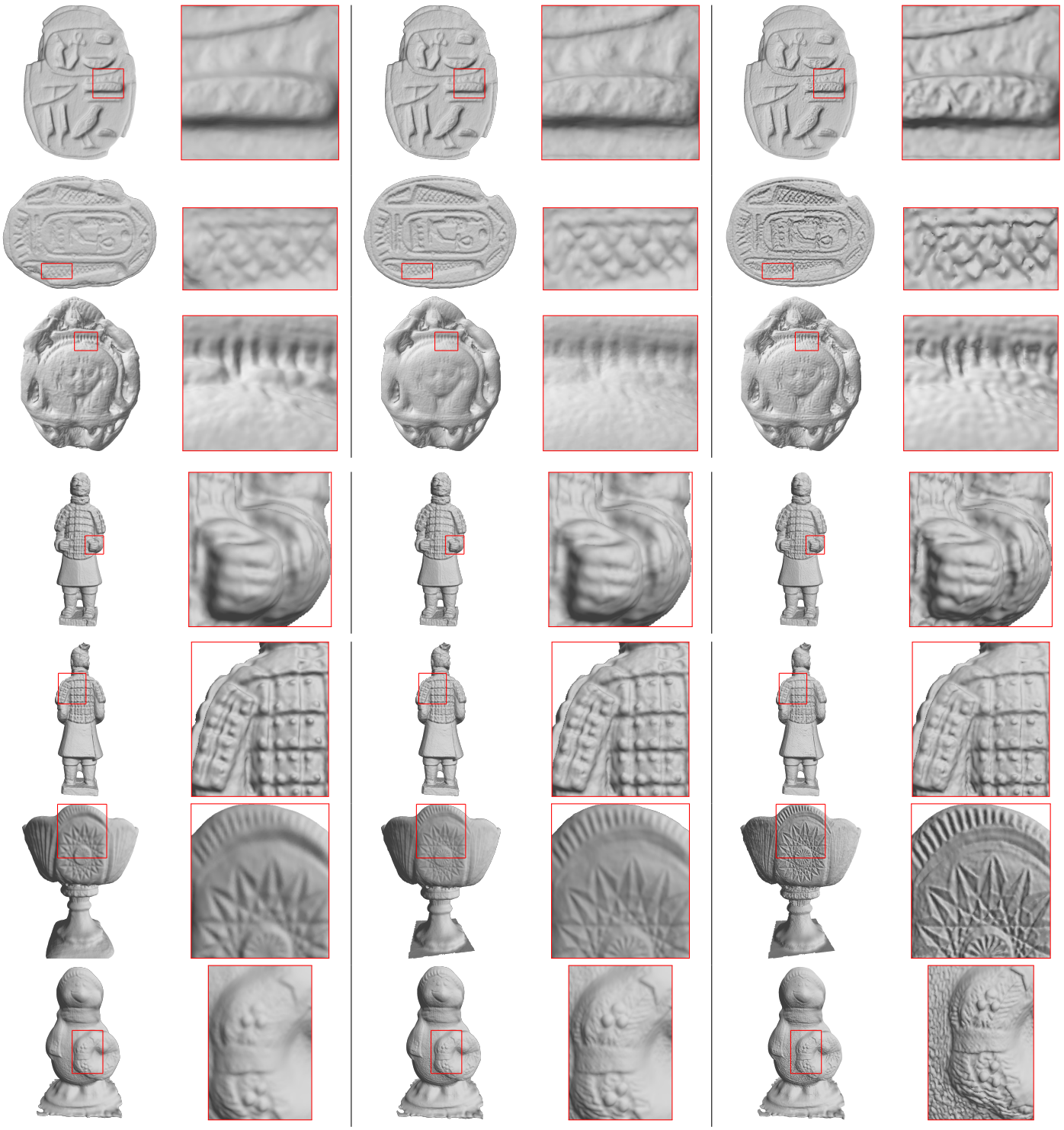


Figure 6. Comparison of results of Poisson surface reconstruction, Nehab et al., and our method. The 3D models in the first five rows are acquired with the NextEngine laser scanner, while the last two are obtained with MS Kinect.