Fit2Ear: Generating Personalized Earplugs from Smartphone Depth Camera Images

Haibin Zhao
haibin.zhao@kit.edu
Karlsruhe Institute of Technology

Tobias Röddiger
tobias.roeddiger@kit.edu
Karlsruhe Institute of Technology

Yufei Feng
yufei.feng@faps.fau.de
Friedrich-Alexander University Erlangen-Nürnberg

Michael Beigl
michael.beigl@kit.edu
Karlsruhe Institute of Technology

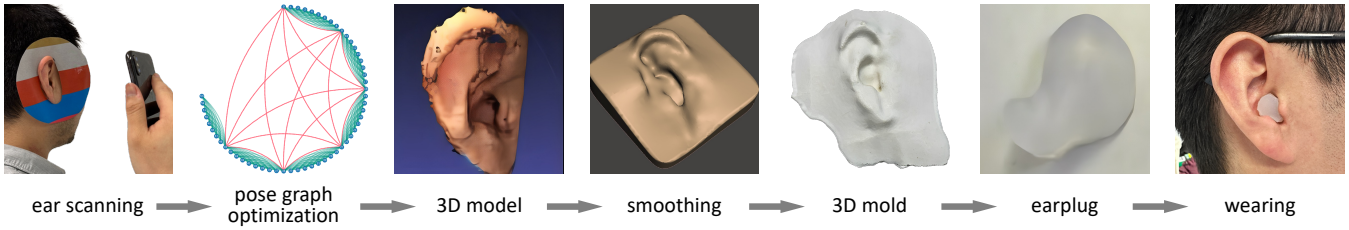

Figure 1: Fit2Ear leverages the TrueDepth camera of iPhones to capture depth scans from different angles of the ears that are fused to generate personalized earplugs.

ABSTRACT

Earphones, due to their deep integration into daily life, have been developed for unobtrusive and ubiquitous health monitoring. However, these advanced algorithms greatly rely on the high quality sensing data. However, the data collected with universal earplugs could potentially generate undesirable noise, such as vibrations or even falling off. As a result, the algorithms may exhibit limited performance. In this regard, we build a dataset containing RGBD and IMU data captured by a smartphone. To provide a precise and solid ground truth, we employ additional control information from a robotic arm that holds the smartphone scanning ears along a predefined trajectory. With this dataset, we propose a tightly coupled information fusion algorithm for the ground truth ear modeling. Finally, we fabricate the earplugs and conduct an end-to-end evaluation of the wearability of the modeled earplugs in a user study.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**; • **Hardware** → *Sensor devices and platforms.*

KEYWORDS

earable, earplug, sensor fusion, 3D modeling, customization

UbiComp Companion '24, October 5–9, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1058-2/24/10.
https://doi.org/10.1145/3675094.3680525

ACM Reference Format:
Haibin Zhao, Tobias Röddiger, Yufei Feng, and Michael Beigl. 2024. Fit2Ear: Generating Personalized Earplugs from Smartphone Depth Camera Images. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '24), October 5–9, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3675094.3680525

1 INTRODUCTION

Earphones today are already pervasive as they can offer private audio spaces. Recently, there is a trend of equipping them with advanced sensing capabilities [20], such as inertial measurement units (IMU) for activity recognition [13] or Electroencephalography (EEG) for monitoring brain activity [2].

However, the unique characteristics of the human ear make it challenging to develop one-size-fits-all earables that are both comfortable and adaptable. This becomes particularly significant when the measurement should be taken in an unobtrusive manner (e.g., during sleeping [22]). Also, poorly fitting earables can produce sensing noise or even fall out, which may significantly impact the performance of health monitoring functions based on fine and stable measurements [23]. Moreover, some signal based on ear pressure [21] necessitates the tight sealing of the ear canal. Consequently, a process that allows easy fitting of earables to improve data quality and wearer comfort is desirable. As by-product, measuring the ear shape of a user can also be leveraged to improve audio rendering quality [34].

Generally, modeling personalized earplugs is a costly and time-consuming process. A traditional way to obtain the personalized earplugs is to take a mold of the ear at the physician, but they necessitate additional appointments. Alternatively, modern devices such as Artec® can generate a virtual scan of the ear canal, but are

costly. Either way, it is unlikely that users will go to a specialized store and spend large amounts of money for personal-fit earplugs.

Therefore, our vision is to leverage the depth-scanning capabilities of smartphone cameras to model users' ears for personalized earplugs. The main challenge is developing an algorithm that can build high precision models under noisy depth scans. Working towards this goal, the main contributions of this work are:

- A dataset[1] comprising 96 ear scans, collected from 24 ears of 12 users. The scans were obtained according to two principles: freehand scanning by the user themselves, and structured scanning by a robotic arm to provide ground truth information. For each principle, scans were conducted both with and without a supplementary feature card to facilitate point cloud registration.
- We propose a pose graph optimization-based modeling algorithm which effectively fuses point clouds, IMUs, and the joint angles of the robotic arm. In this way, we provide solid ground truths for the ear models. The effectiveness of the algorithm is then validated with a user study.

We envision that our dataset will help to create algorithms that can be used for modeling highly precise human ears. Meanwhile, as deep learning based 3D reconstruction [10, 26] is growing fast, this dataset may also contribute as a training resource of this field. Moreover, for a more advanced scientific direction, i.e., the generative AI models [5, 16], this dataset also serves to complement the development of generative models for 3D object generation.

2 DATA ACQUISITION PIPELINE

This section introduces the background of the data required for 3D modeling that is captured by general smartphones. Subsequently, based on the required data, we establish the infrastructure employed to capture the dataset. Lastly, we describe the process of capturing the Fit2Ear dataset. The hardware consists of an iPhone X to capture RGBD and inertial information. In addition, we employed a robotic arm to hold and move the iPhone along a designed trajectory. The trajectory of the robotic arm serves as the ground truth of this dataset. In terms of software, we programmed an application for the iPhone to capture and transmit data, and built a server to control the hardware and process the data stream.

2.1 Data for Ear Modeling

The essential task of 3D reconstruction involves determining the coordinates of the target entities, specifically the point cloud of the entities' surface, within the world coordinate system. The point cloud is denoted by $P^{\mathrm{W}} \in \mathbb{R}^{3\times N}$, consisting of N points with 3 coordinates (x, y, z) for each point. This process encompasses 3 primary stages:

1. Obtain a point cloud representing a fragment of the object surface in the device coordinate system. This point cloud is denoted by $P_k^{\mathrm{D}} \in \mathbb{R}^{3\times N}$;
2. Capture point clouds from multiple perspectives by moving the device (with respect to the world coordinate system), denoted by $P_k^{\mathrm{D}}, k = 1, 2, \cdots$;
3. Estimate the device poses in the world coordinate system at each capture, denoted by the transfer matrices $T_k \in \mathrm{SE}(3)$. Afterwards, point cloud P_k^{D} from the device coordinate system to the world coordinate system through $P_k^{\mathrm{W}} = T_k \cdot P_k^{\mathrm{D}}$.

For the first point, there are several viable solutions for smartphones: *(a)* through dual cameras [31] or structured light [3], triangulation principle can be used to estimate the object's depth, *(b)* through LiDAR, the depth can be calculated based on time-of-flight (ToF) principle [11], and *(c)* in 6-generation (6G) communication technology, the electromagnetic waves can be directly used as a radar to model surrounding objects, known as integrated sensing and communication (ISAC) [17]. Among these, option *(a)* is most commonly supported by smartphone hardware. Many manufacturers have provided APIs that allow direct access to RGB data with corresponding pixel depth information. With these data, as well as the camera intrinsic matrix, we are able to convert a RGBD image into a point cloud. Therefore, we employ option *(a)* as the primary data acquisition method.

Regarding the second point, given that the purpose of the Fit2Ear dataset is to facilitate the establishment of customized 3D ear models through convenience scanning, users will be instructed to hold the phone with freehand, and scan around their ears, more details are described in Section 2.3.

The third point addresses the primary challenge for high precision 3D modeling, which involves accurately estimating the pose T_k of the device within the world coordinate system at each time step k. Through precise pose estimation, we can then effectively fuse multiple point cloud fragments to form a complete 3D model. A naive estimation approach is to conduct point clouds registration between consecutive frames using, e.g., iterative closest point (ICP) [33] or feature point-based methods [14, 24, 25]. However, due to inherent measurement uncertainties, estimating the pose using only point cloud data will inevitably introduce errors. To mitigate these errors, we incorporate an additional information source: the inertial measurement unit (IMU) data. By integration of the IMU data, we are able to provide another estimation of the device pose. Afterwards, by fusing both information, we will be able to further reduce the errors in pose estimation. In addition, to provide a solid ground truth of the device pose T_k, we employ a robotic arm to hold the smartphone moving along a designed trajectory for ear scanning. In this way, we can provide a precise estimation of the device pose during scanning. More details can be found in Section 4.

2.2 Infrastructure

Based on the background on 3D modeling, we designed and built our software and hardware infrastructure for data acquisition.

Smartphone. We applied an iPhone X with its TrueDepth Camera[2] and the IMU[3] to capture the data. Specifically, the TrueDepth Camera captures RGBD data, timestamps, and the camera intrinsic matrices. To enable the collection and transfer of these data, we programmed an iOS app and sent to the server via SocketIO. Notably,

[1] The dataset can be found at https://github.com/OpenEarable/Fit2Ear.

[2] https://developer.apple.com/documentation/avfoundation/cameras_and_media_capture/streaming_depth_data_from_the_truedepth_camera.

[3] https://developer.apple.com/documentation/coremotion.

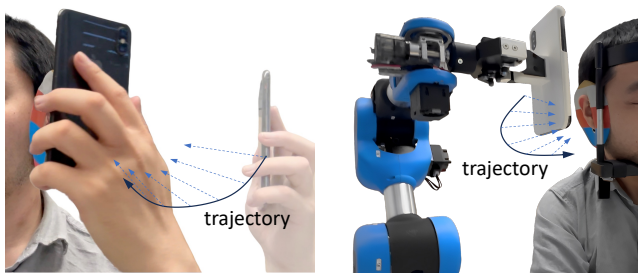

Figure 2: Two scanning principles during data collection. Left: freehand scanning that emulates the real case data capture. The users hold the smartphone and move it around the ear. Right: robotic arm scanning that facilitates the ground truth. The robotic arm holds the phone and robustly moves along a defined trajectory centered around the ear. The user's head is fixed by a chin rest.

although we employed iPhone in our setup, other smartphones usually provide similar functionalities as well[4].

Robotic arm. As mentioned before, in addition to free hand scanning, we also use Niryo® Ned to hold the iPhone moving along a circular-shaped trajectory centered on the ear. This setup aims to provide a solid ground truth for 3D modeling (see Section 4). Because the robotic arm can not only grasp the smartphone and precisely move it along a predefined trajectory for high quality scanning, but also provide precise pose information of the iPhone. Moreover, to account for any deviations of the actual trajectory from the predefined one, the joint angles of the robotic arm are recorded at each time step, allowing for the determination of the actual trajectory of the smartphone movement through the forward kinematics of the robotic arm.

Web application and server. For communication and handling all data, we implemented a simple web app which connects to the smartphone and robotic arm via a *node.js* web server running locally on a laptop. We use the web app to start and stop data recording and configure high level information like participant data.

Artificial features. To improve the robustness of the modeling algorithm, we printed a feature card — a flat surface with different colors and an ear-shaped hollow in the middle — which provides additional color and geometric information and prevents hair from obscuring the ear, as shown in Figure 1. This feature card is utilized not only for robotic-based scanning to obtain ground truth data, but also for freehand scans. We believe such color-printed papers are easily acquired with a standard printer which is commonly available. Nevertheless, we provide scans both with and without this feature card in our dataset.

2.3 Scanning Principle

As previously mentioned, the scanning of this dataset adhered to two principles which each have their own purpose: *freehand scanning* (which intends to simplify ear modeling using a smartphone), and *robotic arm scanning* (which aims to generate highly precise ground truth models).

Freehand scanning. As shown in Figure 2 (left), the users are asked to hold the smartphone without any external help. They point the camera at their ear and rotate their arm to scan the ear from front to back. This data should serve as the target for future algorithmic research in ear modeling, as it emulates the quality of data from actual scenarios, where the users capture their ear information from a smartphone by their own.

Robotic arm scanning. We set up a chin-rest for the users to place their heads on and keep them still, see Figure 2 (right). Afterwards, the robotic arm is activated, moving the smartphone along a prescribed trajectory to capture RGBD and IMU information. To account for any deviations of the actual trajectory from the predefined one, the joint angles of the robotic arm are recorded at each time step, allowing for the determination of the actual trajectory of the smartphone movement through the forward kinematic of the robotic arm.

Table 1: Collected data. (FC=feature card, JA=joint angle)

Folder Name	Side	Capture	FC	RGB-D	IMU	JA
{label}_L_R_F	Left	Robotic arm	✓	✓	✓	✓
{label}_L_R	Left	Robotic arm		✓	✓	✓
{label}_L_H_F	Left	Free hand	✓	✓	✓	
{label}_L_H	Left	Free hand		✓	✓	
{label}_R_R_F	Right	Robotic arm	✓	✓	✓	✓
{label}_R_R	Right	Robotic arm		✓	✓	✓
{label}_R_H_F	Right	Free hand	✓	✓	✓	
{label}_R_H	Right	Free hand		✓	✓	

3 DATASET

In our dataset, we collected information from 12 subjects. For each subject, we collected data 8 times, namely: the left & right ear, capturing with robotic arm & free hand, and with & without feature card, i.e., there are 96 recordings in total in the dataset. Each data is saved in a folder named according to a pseudonym label of the subject, see Table 1, and each data lasts for around 10 seconds.

RGBD. In RGBD data, the RGB part is stored as standard *jpeg* images, while depth and intrinsic matrices are encoded as raw float32 byte array, and we provide a Python script to load them. Additionally, there is a file that contains the timestamps in int64 format. All data are collected at 10 Hz.

IMU. The whole IMU data is stored in IMU.data files, including multiple rows and 17 columns. The first column refers to int64 timestamps. Subsequently, the next 9 columns represent 9-axis IMU data, i.e., acceleration (including gravity), angular velocity, and magnetic strength in x, y, and z directions. The remaining 7 columns are redundant to the 9-axis IMU data, which denote the quaternion (4 columns) for device pose, and acceleration (3 columns, without gravity). The last 7 columns are directly read from CoreMotion API, developed by Apple. The sampling frequency is 100 Hz.

Joint angles. Joint angles of the robotic arm are saved in joints.txt as strings and can be seen as a matrix with 7 columns: the first

[4]https://developer.android.com/reference/android/graphics/ImageFormat#DEPTH16.

column contains the timestamps and the remaining 6 the joint angles. The sampling rate is 10 Hz.

Table 2: Description of data. (IM=intrinsic matrix, TS=time stamp, JA=joint angle)

Information		Filename	Data Type	Frequency
RGB-D	RGB	image_{*n*}.jpeg	jpeg	10 Hz
	Depth	depth_{*n*}.data	float32	
	IM	intrinsic_{*n*}.data	float32	
	TS	time_{*n*}.data	int64	
IMU	IMU	IMU.data	double	100 Hz
	TS		int64	
JA	JA	joints.txt	string	10 Hz
	TS			

Table 2 summarizes the detailed information of the dataset. In addition to the raw data, we initially converted each RGBD frame into the corresponding point cloud with respect to the device coordinate system using the corresponding camera intrinsic matrices.

4 GROUND TRUTH MODELS

Building a high quality 3D model relies on calculating the precise pose of each point cloud in the world coordinate system. In this section, we briefly review the algorithms for directional state estimation and propose our approach.

4.1 Background

To build a 3D model from a series of point clouds, it is essential to estimate the poses T_k to transform the point cloud from the device coordinate system into the world coordinate system, i.e.,

$$P_k^{\mathrm{W}} = T_k \cdot P_k.$$

Due to the sensing uncertainty, the estimated pose $\hat{T}_k$ generally differs from the true value T_k, denoted by

$$\hat{T}_k = \Delta_k \cdot T_k,$$

where $\Delta_k \in \mathrm{SE}(3)$ depicts the residual (error) of the estimation. Moreover, in practice, $\hat{T}_k$ is usually estimated iteratively in between consecutive measurements, i.e.,

$$\hat{T}_k = \prod_{i=0}^{k-1} \hat{T}_{i,i+1} \cdot T_0 = \left(\prod_{i=0}^{k-1} \Delta_{i,i+1} \cdot T_{i,i+1} \right) \cdot T_0, \tag{1}$$

where i, k denote timestamps, $\hat{T}_{i,j} \in \mathrm{SE}(3)$ represents the transfer matrix from the i-th frame to the j-th, which is estimated through the measurement from a certain device, e.g., the registration of point clouds.

For high-precision 3D modeling, reducing estimation residual $\Delta_{i,j}$ is critical. Therefore, multi-modal sensing systems [6, 28] are usually employed to minimize the sensing error $\Delta_{i,j}$. Mathematically, the estimation error between two frames is expressed by

$$\begin{aligned} & T_j = T_{i,j} \cdot T_i \\ \Rightarrow \Delta_{i,j}^d \cdot T_j = & \underbrace{\Delta_{i,j}^d \cdot T_{i,j}}_{=\hat{T}_{i,j}^d} \cdot T_i = \hat{T}_{i,j}^d \cdot T_i, \end{aligned}$$

where, $d \in D$ refers to data from different devices and $\Delta_{i,j}^d$ denotes the estimation residual from the information provided by the device d. Consequently, residual from device d is given by

$$\Delta_{i,j}^d = \hat{T}_{i,j}^d \cdot T_i \cdot T_i^{-1}.$$

With this expression, the fusion of different information source is implemented through

$$\min_{T_i, T_j} \sum_{d \in D} \frac{1}{\sigma^d} \left\| \Delta_{i,j}^d \right\| = \sum_{d \in D} \frac{1}{\sigma^d} \left\| \hat{T}_{i,j}^d \cdot T_i \cdot T_j^{-1} \right\|.$$

Here, σ^d denotes the sensing uncertainty of the device d, which is an inherent factor of the device.

This optimization problem for pose estimation can also be simply described by a *pose graph* [4], as exemplified in the second block in Figure 1. There, the nodes indicate optimization variables T_i, while the edges between two nodes indicate the estimation residual, e.g., $\| \hat{T}_{i,j}^d \cdot T_i \cdot T_j^{-1} \|$, and different colors means different devices, i.e., $\hat{T}_{i,j}^d$. Consequently, the overall optimization problem can be seen as the weighted sum of the residuals on all edges. As the effectiveness of pose graph optimization has been validated by many state-of-the-art works [18, 19, 30], we utilize and adapt this method to build the ground truth models for the Fit2Ear dataset.

Another problem posed in Equation (1) is that, the error will accumulate over time. To address this issue, we introduce a robotic arm to provide pose estimation with time-invariant errors. Specifically, since the joint angles of the robotic arm (and thus the pose of the device) is measured independently at each time step, the error does not propagate over time, therefore, it can provide a stable reference for the pose estimation for ground truth.

4.2 Proposed Method

To leverage the time-invariant measurement of a robotic arm into the pose estimation, we utilize and adapt the state-of-the-art pose graph. The proposed method allows for a tightly-coupling [12] of RGBD, IMU, and robotic arm pose. The 2nd block in Figure 1 represents our graph: Each node corresponds to a pose matrix T_i, the gray edges between adjacent nodes represent the pre-integration of the IMU [9]. The green edges indicate residuals from RGBD data, which are derived from point cloud registration based on feature point-based method, namely fast point feature histogram (FPFH) [25]. The red edges represent residuals from robotic arm poses, calculated by the joint angles with the forward kinematics of the robotic geometric.

Notably, this algorithm splits the nodes into multiple groups and performs a full smoothing [9] (i.e., fully connected) point cloud registration within each group. The advantage of this approach is that: full smoothing ensures high modeling precision, while the grouping prevents registration failures between point clouds with significant difference. Further, on a larger time scale, i.e., across groups, we use the pose information of the robotic arm as the primary. This leverages the advantage that the measurement uncertainty of the robotic joint angles does not accumulate over time, enhancing robustness over large time scales.

Although the optimization variables $T_i,\ i = 1, 2, \cdots, K$ are represented as matrices in SE(3) group, they essentially possess only 6 degrees of freedom, including 3 rotational and 3 translational

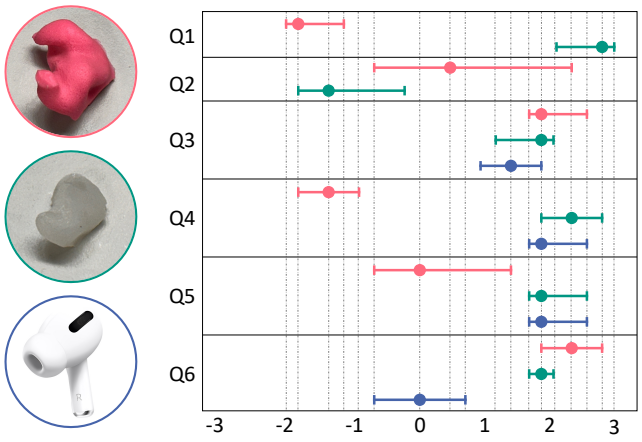

Figure 3: Left: medical earplugs (denoted in pink), our Fit2Ear earplug (denoted in green), and one-fit-all AirPods (denoted in blue). Right: result of the Likert scale questionnaire, points denote median values and bars denote IQR values. Higher scores refer to more positive evaluation.

values. Consequently, the elements in the matrices can not be independently updated during the optimization process, as the update can not ensure that T_i remains on the SE(3) manifold. To address this issue, we first convert T_i into its corresponding Lie algebra [27] for optimization with Sophus[5]. Lie algebra represents the SE(3) transfer matrices directly with 6 variables. In other word, it can transform the optimization problems into an unconstrained one, while guarantees the problem to be solved on the SE(3) manifold. Then, we are allowed to do unconstrained optimization to minimize the ℓ_2 norm of $\Delta_{i,j}$ directly in Lie algebra domain using Ceres [1]. With this approach, we can effectively perform the optimization and subsequently convert it back to the SE(3) matrices.

4.3 Post-Processing

After calculating the optimal pose of each point cloud fragment and converting them from device coordinate system into the world coordinate system, an (optional) post-processing stage is performed. This stage converts the point cloud-based 3D models into physical entities, as illustrated in the 3rd to the 6th blocks of Figure 1.

Specifically, we first remove the invalid points (like human face or background) with MeshLab[6] and convert the point clouds to solid bodies followed by a surface smoothing operation in MeshMixer[7]. Afterwards, we 3D print the molds, cast them in silicone, and finally excised extraneous materials.

5 FEASIBILITY TEST

To provide an end-to-end validation of the ground truth, we conduct a user study to test the wearability of the earplugs captured from the smartphone. Specifically, we modeled the ears with the feature card and all sensor modalities, i.e., both RGBD, IMU, and joint angles. Then, we randomly selected 6 ear models for subsequent post-processing. Subsequently, we invited the respective participants to a professional facility to create medical earplugs (Figure 3). Finally, we invited them to wear the medical earplugs, Fit2Ear earplugs (ours), and AirPods (a one-fit-all solution). We asked them to wear the earplugs for 15 minutes while doing everyday tasks and activities such as reading, using their smartphones, walking, jogging, and other spontaneous behaviors. Afterwards, we ask them to rate the following items on a 7-point Likert scale [15], ranging from *Strongly Disagree* to *Strongly Agree*:

*Q1: I feel comfortable during earplug modeling.
*Q2: It takes a short waiting time to obtain this earplug.
Q3: The earplug fits me well.
Q4: It is easy to put on the earplug.
Q5: I feel comfortable wearing the earplug.
Q6: I **wasn't** worried about the earplug falling out while wearing.
* Questions do not apply to AirPods.

Here, the first two questions aim to investigate the modeling stage of the earplugs, while the remaining questions survey the wearability of the earplugs. The wearability encompasses not only the process of putting the earplugs on, but also their comfort while worn and the stability of the earplugs during both static and dynamic activities. Q6 is designed in a negative expression, because we want to keep the identical metric of the Likert scale, i.e., the evaluation would become more positive if the score is higher. We report the results (median and IQR values) of the questionnaire in Figure 3.

It can be seen, in terms of the ease of acquisition, Fit2Ear earplugs surpass medical earplugs, as the latter necessitate complete filling and blocking of the ear concha and the outer ear canal during the modeling, which is quite invasive and uncomfortable.

Regarding the wearability, although Fit2Ear earplugs do not fit as precisely as medical earplugs (Q3) and exhibits slightly lower stability during usage (Q6), they are significantly simpler to wear (Q4) and more comfortable during usage (Q5). We speculate that, this is because medical earplugs focus excessively on the fine structure of the ears, thus overlooking their wearability. By comparing Fit2Ear earplugs to commercially available AirPods, it is evident that although AirPods achieved similar ratings to Fit2Ear in terms of ease of wear (Q3) and comfort (Q4), users are more concerned about AirPods falling out in some activities (Q6). This is due to the disregard of AirPods on the fine structure of the ear concha and canal, which reduces their unobtrusiveness, wearability, and stability.

In terms of the modeling stage, since the post-processing of Fit2Ear is less mature and takes multiple steps, it takes longer waiting time for the earplug fabrication than medical earplugs. However, this issue might be addressed in the future through, e.g., calculating the negative model (i.e., earplugs) in the software level and 3D printing them directly with materials like silicone.

6 CONCLUSION AND OUTLOOK

This study focuses on the wearability of the earplugs and the signal quality captured on the human ear. To improve these aspects, this work first presents a dataset comprising RGBD, IMU, and robotic arm pose data (as ground truth), supportive for high-precision

[5]https://pypi.org/project/sophus.
[6]https://www.meshlab.net.
[7]https://meshmixer.com.

human ear modeling. With this dataset, we then propose a tightly-coupled algorithm for achieving high-precision modeling. This algorithm enables to effectively reconstruct the 3D ear models of the subjects. To evaluate the captured dataset and proposed algorithm, we conduct a user study to validate its effectiveness.

In the future, the collected dataset can not only facilitate research on personalized earables through smartphones, but it also contributes to numerous utilities. Apart from the aforementioned deep learning-based 3D reconstruction and generative models for 3D objects. In ear-based research, such as authentication [29], recognition [8], or identification [32], our dataset can provide multi-modal data. In terms of sensor fusion tasks [7], this dataset may also serve as a benchmark. We hope more researchers can utilize our dataset and explore further exciting applications.

ACKNOWLEDGMENTS

This work has been funded by the German Federal Ministry of Education and Research as part of the Software Campus initiative in the "Fit2Ear" project. We thank Dr. Jens Ottnand and Dr. Markus Scholz from TRUMPF GmbH + Co. KG for being our project partner.

REFERENCES

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. 2023. *Ceres Solver*. https://github.com/ceres-solver/ceres-solver

[2] Chanavit Athavipach, Setha Pan-Ngum, and Pasin Israsena. 2019. A Wearable In-Ear EEG Device for Emotion Monitoring. *Sensors* 19, 18 (2019), 4014.

[3] Andreas Breitbarth, Timothy Schardt, Cosima Kind, Julia Brinkmann, Paul-Gerald Dittrich, and Gunther Notni. 2019. Measurement accuracy and dependence on external influences of the iPhone X TrueDepth sensor. In *Photonics and Education in Measurement Science 2019*, Vol. 11144. SPIE, 27–33.

[4] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. 2015. Initialization Techniques for 3D SLAM: a Survey on Rotation Estimation and its Use in Pose Graph Optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 4597–4604.

[5] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. 2020. Learning Generative Models of 3D Structures. In *Computer graphics forum*, Vol. 39. Wiley Online Library, 643–666.

[6] Ling Chen and Huosheng Hu. 2012. IMU/GPS Based Pedestrian Localization. In *2012 4th Computer Science and Electronic Engineering Conference (CEEC)*. IEEE, 23–28.

[7] Wilfried Elmenreich. 2002. An Introduction to Sensor Fusion. *Vienna University of Technology, Austria* 502 (2002), 1–28.

[8] Žiga Emeršič, Vitomir Štruc, and Peter Peer. 2017. Ear recognition: More than a survey. *Neurocomputing* 255 (2017), 26–39.

[9] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. 2016. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Transactions on Robotics* 33, 1 (2016), 1–21.

[10] Kui Fu, Jiansheng Peng, Qiwen He, and Hanxiao Zhang. 2021. Single image 3D object reconstruction based on deep learning: A review. *Multimedia Tools and Applications* 80, 1 (2021), 463–498.

[11] Mehmet Akif Günen, İlker Erkan, Şener Aliyazıcıoğlu, and Cavit Kumaş. 2023. Investigation of geometric object and indoor mapping capacity of Apple iPhone 12 Pro LiDAR. *Mersin Photogrammetry Journal* 5, 2 (2023), 82–89.

[12] Chao Hu, Shiqiang Zhu, Yiming Liang, and Wei Song. 2022. Tightly-Coupled Visual-Inertial-Pressure Fusion Using Forward and Backward IMU Preintegration. *IEEE Robotics and Automation Letters* 7, 3 (2022), 6790–6797.

[13] Jingyang Hu, Hongbo Jiang, Daibo Liu, Zhu Xiao, Qibo Zhang, Jiangchuan Liu, and Schahram Dustdar. 2023. Combining IMU With Acoustics for Head Motion Tracking Leveraging Wireless Earphone. *IEEE Transactions on Mobile Computing* (2023).

[14] Zehua Jiao, Rui Liu, Pengfei Yi, and Dongsheng Zhou. 2019. A Point Cloud Registration Algorithm Based on 3D-SIFT. *Transactions on Edutainment XV* (2019), 24–31.

[15] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert Scale: Explored and Explained. *British journal of applied science & technology* 7, 4 (2015), 396–403.

[16] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[17] Danny Kai Pin Tan, Jia He, Yanchun Li, Alireza Bayesteh, Yan Chen, Peiying Zhu, and Wen Tong. 2021. Integrated Sensing and Communication in 6G: Motivations, Use Cases, Requirements, Challenges and Future Directions. In *2021 1st IEEE International Online Symposium on Joint Communications & Sensing (JC&S)*. 1–6.

[18] Johannes Pöschmann, Tim Pfeifer, and Peter Protzel. 2020. Factor Graph based 3D Multi-Object Tracking in Point Clouds. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10343–10350.

[19] Tong Qin, Peiliang Li, and Shaojie Shen. 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics* 34, 4 (2018), 1004–1020.

[20] Tobias Röddiger, Christopher Clarke, Paula Breitling, Tim Schneegans, Haibin Zhao, Hans Gellersen, and Michael Beigl. 2022. Sensing with Earables: A Systematic Literature Review and Taxonomy of Phenomena. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–57.

[21] Tobias Röddiger, Christopher Clarke, Daniel Wolffram, Matthias Budde, and Michael Beigl. 2021. EarRumble: Discreet Hands- and Eyes-Free Input by Voluntary Tensor Tympani Muscle Contraction. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[22] Tobias Röddiger, Christian Dinse, and Michael Beigl. 2021. Wearability and Comfort of Earables During Sleep. In *Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 150–152.

[23] Tobias Röddiger, Daniel Wolffram, David Laubenstein, Matthias Budde, and Michael Beigl. 2019. Towards Respiration Rate Monitoring Using an In-Ear Headphone Inertial Measurement Unit. In *Proceedings of the 1st International Workshop on Earable Computing*. 48–53.

[24] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*. Ieee, 2564–2571.

[25] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. 2009. Fast Point Feature Histograms (FPFH) for 3D Registration. In *2009 IEEE international conference on robotics and automation*. IEEE, 3212–3217.

[26] Taha Samavati and Mohsen Soryani. 2023. Deep learning-based 3D reconstruction: A survey. *Artificial Intelligence Review* 56, 9 (2023), 9175–9219.

[27] Jon M Selig. 2004. Lie Groups and Lie Algebras in Robotics. In *Computational Noncommutative Algebra and Applications*. Springer, 101–125.

[28] Shaojie Shen, Nathan Michael, and Vijay Kumar. 2015. Tightly-Coupled Monocular Visual-Inertial Fusion for Autonomous Flight of Rotorcraft MAVs. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5303–5310.

[29] Riccardo Spolaor, QianQian Li, Merylin Monaro, Mauro Conti, Luciano Gamberini, and Giuseppe Sartori. 2016. Biometric Authentication Methods on Smartphones: A Survey. *PsychNology Journal* 2 (2016).

[30] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. 2022. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics* 38, 4 (2022), 2053–2073.

[31] Jinrui Zhang, Huan Yang, Ju Ren, Deyu Zhang, Bangwen He, Ting Cao, Yuanchun Li, Yaoxue Zhang, and Yunxin Liu. 2022. MobiDepth: Real-Time Depth Estimation Using On-Device Dual Cameras. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 528–541.

[32] Lin Zhang, Zhixuan Ding, Hongyu Li, and Ying Shen. 2014. 3D Ear Identification Based on Sparse Representation. *PloS one* 9, 4 (2014), e95506.

[33] Zhengyou Zhang. 2021. Iterative closest point (ICP). In *Computer vision: a reference guide*. Springer, 718–720.

[34] DYN Zotkin, Jane Hwang, R Duraiswaini, and Larry S Davis. 2003. HRTF personalization using anthropometric measurements. In *2003 IEEE workshop on applications of signal processing to audio and acoustics (IEEE Cat. No. 03TH8684)*. Ieee, 157–160.