FOUND: Foot Optimization with Uncertain Normals for Surface Deformation Using Synthetic Data
Supplementary material

Oliver Boyne Gwangbin Bae James Charles Roberto Cipolla
Department of Engineering, University of Cambridge, U.K.
{ob312, gb585, jjc75, rc10001}@cam.ac.uk

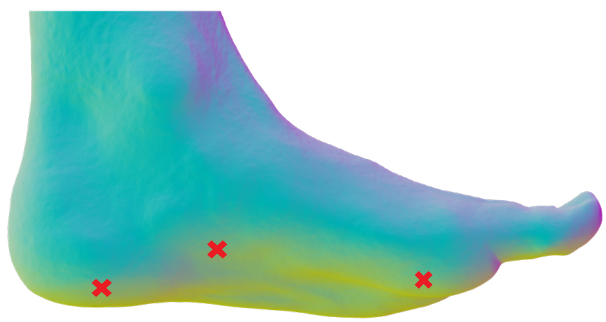

Figure 1. We show the 'arch' keypoints labelled on an example foot, showing the surface normals.

1. Keypoint definitions

We define 12 keypoints on the foot:

- **Toes** (5) - We label the most extremal point on each of the 5 toes
- **Width** (2) - We label the 'inner extrema' and 'outer extrema', the widest points on the front of the foot
- **Heel** (2) - We label two keypoints on the heel - one corresponding to the extremal (furthest 'backwards') point at the bottom of the foot, and one at the point where this contacts the floor
- **Arch** (3) - We label three keypoints to define the arch of the foot - we achieve this by viewing the surface normals of the foot, and identifying the arch by the transition of colour from yellow (pointing downwards) to blue (towards the right of the foot) when viewed side on. As can be seen in Figure 1, we label the three points defining the ends and highest point of the arch.

2. Further examples

Synthetic dataset. In Figure 2, we show additional samples of our synthetic dataset.

In-the-wild performance. In Figures 3 and 4, we show further qualitative predictions on in-the-wild images, of our keypoint and normal predictors.

Reconstruction. In Figure 5, we show further examples of our 3D reconstructions.

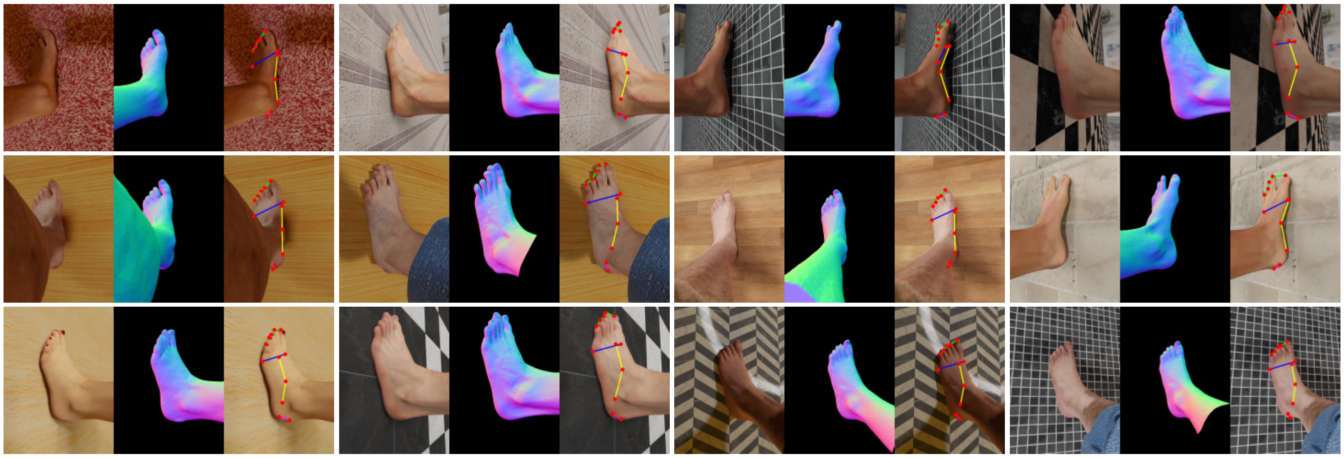

Figure 2. Further samples from our synthetic dataset. We show (a) RGB, (b) normals and (c) keypoints

Figure 3. Examples of our keypoint predictor on real, in-the-wild images.

Comparison to Commercial. We show in Figure 6 a visual comparison of the quality of our reconstruction compared to a typical PCA commercial implementation, noting improved reconstruction quality around the toes.

3. Data augmentation details

For each augmentation, we use p to denote the probability of applying it to a training image.

Downsample-and-upsample ($p = 0.5$). Bilinearly downsample the input image by ratio $r \in \mathcal{U}(0.2, 1.0)$ and upsample it back to its original resolution.

Horizontal flipping ($p = 0.5$). Horizontally flip the input image.

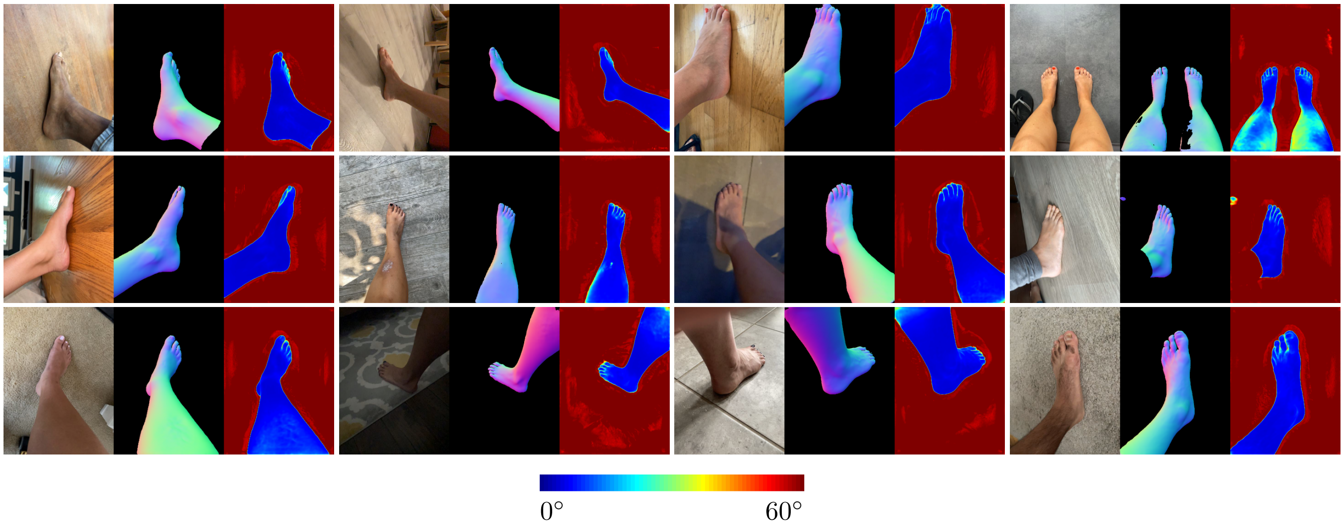

Figure 4. Examples of our normal predictor on real, in-the-wild images in a variety of lighting conditions and viewpoints. For each, we show (a) RGB, (b) normals, (c) uncertainty. Note that our network is only trained on single foot images, but can still handle the multi-foot case.

JPEG compression ($p = 0.5$)**.** Compress the input `png` image into `jpeg`, with quality $q \in \mathcal{U}(10, 90)$.

Gaussian blur ($p = 0.5$)**.** Apply Gaussian blur of kernel size $(7, 7)$ and $\sigma \in \mathcal{U}(0.1, 10.0)$.

Gaussian noise ($p = 0.5$)**.** Add Gaussian noise of $\sigma = 0.01$ (the image is pre-normalized to $[0.0, 1.0]$).

Color jitter ($p = 1.0$)**.** Apply `ColorJitter` augmentation in PyTorch with `brightness=0.5`, `contrast=0.5`, `saturation=0.5` and `hue=0.1`.

Grayscale ($p = 0.02$)**.** Change the image into grayscale.

Perspective ($p = 1.0$)**.** Rotate the camera around the yaw, pitch and roll axes, with angles $\theta_{\text{yaw}} \in \mathcal{U}(-20°, +20°)$, $\theta_{\text{pitch}} \in \mathcal{U}(-20°, +20°)$, and $\theta_{\text{roll}} \in \mathcal{U}(-180°, +180°)$.

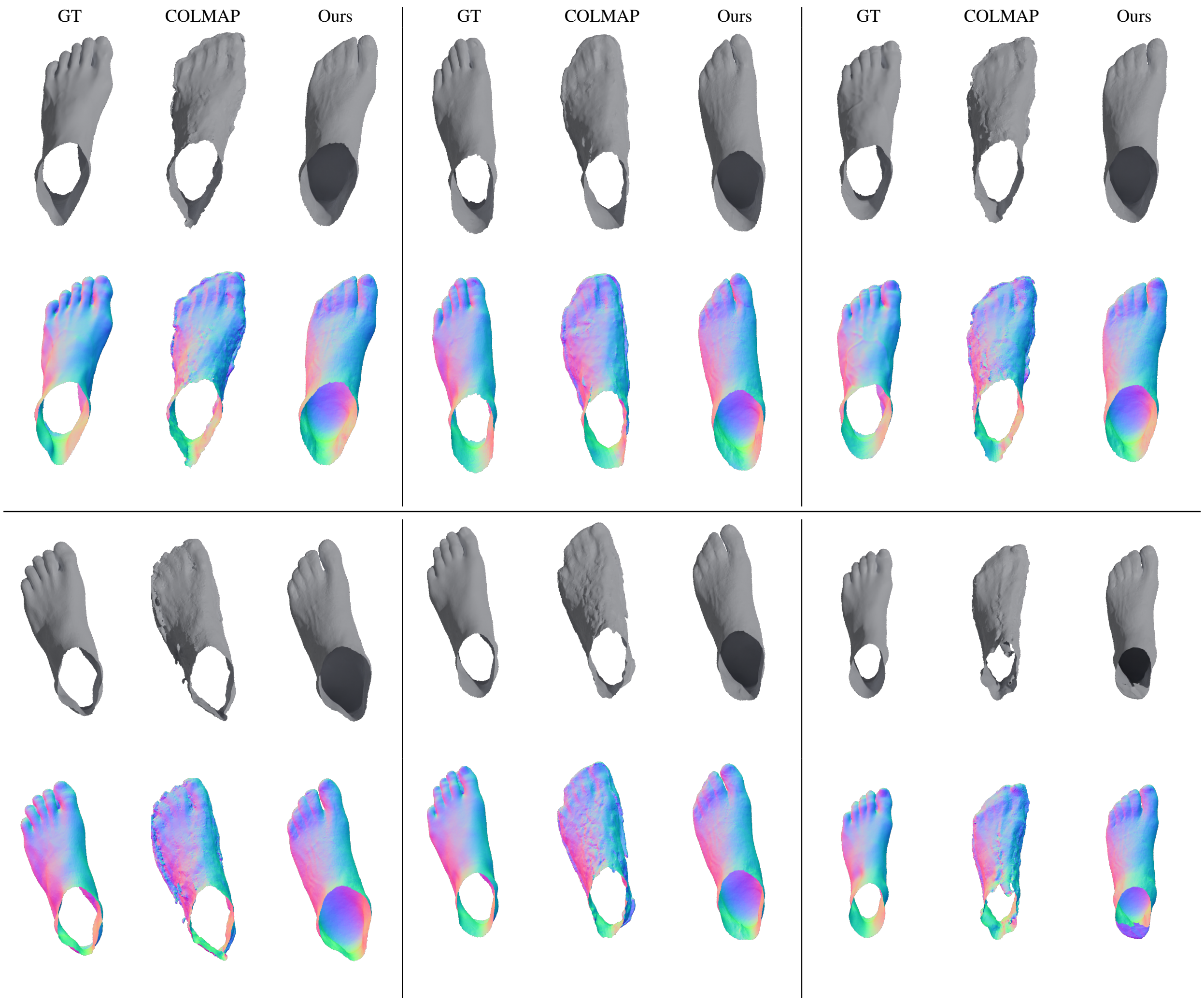

Figure 5. Further qualitative results of reconstructions from our method, compared to COLMAP.

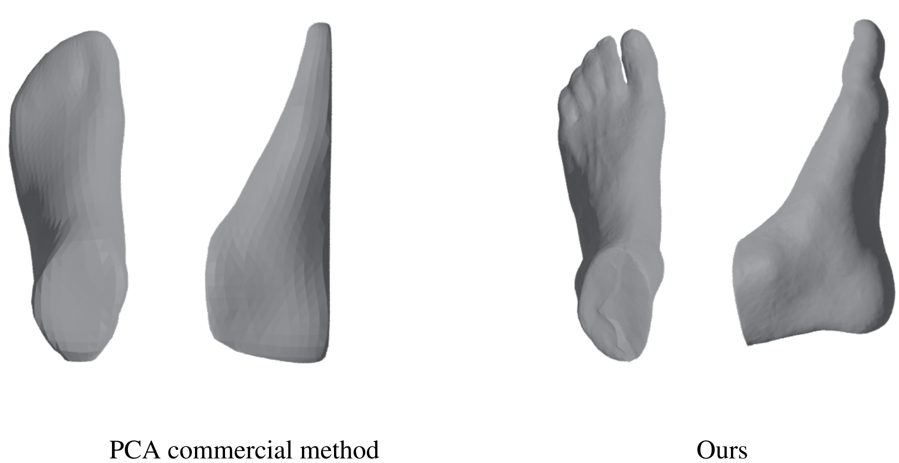

Figure 6. We compare a typical reconstruction of ours to that of a commercial, PCA based implementation.