Artificial Intelligent System for Shoe Last Personalization

Ricardo C. Villarreal-Calva, Ponciano J. Escamilla-Ambrosio, Juan H. Sossa-Azuela
Robotics and Mechatronics Lab. - Centro de Investigación en Computación
Instituto Politécnico Nacional, Mexico
{rvillarrealc2020@, pescamilla@, hsossa@}cic.ipn.mx

Abstract

The research advancements to develop a computer vision system that maps a user foot to a shoe last are presented. The system consists of a developed 3D scanner and algorithms to process the scene to extract parametric curves from the foot's surface. The shoe last design requirements are encoded in the parametric curves using a process of optimization subject to bounds and constraints, from those curves the surface of the shoe last will be created. This is work in progress.

1. Introduction

Technologies to 3D scan objects are becoming more accessible, at the same time, 3D acquired data from the human body allows to create personalized products [4, 6, 8]. The presented research proposes to create an automatic Artificial Intelligent system that maps a user's feet to a shoe last (mold that gives a shoe its shape). The shoe last would allow a shoemaker to manufacture shape personalized shoes, this is relevant as shoe's mass manufacture with one width per length only covers 40.1 percent of the population with a proper shoe, even with 3 widths per length there will be 13.2 percent of the population that will not find a given shoe model with a right size [2]. People without access to proper shaped shoes might use constricting footwear that can develop health issues such as bunionettes or hammer toes [5].

Similar systems to the proposed modify a base shoe last to fit the user's feet, either relying on human operators [3] or extracting features to guide the modifications [11]. The system described in this work does not require a base shoe last, instead, it encodes the design requirements in optimization constraints.

The following structure is followed to explain the research advancements: in Section 2, the 3D Scan system is explained; in Section 3, the foot localization, orientation and semantic segmentation are explained; in Section 4, the extraction of curves from the process is described. Finally, in Section 5, the conclusions and future work are discussed.

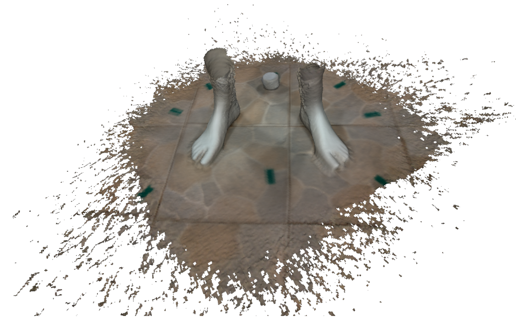


Figure 1. Example of a 3D feet scan.

2. 3D scanner

The 3D scanner used to digitalize the feet is based on the Open3D's Offline Legacy Reconstruction System (OOLRS) [14]. The original system requires as input an RGB-D sequence which is partitioned to create local geometry surfaces (fragments) $\{P_i\}$ from continuous frames, then odometry between fragments $\{T_i\}$ is computed and a global pose graph is created, a third step does global optimization of the pose graph and finally all RGB-D frames are integrated into a single 3D model.

The RGB-D sequence is captured with a handheld Intel® Realsense™ L515 depth camera with intrinsic values of 1024x768 of depth resolution and depth scale of 4000. Three modifications were done to OOLRS to improve the quality of the reconstruction: first, Visual odometry is changed to Visual-Inertial odometry; second, depth image's noise is filtered; third, a sliding window between fragments is implemented. An example of the reconstruction is shown in Figure 1.

2.1. Odometry

Open3D's legacy offline reconstruction system computes relative odometry between RGB-D images at the initial stage of the reconstruction. The initial computation of the visual odometry is replaced with the odometry captured with Intel® Realsense™ T265 tracking camera. This odometry sensor was selected for its availability and good evaluations against the baseline ORB-SLAM2 [1].

T265 captures global pose of its reference frame, but Open3D reconstruction system requires relative poses of the RGB-D sensor. To compute L515 camera's global pose an homogeneous transformation matrix SO(3) is used to transform T265 global pose to L515 global pose. With L515 global pose it is possible to compute relative poses $\{G\}$ as Open3D requires.

$$G_{s-t}^{L515} = X^{-1} \tag{1}$$

$$G_s^{L515} X = G_t^{L515} \tag{2}$$

Where G_s^{L515} and G_t^{L515} are the global poses of the source and target RGB-D frames. From Eq. 2, X is solved so it can be used in Eq. 1 to solve G_{s-t}^{L515} which is the relative pose required by OOLRS.

2.2. Depth noise filter

A filter applied to each depth image improves the quality of the OOLRS reconstruction. The filtered depth images are used to integrate the 3D scene by the reconstruction system. The original depth images are used to refine the pose graphs of the scene, as better results were observed with this setup. The filter is composed of well known image processing algorithms implemented in the following sequence with the OpenCV library: Morphological gradient (Scharr), simple image threshold, image opening, image closure. The closed image is used as mask to set to zero pixels of the original depth image, finally a bilateral filter is applied. All kernels have sizes of 3x3 and the value of the simple image threshold equals 21.

2.3. Sliding window

The first stage of the reconstruction creates fragments from a subset of continuous RGB-D frames. The original code of OOLRS do not have RGB-D frames overlap between fragments. A better reconstruction was possible by adding an overlap of 10 frames between fragments.

3. Foot segmentation and orientation

A process to segment and orient each foot is done to make the system invariant to SO(3) homogeneous transformations of the feet scan. The process used is the one described in [12], the algorithm parameters are adapted to the reconstruction quality obtained with the L515 sensor. The foot segmentation and orientation returns a point cloud for each foot where the system coordinate's origin is located at the back of the heel, X positive axis points towards the foot's width, Y positive axis towards the toes and Z positive axis towards the knee, the XY plane lies over the floor, see Figure 2a.

4. Curves approximation

With the 3D model of the foot and a known localization and orientation as priors, it is possible to extract curves of interest for the creation of the shoe last surfaces, see Figure 2c. The framework selected for this task is the B-Spline parametric curve [10]. The general process includes:

1. A plane is defined with a point and a normal vector.
2. All points within a threshold distance from the plane are selected.
3. All selected points are projected into the plane.
4. The points are sorted counter clockwise with the plane's normal vector as rotation axis.
5. An equidistant sampling of the points is created.
6. Through least square fit, a B-Spline curve approximates the equidistant sampled points.
7. The B-Spline is used as initial guess for an optimization of the position vectors B_i of the control polygon vertices subject to bounds and constraints.

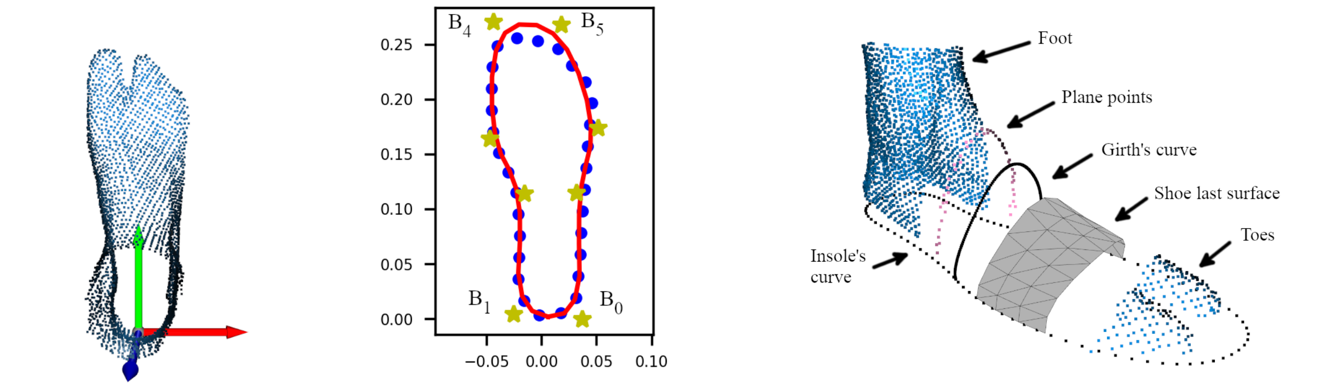
For the creation and the least square fitting of the B-Spline the Splipy library is used [7]. The optimizations with bounds and constraints are done with the library SciPy [13]. The optimizations allow to encode the design requirements in the B-Spline to create the shoe last's surfaces. The optimization problem is explained for two curves: the insole curve and the girths over the metatarsals bones as other foot zones are still under development.

4.1. Bottom pattern design

The insole of the shoe follows the bottom pattern design of the last, which follows the shape of the foot outline. The foot outline can be modeled with a set of 8 control points through a NURBS curve [9]. For the insole, a third order periodic B-Spline with uniform knot vector and 8 control points is chosen (Figure 2b), this ensures continuity up to the curve's first derivative. Before creating the equidistant sampling, the toe zone points are replaced with the convex hull of the inlier points to the plane so the footprint is obtained as shown in Figure 2b. The constrained optimization problem used to encode the shoe last design requirements is defined as follows:

1. Minimize the sorted points not overlapped by the B-Spline curve when plotted in the same graph.
2. The search area for each control point B_i is within a radius of 1.5 cm from its initial position.
3. As constraint, the absolute difference between the two heel control points $\{B_0, B_1\}$ must be lower to 5 mm in the Y axis direction.

The above optimization is done for each control polygon vertex B_n with it's preceding and consecutive control points



(a) Foot segmentation, localization and orientation.

(b) Footprint points (circles), B-Spline curve approximation (line), 8 control points (stars).

(c) Illustration of the different steps carried to map the foot surface to the shoe last surface.

Figure 2. Workflow to create shoe last surfaces from parametric curves.

$B_{n-1 \ mod \ 7}, B_{n+1 \ mod \ 7}$. As the curve is periodic, negative indexing of arrays are used.

The control polygon vertices around the toes $\{B_4, B_5\}$ are translated 1.5 cm in the positive Y direction as the shoe last require a toe allowance and the optimization process produces a B-Spline that follows the foot print.

4.2. Metatarsal girth

To approximate a metatarsal girth a third order B-Spline with open uniform knot vector and 5 control points is chosen. This selection let us fix the endpoints of the curve which must be in the insole's curve. As we can see in Figure 2c, the curve will be used to create the upper surface of the shoe last in the metatarsal region of the foot. Equidistant sample points of the convex hull are obtained and then sorted. Once again, the least-square fit of the B-Spline is used as initial guess for the optimization. The optimization problem is defined as follow:

1. The parameter to optimize is the length x between the first B_0 and the second B_1 control point
2. Minimize the curve length difference between the convex hull and the B-Spline.
3. The angle between the line B_0B_1 and the X axis must be 75°
4. The distance between the control points B_3, B_4 is a fixed fraction of the distance between B_0 and B_1. The angle between the line B_3B_4 and the X axis is 105°.

Further curves need to be approximated to complete the design of the shoe last. The process will be similar to the described with the main differences over the bounds and constraints of the optimization as this helps to encode the design requirements for the shoe last.

5. Conclusions

Future users of real time 3D scanner devices will create larges amounts of 3D data that can be used beyond the virtual world. The authors propose a computer vision system that maps a human's feet to a pair of personalized shoe lasts that can be manufactured with a 3D printer, without expert knowledge. A shoe maker can use it to create personalized shoes, not only in fashion or function but also in shape. Such a system will help people that can not find proper sized shoes to reduce their risk to suffer foot deformities.

The system consist of a handheld 3D Scanner based on the Realsense™ L515 RGB-D sensor and Realsense™ T265 tracking sensor. To integrate the frames captured by the device a modified version of the Open3D legacy offline reconstruction system is used. The modifications improve the accuracy of the 3D model captured.

With the 3D feet model a process to segment and orient each foot is followed. Once the foot orientation is known the insole pattern and other patterns along the foot's length can be extracted. Each pattern requires an optimization process that encodes the shoe last design characteristics required for the different zones of the foot.

The research can be helpful for other scenarios, for example to create sew patterns for personalized clothes or to do inverse engineering of mechanical pieces. Future work can use Reinforcement Learning to capture in a deep neural network the rules to create the B-Spline curves from the foot surface.

6. Acknowledgement

The authors acknowledge financial support from IPN (SIP-20221495, SIP-20220226) and CONACYT (MSc. Scholarship and FORDECYT-PRONACES 6005).

References

[1] A. Alapetite, Z. Wang, J. P. Hansen, M. Zajączkowski, and M. Patalan. Comparison of three off-the-shelf visual odometry systems. *Robotics*, 8(3), 2020. 2

[2] J. Ales, J. Zabkar, and S. Dzeroski. Foot width dispersion of male customers in the us and canada. *Footwear Science*, 11(sup1):S163–S165, 2019. 1

[3] P. W. Anggoro, M. Tauviqirrahman, J. Jamari, A. P. Bayuseno, J. Wibowo, and Y. D. Saputro. Optimal design and fabrication of shoe lasts for ankle foot orthotics for patients with diabetes. *INTERNATIONAL JOURNAL OF MANUFACTURING MATERIALS AND MECHANICAL ENGINEERING*, 9(2):62–80, APR-JUN 2019. 1

[4] Xiaobo Bai, Omar Huerta, Ertu Unver, James Allen, and Jane E. Clayton. A parametric product design framework for the development of mass customized head/face (eyewear) products. *APPLIED SCIENCES-BASEL*, 11(12), JUN 2021. 1

[5] Michael J Coughlin. Common causes of pain in the forefoot in adults. *The Journal of bone and joint surgery. British volume*, 82(6):781–790, 2000. 1

[6] Hyunjung Han, Hyunsook Han, and Taehoon Kim. Patternmaking for middle-aged women's swimsuit applying 3d scan pattern development. *INTERNATIONAL JOURNAL OF CLOTHING SCIENCE AND TECHNOLOGY*, 32(5):743–759, SEP 7 2020. 1

[7] K. A. Johannessen and E. Fonn. Splipy: B-spline and NURBS modelling in python. *Journal of Physics: Conference Series*, 1669(1):012032, oct 2020. 2

[8] Shiya Li, Usman Waheed, Mohanad Bahshwan, Louis Zizhao Wang, Livia Mariadaria Kalossaka, Jiwoo Choi, Franciska Kundrak, Alexandros Lattas, Stylianos Ploumpis, Stefanos Zafeiriou, and Connor William Myant. A scalable mass customisation design process for 3d-printed respirator mask to combat covid-19. *RAPID PROTOTYPING JOURNAL*, 27(7):1302–1317, AUG 3 2021. 1

[9] Ameersing Luximon, Ravindra Goonetilleke, and Kwok-Leung Tsui. Foot landmarking for footwear customization. *Ergonomics*, 46:364–83, 04 2003. 2

[10] David F. Rogers. *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001. 2

[11] Ning Shi, Shuping Yi, Shuping Xiong, and Zuhua Jiang. A cad system for shoe last customization. *International Joint Conference on Computational Sciences and Optimization: 24-26th April 2009; Sanya, China*, 1:957–960, 04 2009. 1

[12] Ricardo C. Villarreal-Calva, Ponciano J. Escamilla-Ambrosio, and Juan H. Sossa-Azuela. Feet point cloud orientation, localization and semantic segmentation. *Research in Computing Science*, 150(11), 2021. In press. 2

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. 2

[14] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 1