Deep Learning Anthropomorphic 3D Point Clouds from a Single Depth Map Camera Viewpoint

Nolan Lunscher
University of Waterloo
200 University Ave W.
nlunscher@uwaterloo.ca

John Zelek
University of Waterloo
200 University Ave W.
jzelek@uwaterloo.ca

Abstract

In footwear, fit is highly dependent on foot shape, which is not fully captured by shoe size. Scanners can be used to acquire better sizing information and allow for more personalized footwear matching, however when scanning an object, many images are usually needed for reconstruction. Semantics such as knowing the kind of object in view can be leveraged to determine the full 3D shape given only one input view. Deep learning methods have been shown to be able to reconstruct 3D shape from limited inputs in highly symmetrical objects such as furniture and vehicles. We apply a deep learning approach to the domain of foot scanning, and present a method to reconstruct a 3D point cloud from a single input depth map. Anthropomorphic body parts can be challenging due to their irregular shapes, difficulty for parameterizing and limited symmetries. We train a view synthesis based network and show that our method can produce foot scans with accuracies of 1.55 mm from a single input depth map.

1. Introduction

In the footwear market, there are countless brands and models that come in all shapes and sizes. Similarly, individual feet vary widely and pairing a person to the best-suited footwear is not obvious [11]. Finding the optimal footwear can be of importance to consumers as their fit largely determines performance and comfort. In footwear, typically the only indicator used to estimate fit is shoe size, which does not fully characterize the profile of a shoe or a foot [10]. Foot morphology for describing foot shape can be complex and include measures for various lengths, widths, girths and angles [9]. This makes it difficult to determine how some footwear will fit without trying them on. This can be especially inconvenient with the rise of online shopping, where items cannot be tried-on before purchase. This process could be improved if the precise 3D shape of a foot could be virtually fitted with the 3D volumetric shape of a shoe cavity.

The first step of this process includes determining the shape of a persons feet beyond what is captured by a simple shoe size measurement. 3D scanning presents a great solution to this problem, as it can provide an accurate model of a person's foot. Such systems have already started to hit the retail space, including the Vorum Yeti[1] and the Volumental scanner[2], however they tend to be expensive or cumbersome to operate.

In developing a cheap and simple 3D scanner, RGBD cameras are compelling as they are affordable with sufficient accuracy and easy to operate. These cameras use an infrared projector and camera pair that can measure the depth of a surface using a structured light or a time of flight system. 3D scanning using RGBD cameras present a number of challenges. A RGBD camera can only capture points from a single viewpoint at a time, and obtaining a full 3D scan of an object requires either a moving camera or multiple fixed cameras. When scanning a person, stationary cameras are a better solution, as they are not as mechanically complex but also faster and leave less opportunity for a person to move during scanning. When using RGBD cameras however, interference between multiple camera projector patterns prevents them from being able to capture simultaneous images, which can still allow for some movement by the person [14]. In order to form a 3D scan, the data from each view point needs to be registered to produce a properly aligned scan. This process typically works best when there is significant overlap between views, thus requiring many views that are close together. When using fixed cameras, more required views results in more required camera hardware (e.g. 8 cameras are needed to have one at each 45 degree interval around an object).

Many of the described complications in building a 3D scanner have to do with the need to capture every aspect

[1] vorum.com/footwear/yeti-3d-foot-scanner

[2] volumental.com

of an object in order to reconstruct a model. Humans do not tend to be limited in this same way. We have the ability to form a complete mental model of an object from far less information. This has been shown in experiments that demonstrate our ability to perform mental rotation on 3D objects [22]. We seem to be able to leverage prior information and semantics about objects to more efficiently create models and to fill in missing information. Brain inspired neural networks and deep learning approaches have been shown to perform well in a multitude of vision related tasks [13] by leveraging abstract representations to implicitly learn and classify models from large databases. Deep learning methods have been shown to be able to model 3D shape from limited inputs through voxel volume representations [5, 6, 20, 23, 25, 27] and with view synthesis [2, 18, 24, 29, 30] in objects such as furniture and vehicles.

We leverage the architecture of Tatarchenko et al. [24] and apply it to infer the full 3D shape of anthropomorphic body parts using a single input view. We used data from the MPII Human Shape [19] meshes of the CAESAR database [21], and focus on how this method can be applied to scanning a persons feet, such that it may be used in determining a personalized recommendation for the best fitting footwear. Our network learns to extract the necessary information, including knowledge about left vs. right foot, and various foot structures and their measures from a partial representation. This information is used to produce a full 3D reconstruction of the overall foot.

As far as we are aware, we are the first to apply deep learning to implicitly learn 3D shape of anthropomorphic body parts. Body parts can be particularly challenging compared with other objects studied in literature (i.e. cars, chairs, planes, etc...) due to their irregular shapes and limited symmetries. Secondly, we are the first to apply deep learning to facilitate more efficient 3D scanning of anthropomorphic body parts; i.e., we learn other 3D viewpoints from a single viewpoint and thus are subsequently able to synthesize the entire 3D foot from a single input viewpoint.

2. Previous Works

Due to the abilities of RGBD cameras to quickly deliver depth maps, while being available at a low cost, they have become very popular in recent years for many applications, including 3D scanning. One widely used method is the Kinect Fusion algorithm [17], which reconstructs a scene from a moving RGBD cameras video. This system captures many frames from a scene and is able to produce high quality scans of objects, however the process can take a long time to complete as the camera must be moved through all necessary viewpoints. Multiple RGBD cameras can be used to provide faster scans [4, 14], however a large camera apparatus is then needed, as well as complex calibration and registration techniques to produce final scans, making these systems less ideal for a real world use.

An alternative approach to capturing a complete scan of an object is to capture only sufficient information to estimate the parameters of the completed object, and to use this to deform a template object to match these estimates. It has been shown that the parameters of foot shape can be compacted using statistical models while still containing sufficient information to reconstruct the overall shape from as few as 4 input measurements [15] or from foot outlines [16]. Similarly, a number of methods have been explored to create parameterized models of whole bodies [3, 19, 31], by fitting a reduced set of vertices to a set of complete body scans. These parameterized models can be leveraged in learning methods to produce 3D body models from images, by determining a mapping from an image or images to a set of parameters used to deform a template model [7, 8]. The main drawback to these methods is that they are dependent on a complete set of predefined parameters to characterize the object being scanned. Often the measurement of these parameters requires some skill and patience as well as the localization of the necessary datum points for the parameters can sometimes be very difficult. In other words these methods cannot learn shape directly from a set of arbitrary scans or shape models.

The idea of estimating the completed object shape from limited inputs has also been considered as a shape completion problem. In deep learning, the typical approach involves representing a shape using a voxel volume representation, which can be operated on using 3D convolutional neural networks. These networks have been explored to work directly on limited voxel inputs [6, 20, 25], or from limited input images [5, 20, 23, 27], to form a completed 3D shape voxel representation. These methods have been shown to perform well in shape completion tasks, however their usefulness in 3D measurement is far more limited. Voxel representations are computationally intensive in deep learning, limiting the output resolutions used in current implementations (usually 32x32x32 voxels or less).

Another approach to acquiring missing object information is to treat it as a view synthesis problem. The goal of view synthesis is to synthesize a novel view of an object or scene given a single or a set of arbitrary views. Using this idea, missing views of the object can be synthetically scanned to complete the object. The benefits of this approach are that no explicit object parameters are required, making it more generally applicable, and in deep learning, images can contain higher resolution information than voxels for similar computational complexity. In order to maximize the visual appearance of a synthesized view, a number of methods have been explored, including: appearance flow [18, 30], where the output view samples pixels from the input view, and adversarial networks [29], where the output view attempts to fool a discriminator network. These approaches focus on RGB views however, making it difficult to utilize these methods to extract 3D object shape. Tatarchenko et al. [24] demonstrated how view synthesis can be used to create useful 3D object representations. Their approach uses a convolutional neural network that takes an RGB view as input and produces a RGBD view as output. This network learned to produce the color and depth values of objects from scratch, which resulted in less accurate reconstructions than those in [18, 29, 30] but its inclusion of the depth component allowed for the reconstruction of an object from a single image. This concept of using depth map views to represent object shape has also been explored in generative models [2].

Our approach is similar to [24], however we focus on the 3D information in the depth map as input as opposed to RGB values, and apply it to foot scanning with a single depth map camera. We propose a method that uses a deep neural network to take as input the shape information from a depth map and synthesizes novel shape information through an output depth map. This shape to shape approach should contain more information about the 3D structure of a foot than the image to image approach seen in other view synthesis approaches, and thus be more practical in object scanning. Additionally our model is not told explicitly how to represent shape, allowing it to be trained directly from a set of non-parameterized arbitrary scans or shape models.

3. Proposed Method

We frame the problem of extrapolating a limited foot scan into an entire point cloud as a view synthesis problem. We leverage the power of deep learning to implicitly learn foot shape, and relationships between how it can appear from view to view.

Our depth map view configuration is shown in Figure 1. A foot is placed at the origin, and depth map images can be taken from camera poses at various azimuth and elevation angles, as well as at varying radii. We also allow for variations that reflect real world imperfect camera mounting, where its orientation can have additional roll, pitch angle offset and heading angle offset, rather than always looking directly at the foot object. In training, input view of the foot are randomly distributed in azimuth angle, elevation angle and radius, while also randomly having some degree of roll, pitch angle offset and heading angle offset. Our method then estimates any desired depth map view from another camera pose of the same foot object at an arbitrary azimuth angle and elevation angle. We train our network to produce output views that do not contain any imperfect framing from roll or offset.

3.1. Network Architecture

Our network architecture is shown in Figure 2, and is similar to [24]. Our network takes in as input an arbitrary depth map view of an object denoted x, and an encoding specifying the desired view camera pose denoted θ. From this input, the network provides an estimate for the depth map at the desired pose denoted y. We train a convolutional neural network encoder and decoder, where the encoder encodes the input depth map x into a vector representation. The desired view camera pose θ is encoded by a set of fully connected layers before being appended to the xs vector representation. The decoder processes this new vector representation, and uses deconvolutions [28] to synthesize an output depth map of the same size as the input depth map.

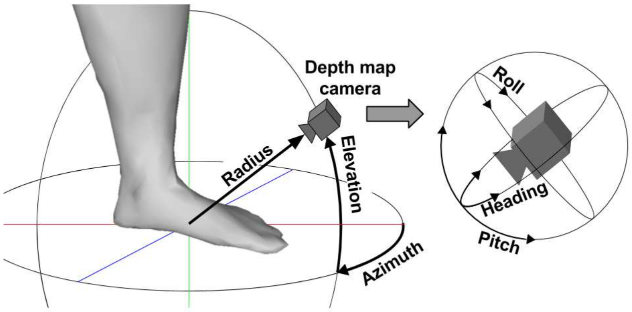

Figure 1. Depth camera pose configuration.

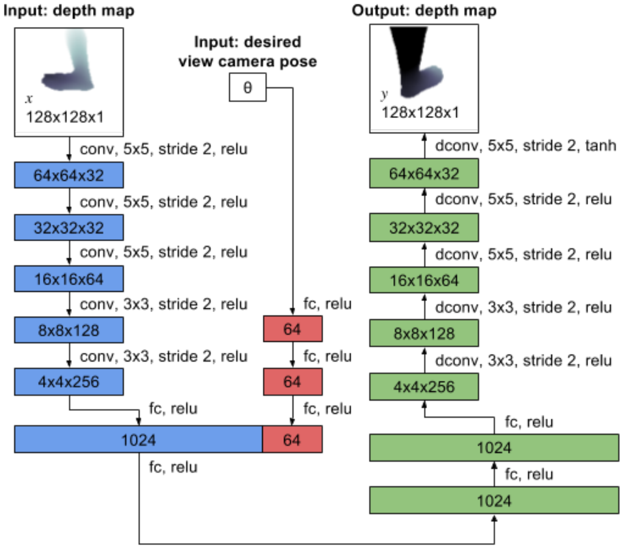
Input: depth map
x
128x128x1
conv, 5x5, stride 2, relu
64x64x32
conv, 5x5, stride 2, relu
32x32x32
conv, 5x5, stride 2, relu
16x16x64
conv, 3x3, stride 2, relu
8x8x128
conv, 3x3, stride 2, relu
4x4x256
fc, relu
1024
fc, relu
Input: desired view camera pose
θ
fc, relu
64
fc, relu
64
fc, relu
64
Output: depth map
y
128x128x1
dconv, 5x5, stride 2, tanh
64x64x32
dconv, 5x5, stride 2, relu
32x32x32
dconv, 5x5, stride 2, relu
16x16x64
dconv, 3x3, stride 2, relu
8x8x128
dconv, 3x3, stride 2, relu
4x4x256
fc, relu
1024
fc, relu
1024

Figure 2. Network architecture. Blue: depth map encoder, Red: desired view camera pose encoder, Green: depth map synthesis decoder.

3.2. Complete Point Cloud Reconstruction

In order to completely reconstruct the foot object from a single view, k forward passes through our network are required. For any input image x, a set of desired view camera poses $\theta_{1,2,...,k}$ must be specified that would sufficiently cover the object being scanned. Our network is then used to estimate the scans $y_{1,2,...,k}$, through k forward passes.

These k depth map estimates are then reprojected and registered using the intrinsic parameters of the camera used to create the training data and the camera poses encoded in $\theta_{1,2,...,k}$, resulting in a complete point cloud.

3.3. Dataset

In order to train our network to predict 3D foot shapes, we use the meshed models from MPII Human Shape [19]. These models are created by fitting a statistical model of 3D human shape to full body scans from the CAESAR database [21]. We use the 4301 mesh body models that were fit using the posture normalization of Wuhrer et al. [26]. Each full body model consists of 6449 vertex points, with about 800 vertex points in each foot up to the knee. While the MPII Human Shape models are technically a parameterized shape representation, we do not use these parameters anywhere in our method, to allow our network to learn its own representations for shape.

For each of the meshed body models, we isolate the points associated to the left and right feet up to the knee. We then transform each set of foot points such that the origin is the center of the second toe and the heel, and shrink down the scale to 0.003 (from mm units). This process results in 8602 meshed foot objects to train our network, samples are shown in Figure 3. Depth map images of the foot objects are rendered using Panda3D[3] before training, at a size of 128x128. The input and output views of the foot objects are randomly rendered using the parameter ranges described in Table 1.

3.4. Implementation Details

Our dataset of 8602 feet was separated by individuals such that both of a persons feet would be in the same set. 80% of the data was used for training and 20% for testing. Our network was implemented in Tensorflow [1] on a Linux machine with an Nvidia K80 GPU. Training was done with mini batch sizes of 64 using the Adam optimizer [12] and a learning rate of 5e-5. Our loss function was the mean L_1 distance between the output depth map pixels $\hat{d}_i$ and ground truth d_i, shown in the following equation:

$$\mathcal{L} = \sum_i \|d_i - \hat{d}_i\|_1. \tag{1}$$

When reconstructing the entire point cloud of each foot object we use a set of $k = 24$ desired viewpoint scans. We run our network to estimate scans positioned at a radius of 2, every 45 degrees in azimuth angle for elevation angles of -30, 0 and 30 degrees. We further use 3D cropping and MATLABs *pcdenoise* function to remove outliers and clean our final representation.

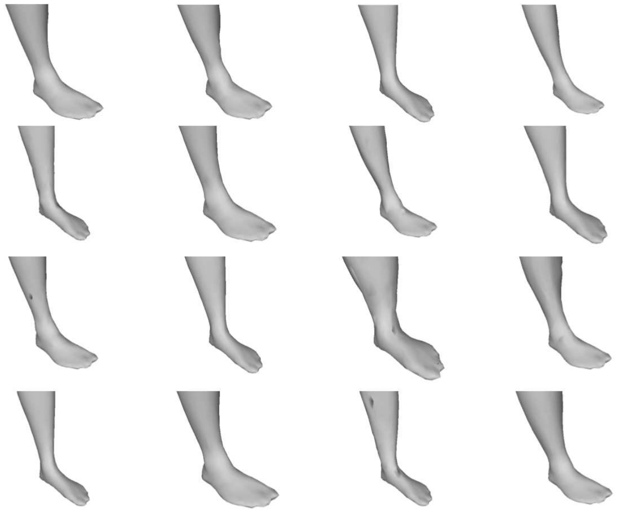

Figure 3. Meshed foot objects from MPII Human Shape [19].

Parameter name	Input	Output
Azimuth	0, 5, . . . , 355	0, 20, . . . , 340
Elevation	-30, -25, . . . , 40	-30, -20, . . . , 40
Roll	-5, -4, . . . , 5	0
Radius	1.9, 1.95, . . . , 2.1	2
Pitch offset	-2, -1.5, . . . , 2	0
Heading offset	-2, -1.5, . . . , 2	0

Table 1. Rendering camera pose parameter ranges for the network input and output (angles are in degrees).

4. Results

Our test set consists of 1720 random foot objects not used during training. The accuracy of our outputs is evaluated in two ways. First, we evaluate the networks ability to generate novel depth map views given an arbitrary input depth map view. Second, we evaluate our methods ability to fully reconstruct a foot point cloud given a single input depth map view. In both cases we compare our method with the performance of the mean model supplied as part of the dataset [19]. The mean model was made using the mean of all parameters in the statistical human shape model, across all scanned bodies in the dataset. With this mean foot method, given an input depth map view and a desired view camera pose, we simply return a depth map of the mean foot from the inputted desired view camera pose. We return a scan of either the left or right mean foot based on what object is given as input. We make this comparison to ensure our network is not learning the local minimum solution of only learning the mean foot shape, rather than individual shapes.

[3] panda3d.org

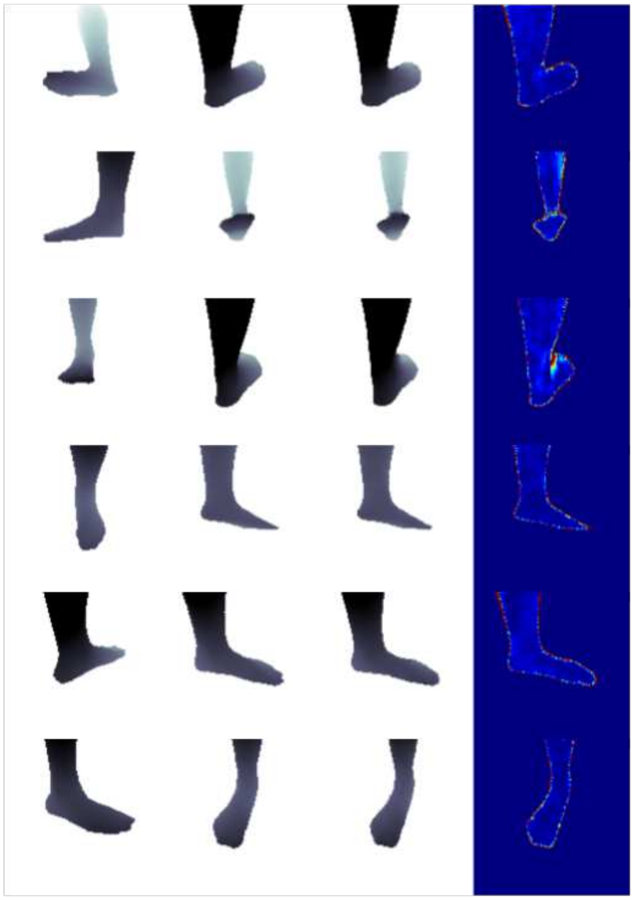

Figure 4. Depth map estimates (best viewed in color). First column: input depth map, Second column: ground truth, Third column: output depth map estimate, Fourth column: output depth map error.

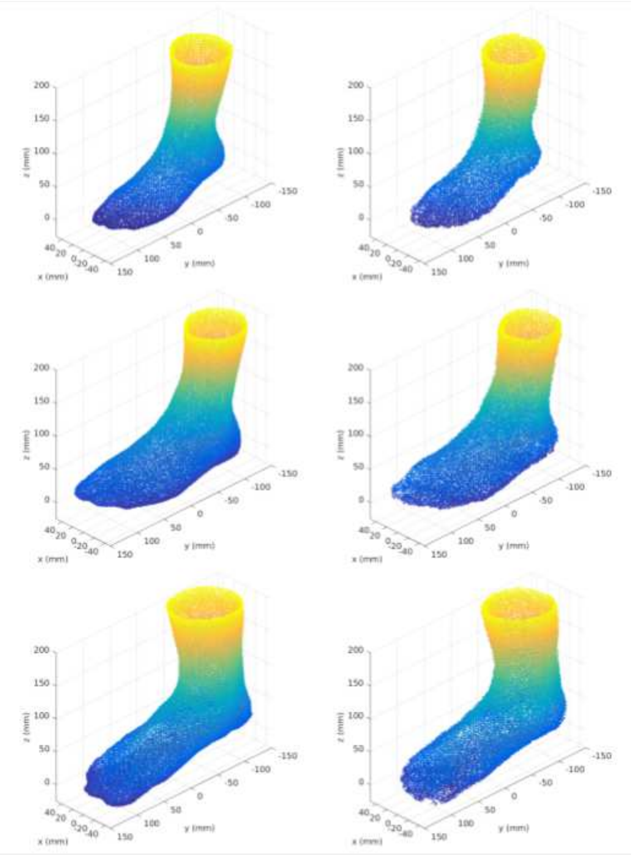

Figure 5. Estimated point clouds generated from a single input depth map using our method. First column: ground truth point cloud, Second column: estimated point cloud.

Metric Name	Ours	Mean Foot
Depthmap Error	0.0072	0.0192
Point Cloud Error (mm)	1.5455	5.9193
Point Cloud Error Std. (mm)	0.4077	2.2442

Table 2. Error across the test sets. Abbreviations used: Standard Deviation (Std).

4.1. Depth Map Results

In order to evaluate our networks ability to generate novel depth map views, we use 64 random input-output pairs for each foot in the test set. Our error measure is the mean L_1 output depth map pixel difference with the ground truth. Our results are shown in Table 2. Samples of the depth maps from the network are shown in Figure 4, as well as the distribution of the error across the depth maps. It can be seen that the majority of the error comes from the pixels around the outline of the foot. It appears that in these regions the network is uncertain whether these pixels should belong to the foot or the background. These high error outline pixels do not pose much of a problem when forming the foot point cloud however, as they are easily filtered out in our post processing step.

4.2. Point Cloud Results

In order to evaluate our networks ability to reconstruct the complete foot point cloud, we used a single input depth map, and generate a point cloud from 24 estimated depth maps, as described previously. We test using 24 different camera poses as the single input viewpoint, to explore what camera placement works best for foot scanning. Our point cloud error is calculated by comparing the estimated point cloud against the point cloud formed by the ground truths for the same 24 depth maps. Additionally, we compare our error results with that of the mean foot point cloud, formed using the same 24 depth maps.

Our error measure is similar to that used by Luximon et al. [15], who used a statistical model to parameterize a foot point cloud based on 4 measures. We use a two directional nearest neighbor euclidean distance metric to measure simi-

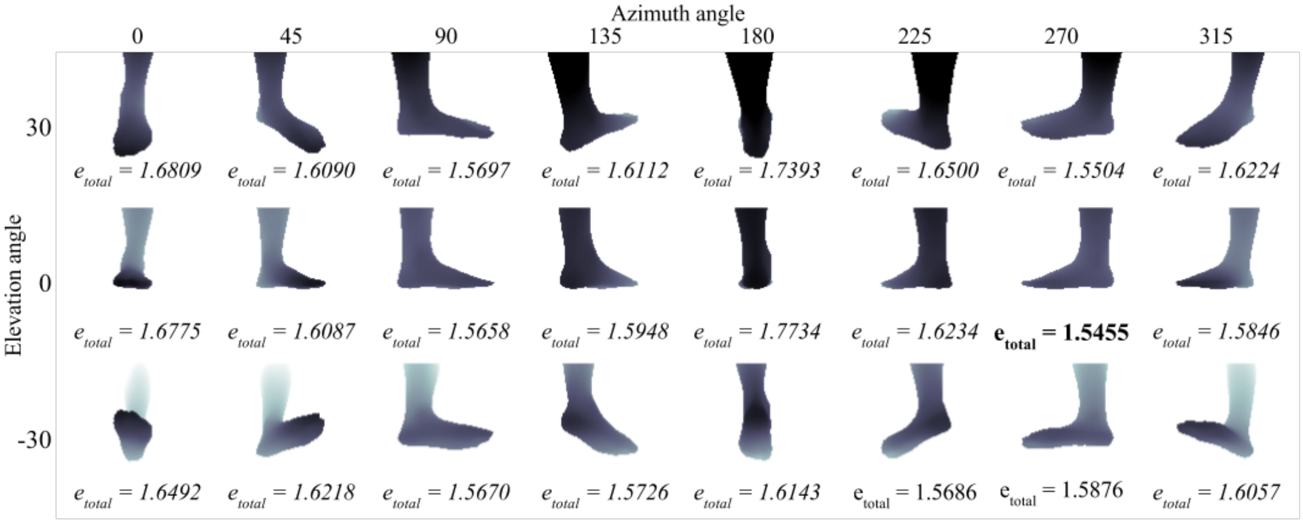

Figure 6. The 24 input depth map views explored and their corresponding estimated point cloud error on the test set.

larity between point clouds. For each point $\mathbf{p}_{est,i}$ in the estimated point cloud, we calculate its euclidean distance to the nearest point in the ground truth point cloud, and for each point $\mathbf{p}_{gt,j}$ in the ground truth point cloud, we calculate its euclidean distance to the nearest point in the estimated point cloud using the following equations:

$$e_{est,i} = min_j \|\mathbf{p}_{est,i} - \mathbf{p}_{gt,j}\|_2, \tag{2}$$

$$e_{gt,j} = min_i \|\mathbf{p}_{gt,j} - \mathbf{p}_{est,i}\|_2, \tag{3}$$

where $e_{est,i}$ is the error for point $\mathbf{p}_{est,i}$ in the estimated point cloud to the ground truth, and $e_{gt,j}$ is the error for point $\mathbf{p}_{gt,j}$ in the ground truth point cloud to the estimate. We average the distance measures by the number of points in each cloud, then average the two measures to form our overall error of our point cloud estimate using the following equation:

$$e_{total} = \frac{\frac{1}{N}\sum_i e_{est,i} + \frac{1}{M}\sum_j e_{gt,j}}{2}, \tag{4}$$

where N and M are the number of points in the estimated and ground truth point clouds respectively, and e_{total} is the total error reported for an estimate point cloud. We report our error measurements in mm units.

Our point cloud accuracy results are shown in Table 2, and sample point clouds are shown in Figure 5. A breakdown of the point cloud error for estimates generated from each of the 24 input views explored is shown in Figure 6. Our model performs with average errors of less than 1.55 mm from the best input view.

Looking more closely at Figure 6, it can be seen that point clouds generated when given an input view of the bottom of the foot were generally more accurate than when given a view of the top of the foot. This suggests that the most important information about foot shape is contained on the bottom surface of the foot rather than the top. Interestingly however, the most accurate point clouds are formed when given a profile view of the foot, with 0 elevation angle. This result suggests that when building a scanner with a single camera, the camera should be mounted at this profile view position. This finding is also inline with those from Luximon et. al. [16], who found that having information from the foot profile was important for overall shape reconstructions.

5. Discussions and Conclusions

We have presented a method for leveraging deep learning to allow for more efficient 3D object scanning from a single input view in the application of foot scanning. Our network successfully learned the 3D shape an anthropomorphic body part from incomplete information and was shown to be capable of accurately generating a complete point cloud representation of a foot from a single depth map. Our method was able to reconstruct full point clouds with an accuracy of 1.55±0.41 mm, which is significantly smaller than the English and American shoe sizing half size increment of 4.23 mm [9].

This method has a number of benefits over more traditional methods of RGBD scanning. Our method requires only a single input viewpoint, which allows the scanner to be significantly cheaper and simpler to operate, and allows us to avoid complications associated with multi-camera setups. Without the need for multiple cameras, the scanning process can be near instantaneous without giving a person any opportunity to move during a scan. This also has potential applications for capturing the foot loading and dynamics in a 3D video. This method of scanning also does not require any special calibration or registration between views due to how our network implicitly accounts for input camera pose.

Despite our promising results, our method has some limitations over more traditional RGBD scanning methods. The point cloud produced by this method is not as accurate as a scan using multiple viewpoints, which would contain the true information about the object's shape from all views. Our method is also limited to only scanning new instances of objects that are mostly similar to those seen during training. This method will fail if for example someone for whatever reason has a particularly unique foot shape, or if there is any sort of abnormality on a part of the foot not seen by the cameras single viewpoint. For these reasons, our method may not be practical to completely replace more traditional scanning methods, however in our application, for most feet it is sufficient to capture an accurate representation.

Our future works are aligned with tackling the limitations of our method. We plan to investigate changes in network architecture to improve the accuracies of the produced scans, as well as methods of preprocessing and postprocessing the data. We also plan to investigate the use of color cameras in single view scanning, which can have resolutions much higher than typical RGBD depthmaps.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.

[4] Y. Chen, G. Dang, Z.-Q. Cheng, and K. Xu. Fast capture of personalized avatar using two kinects. *Journal of Manufacturing Systems*, 33(1):233–240, 2014.

[5] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016.

[6] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *arXiv preprint arXiv:1612.00101*, 2016.

[7] E. Dibra, H. Jain, C. Öztireli, R. Ziegler, and M. Gross. Hs-nets: Estimating human body shape from silhouettes with convolutional neural networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 108–117. IEEE, 2016.

[8] E. Dibra, C. Öztireli, R. Ziegler, and M. Gross. Shape from selfies: Human body shape estimation using cca regression forests. In *European Conference on Computer Vision*, pages 88–104. Springer, 2016.

[9] R. S. Goonetilleke. *The science of footwear*. CRC Press, 2012.

[10] M. R. Hawes and D. Sovak. Quantitative morphology of the human foot in a north american population. *Ergonomics*, 37(7):1213–1226, 1994.

[11] E. Holmes. Feet are getting bigger, and many people wear shoes that don't fit right. *The Wall Street Journal*, 2014.

[12] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations 2015*, pages 1–15, 2015.

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[14] S. Lin, Y. Chen, Y.-K. Lai, R. R. Martin, and Z.-Q. Cheng. Fast capture of textured full-body avatar with rgb-d cameras. *The Visual Computer*, 32(6-8):681–691, 2016.

[15] A. Luximon and R. S. Goonetilleke. Foot shape modeling. *Human Factors*, 46(2):304–315, 2004.

[16] A. Luximon, R. S. Goonetilleke, and M. Zhang. 3d foot shape generation from 2d information. *Ergonomics*, 48(6):625–641, 2005.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[18] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. *arXiv preprint arXiv:1703.02921*, 2017.

[19] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3d human modeling. *CoRR*, abs/1503.05860, 2015.

[20] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems*, pages 4996–5004, 2016.

[21] K. M. Robinette, H. Daanen, and E. Paquet. The caesar project: a 3-d surface anthropometry survey. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 380–386. IEEE, 1999.

[22] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[23] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single

depth image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[26] S. Wuhrer, C. Shu, and P. Xi. Posture-invariant statistical shape analysis using laplace operator. *Computers & Graphics*, 36(5):410–416, 2012.

[27] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[28] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[29] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, and J. Feng. Multi-view image generation from a single-view. *arXiv preprint arXiv:1704.04886*, 2017.

[30] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016.

[31] S. Zuffi and M. J. Black. The stitched puppet: A graphical model of 3d human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3546, 2015.