ISSN 1870-4069

Feet Point Cloud Orientation, Localization and Semantic Segmentation

Ricardo C. Villarreal-Calva, Ponciano J. Escamilla-Ambrosio,
Juan Humberto Sossa Azuela

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{rvillarrealc2020, pescamilla, hsossa}@cic.ipn.mx

Abstract. In this work, an algorithm towards automatic feet measurements extraction for personalized footwear design is presented. The algorithm performs orientation, localization and semantic segmentation of a point cloud produced by a 3D feet scan system. The algorithm input instance is a point cloud from a 3D scan of a user standing on their feet on a flat surface. The outputs are two point clouds; each one is a segmentation of the user´s left and right foot with its own coordinate system. The algorithm implementation is done with the point cloud library Open3D. It combines linear programming and techniques from Machine Learning such as RANSAC and DBSCAN together with the point cloud processing algorithm Screened Surface Poisson Reconstruction to process the localization and segmentation of the foot. The proposed method contributes to the extraction of anthropometric measures of the feet in order to be able to build personalized footwear, which is ongoing research.

Keywords: Point cloud, semantic segmentation, localization, footwear customization.

1 Introduction

Technologies available for specialized users, with time will reach the maturity required for mass consumption. Such an actual case is the light detection and ranging (LiDAR) sensors which was popularized after the DARPA urban challenge competition on self-driving cars [1]. Today, this technology is embedded into higher end tablets and smartphones produced by the company Apple®.

The LiDAR and cameras on these devices produce an RGB-D image, which is a regular image with a distance value for each pixel. Several RGB-D images can be processed to create a 3D scan of a scene in a process known as scene registration [2]. It is expected that this technology will improve the experience of online stores as users will be able to scan its whole body or parts of it and get the proper size for every product they buy online [3].

Fig. 1. Typical input point cloud from the feet scan process. For the Multi View shape representation around 60 photographs of the feet are taken; then the photographs are processed to change from Multi View to point cloud shape representation. The icons around the feet represent the position and orientation of the camera for each photograph taken.

Unfortunately, current footwear mass production provide a proper fit to 90% of customers with 3 shoe widths per length [4], so there exists an important percentage of the population that uses wrong sized shoes, which can produce foot deformities and decrease life quality over time. The present work proposes a method to do semantic segmentation of the feet scan of a person in standing position.

The method produces a point cloud and a coordinate system for each foot, so it will be possible to further process it to extract anthropometric measures such as the ones described in [5] to design personalized footwear. The outline of this work is as follow: in section two, the input instance expected for the algorithm is described. Section three presents the floor segmentation and initial alignment of the Z axis. Section four describes the foot clustering and removal of foot noise. Finally, in section five the creation of the new coordinate system for each foot is introduced.

2 Data

The algorithm requires as input a point cloud X, such that each point $x \in X$ belongs to the Euclidean space R^3. Even tough the point cloud might have color information for each point; this data is deliberately ignored, as it would require us to consider aspects like different skin tones and floor color for the segmentation.

The point clouds used for this study are obtained from Multi View 3D object representation [6], which consists of around 60 photographs of the subject to scan. Each photograph is processed through the photogrammetry software COLMAP [7, 8] to transform them from Multi View to Point Cloud shape representation.

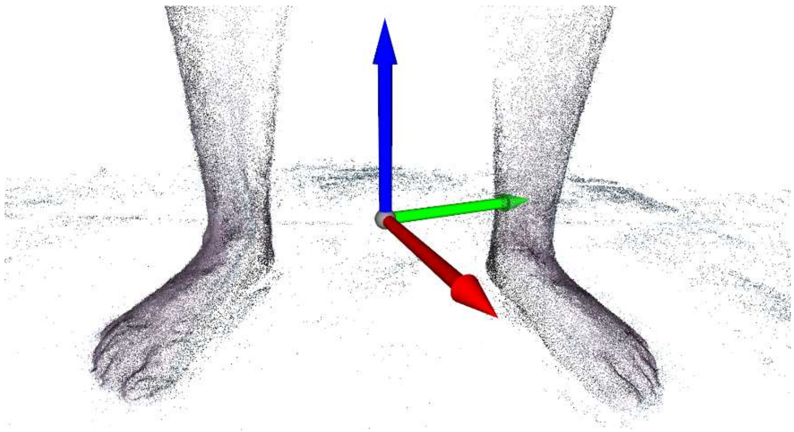

Fig. 2. Point cloud after removal of all points identified to belong to the floor's plane and alignment of the floor to the Z-axis with its direction pointing upward from the floor. The red arrow represents the X-axis, the green arrow the Y-axis and the blue arrow the Z-axis.

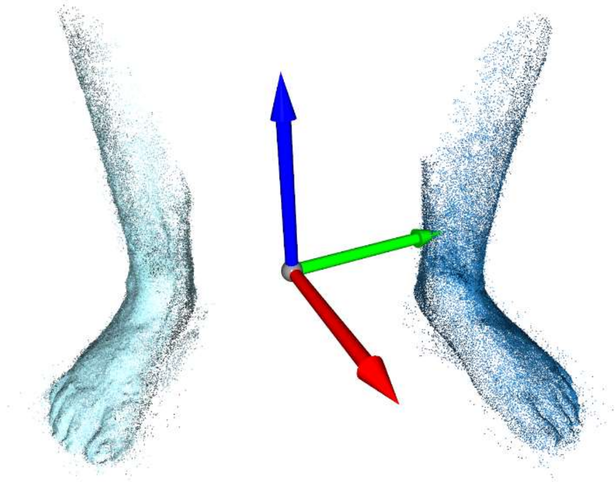

Fig. 3. Points of the two largest clusters found by DBSCAN (without points from the noise cluster).

This method assigns an arbitrary coordinate system to the point cloud, which is dimensionless; the cardinality of the point cloud is in the order of hundreds of thousands of points. The 3D feet scan must comply with the following characteristics:

- The person at time of the 3D scan must be standing on a flat surface (knee above heel).
- The scan must cover at least 1/2 of the knee height and up to half meter.
- The scan must have at least one square meter of floor.
- The scan must not have points below the floor.

– Feet must be free of strong amputations.

– Foot must have a separation of at least 15 cm between them.

3 Floor Segmentation and Z Axis Alignment

The first step towards foot localization is to identify the points that belong to the floor. Those points lie over a plane, which needs to be characterized to align it to the Z-axis. We also need to ensure that the positive Z-axis direction point upward from the floor.

3.1 Floor Segmentation

From Figure 1 we can see that we could do the floor segmentation based on color data. We could choose blue points and classify them as floor, but such strategy would fail with other floor colors. We could also do the segmentation based on skin color, but this would have the same weakness as people around the world have different skin tones.

As we want to create a robust algorithm for the segmentation, the strategy is to identify every point that belongs to the floor through the use of the Random Sample Consensus (RANSAC) [9] plane segmentation algorithm implemented on the Open3D library [10].

RANSAC takes three aleatory points from the point cloud to define a plane, then counts the inlier points (those closer than a threshold to the plane) and repeats this process several times. The set of points from the plane with more inlier points is used to find the plane such that the summed squared distances from the plane to all points is minimized.

The fitted plane is returned as (a, b, c, d) such that for each inlier point (x, y, z) we have $(ax + by + cz + d \approx 0)$. The algorithm also returns a list of indices of the inlier points. RANSAC algorithm requires three arguments: the maximum distance (δ) a point can have to an estimated plane to be considered an inlier, the number of points (n_p) that are randomly sampled to estimate a plane, and the number of times (n_t) a random plane is sampled and verified.

A worst case scenario is used to quantify the probability that RANSAC will find the floor plane; we define a box with the biggest foot dimensions found in the Mondopoint scale [11] and the maximum height of the scan. The box surface but the face touching the floor represents the area of the foot. To compute the area of the feet A_{feet} we have:

$$A_{\text{feet}} = 2(2lh + 2wh + lw), \tag{1}$$

where $l = 32\ cm$, $w = 12.2\ cm$ and $h = 50\ cm$, so $A_{\text{feet}} = 9{,}620\ cm^2$, we have the minimum floor area which is $A_{\text{floor}} = 10{,}000\ cm^2$. We define a Binomial Random Variable to calculate the probability P_f that RANSAC will choose three points from the floor to define a plane:

$$P_f = \binom{n}{k} p^k (1-p)^{n-k}, \tag{2}$$

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}, \tag{3}$$

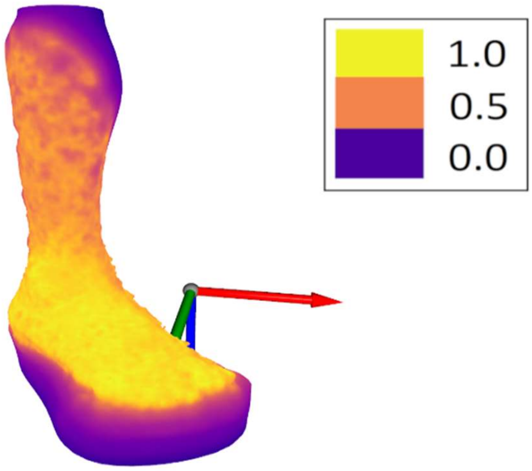

Fig. 4. Mesh for one foot displayed with a shader where violet indicates low point density and yellow high point density.

where p is the probability of choosing a point from the floor and is calculated by:

$$p = \frac{A_{\text{floor}}}{A_{\text{floor}} + A_{\text{feet}}}. \tag{4}$$

And we want to calculate the probability of three successes in three trials $(n = 3,\ k = 3)$. Performing computation with the previous parameters give $p = 0.51$ and $P_f = 0.132$. As RANSAC repeats this process, by The Central Limit Theorem we can approximate the distribution of planes found from floor points with a normal density function with parameters $u = np$ and $\sigma = \sqrt{npq}$ where $n = n_t,\ p = P_f$ and $q = 1 - P_f$.

For this work $n_t = 1{,}000$, therefore, we have that RANSAC will find a mean $u = 132$ of different planes from points of the floor with a standard deviation of $\sigma = 10.7$. Therefore, RANSAC with a probability higher than 99.7% will chose the best floor plane from more than 100 plane samples from the floor.

3.2 Z Axis Alignment

The equation of the floor plane is used to rotate the point cloud such that its normal vector is parallel to the Z-axis; then a translation transformation is applied to the point cloud such that the center of the floor´s inlier points is translated to the origin of the coordinate system.

The center of the point cloud must be between the floor and the knee of the user, so if the Z coordinate of the point cloud center has a negative value, then the point cloud is rotated 180° around the X-axis; otherwise, it is left in its current configuration.

Finally, all floor's inlier points are removed from the point cloud. Figure 2 shows the point cloud once this process ends with the coordinate system´s Z-axis pointing upward from the floor and the XY plane aligned to the floor plane.

4 Foot Clustering

From Figure 2 we can see that not all floor points are removed, therefore, further processing is required. The algorithm chosen is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) implemented in Open3D. DBSCAN is a recursive algorithm that starts with an arbitrary point and counts the number of points at a distance lower than a threshold (eps), if the quantity is more than the minimum of points required to form a cluster, then a cluster is created; then close clusters are merged.

The algorithm returns for each point either its cluster label or that it is classified as noise, in Figure 3 we can observe the two largest clusters found. DBSCAN requires two parameters: eps, which defines the maximum distance between neighbors in a cluster and min points, which defines the minimum number of inlier points required to form a cluster.

4.1 Foot Noise Removal

To filter the noise around both foots, the Screened Poisson Surface Reconstruction (SPSR) is used. SPSR produces watertight surfaces from oriented point sets by (1) transforming the oriented point samples into a continuous vector field in 3D, (2) finding a scalar function whose gradients best match the vector field, and (3) extracting the appropriate isosurface [12]. The surface has different densities as the algorithm will also create triangles in areas of low point density, and even extrapolates into some areas without points.

The input point cloud to SPSR must have normal vectors, Figure 4 shows the resulting mesh with a shader where normalized point densities of the mesh are displayed with pseudo color, where yellow indicates high point density, orange midpoint density, and violet low point density.

The area under the foot clearly does not have supporting points. SPSR requires parameter depth, with high values the final mesh will have more details, lower values produce smoother surfaces. In Figure 5 we can observe the mesh without low density triangles inside an Axis Aligned Bounding Box (AABB) which serves as guide to find the proper coordinate system in further steps.

5 Foot Alignment to the Coordinate System

From Figure 5 we can observe that every AABB edges are parallel to one of the axes of the coordinate system. To find a coordinate system for each foot, some properties of the AABB will be used:

1 An AABB encloses all vertices in the input mesh with the lowest possible volume.

2 Each AABB face normal vector is parallel to some axis of the coordinate system.

First, as explained in Section 3, we know that the Z axis is parallel to the floor's plane normal vector, and its direction is pointing upward from the floor. Now, we will find the Z-axis rotation for each foot that minimizes the length of the AABB along the X-axis using linear programming; the rotation search space is $[0, \pi]$. The minimization

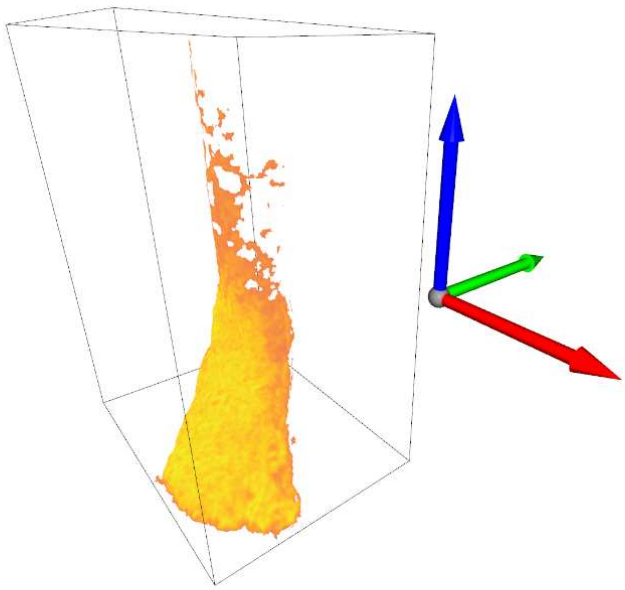

Fig. 5. Resulting mesh after removal of low-density triangles inside the AABB.

algorithm used is a modified version of [13] implemented in SciPy library [14]. The returned angles found are used to rotate each foot model around the Z-axis.

5.1 Select of the AABB Face to be Aligned to the XZ Plane

We want to position the foot such that the direction from heel to toes points towards the positive Y direction. Linear programming does not guarantee this, it just aligns the foot in the X and the Y-axes, but the solution found might be in the wrong direction, as seen in the foot on the right of Figure 6. The next algorithm is used to identify if the foot needs to be rotated to properly be aligned to the Y-axis.

1 The input instance is the point cloud cluster of one foot rotated as the linear programming output.
2 Select the two vertices of the AABB of the cluster's high-density foot mesh with lower X value and higher Z value.
3 Count the points inside a radius of length r from each vertex selected, where r is the width of the AABB.
4 Choose vertex with higher points.
5 If the vertex selected is the one with higher Y coordinate, then rotate the foot by 180° around the Z-axis.

5.2 Translation of the Foot to the Origin

At this point, we have both feet properly aligned to the Z and the Y-axes, but we still need to translate them to the origin of the coordinate system.

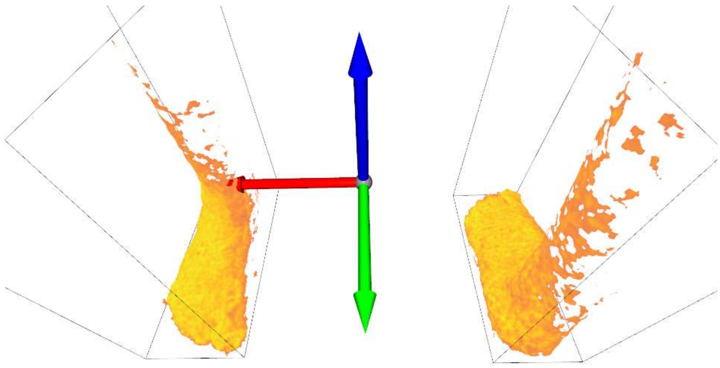

Fig. 6. Each foot mesh is rotated by angle found through linear programming. Notice that both foot lengths and widths are aligned to the Y-axis and the X-axis respectively.

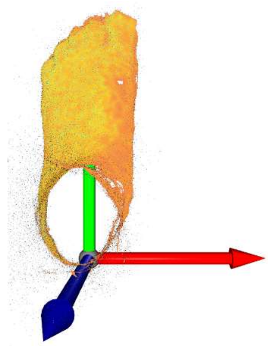

Fig. 7. Foot aligned to the coordinate system.

To do this, we use the AABB vertex with lower X, Y and Z values as anchor point of the foot and move it to the coordinate (0,0,0) and then move it half of the X width in the negative direction along the X axis. The resulting aligned foot point cloud is shown in Figure 7.

5.3 Discussion

The method presented in this work has the objective to do localization, orientation and semantic segmentation of the point cloud of the feet of a person. The evaluation at this stage is qualitative, as the photogrammetry method to recover the 3D shape produces a dimensionless point cloud; further work is required to produce a 3D shape that recovers dimension.

Future development will require creating a database of several point clouds with labels of each point to measure the precision and recall of the segmentation method. Nevertheless, the presented method is an important step towards the extraction of anthropometric measurements of a person's feet, which is a requirement to design personalized footwear.

The method to define the final precision of the anthropometric measurements is in evaluation; the two options being considered are based on Machine Learning theory and variance propagation.

The Machine Learning approach requires capturing a set of data, labeling it and defining a loss function so the error can be measured with enough confidence to state that the method generalizes properly.

The variance propagation requires to measure the error at each step of the process so an estimate of the variance of the final measurements can be calculated. As stated before, this is work in progress.

6 Conclusion

In this work, an algorithm to do point cloud segmentation and localization of foot has been presented. As first step the RANSAC algorithm to identify floor points is used and the Z axis is aligned to the plane of the floor; next with the DBSCAN algorithm most floor noise is filtered and both feet are clustered; to remove noise from the foot the SPSR algorithm is used; through linear programming the foot is aligned to the Y axis and with radius search the face of the AABB closer to the heel is detected and the correct direction in the Y axis is set. Finally, the foot shape representation is moved properly to the origin of the coordinate system.

As future work, the point cloud will be in meters so the length and width of the foot can be known, as well as other measurements required to find the proper footwear for a given user.

Acknowledgments. The authors would like to acknowledge the support provided by the Instituto Politécnico Nacional under projects 20200630 and 20210788, CONACYT under projects 65 (Fronteras de la Ciencia) and 6005 (FORDECYT-PRONACES) to carry out this research. The first author thanks CONACYT for the scholarship to undertake his master studies and to the Instituto Politécnico Nacional for the BEIFI scholarship under project 20210039.

References

1. Urmson, C., Baker, C., Dolan, J., Rybski, P., Salesky, B., Whittaker, W., Ferguson, D., Darms, M.: Autonomous Driving in Traffic: Boss and the Urban Challenge. AI Magazine, vol. 30, no. 2, pp. 17–28 (2009) doi: 10.1609/aimag.v30i2.2238
2. Dong, W., Park, J., Yang, Y., Kaess, M.: GPU Accelerated Robust Scene Reconstruction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7863–7870 (2019) doiI: 10.1109/IROS40897.2019.8967693
3. Sohn, J. M., Lee, S., Kim, D. E.: An exploratory study of fit and size issues with mass customized men's jackets using 3D body scan and virtual try-on technology. Textile Research Journal, vol. 90, no. 17-18 (2020) doi: 10.1177/004051 7520904927
4. Jurca, A., Žabkar, J. & Džeroski, S. Analysis of 1.2 million foot scans from North America, Europe and Asia. Scientific Reports, vol. 9, no. 10 (2019) doi: 10.1038/ s41598-019-55432-z
5. Luximon, A., Luximon, Y.: Shoe-last design innovation for better shoe fitting. Computers in Industry, vol. 60, no. 8, pp. 621–628 (2009) doi: 10.1016/j.compind. 2009.05.015

6. Gezawa. A. S, Zhang, Y., Wang, Q. C, Lei, Y. Q.: A review on deep learning approaches for 3D data representations in retrieval and classifications. IEEE Access, vol. 8, pp. 57566–57593 (2020) doi: 10.1109/access.2020.2982196
7. Schönberger, J. L., Zheng, E., Frahm, J. M., Pollefeys, M.: Pixelwise View Selection for Unstructured Multi-View Stereo. Computer Vision Lecture Notes in Computer Science, vol. 9907 (2016) doi: 10.1007/978-3-319-46487-9_31
8. Schönberger, J. L., Frahm, J. M.: Structure-from-Motion Revisited. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
9. Fischler, M. A, Bolles, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol. 24, no. 6, pp. 381–395 (1981) doi: 10.11 45/358669.358692
10. Zhou, Q. Y, Park, J., Koltun, V.: Open3D: A Modern Library for 3D Data Processing (2018) doi: 10.48550/arXiv.1801.09847
11. Footwear - Sizing - Conversion of sizing systems. ISO/TS 19407 2015(E). Geneva, CH: International Organization for Standardization (2015)
12. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics, vol. 32, no. 3, pp. 1–13 (2013) doi: 10.1145/2487228.2 487237
13. Powell, M. J.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal, vol. 7, no. 2, pp.155–162 (1964) doi: 10.1093/comjnl/7.2.155
14. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K. J., Mayorov N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, vol. 17, no. 3, pp. 261–272 (2020) doi: 10.1038/s41592-019-0686-2