Acuity: Creating Realistic Digital Twins Through Multi-resolution Pointcloud Processing and Audiovisual Sensor Fusion

Jason Wu
Electrical and Computer Engineering Department,
University of California, Los Angeles
Los Angeles, California 90095, US
jaysunwu@g.ucla.edu

Ziqi Wang
Electrical and Computer Engineering Department,
University of California, Los Angeles
Los Angeles, California 90095, US
wangzq312@g.ucla.edu

Ankur Sarker
Electrical and Computer Engineering Department,
University of California, Los Angeles
Los Angeles, California 90095, US
as4mz@virginia.edu

Mani Srivastava
Electrical and Computer Engineering Department,
University of California, Los Angeles
Los Angeles, California 90095, US
mbs@ucla.edu

ABSTRACT

As augmented and virtual reality (AR/VR) technology matures, a method is desired to represent real-world persons visually and aurally in a virtual scene with high fidelity to craft an immersive and realistic user experience. Current technologies leverage camera and depth sensors to render visual representations of subjects through avatars, and microphone arrays are employed to localize and separate high-quality subject audio through beamforming. However, challenges remain in both realms. In the visual domain, avatars can only map key features (e.g., pose, expression) to a predetermined model, rendering them incapable of capturing the subjects' full details. Alternatively, high-resolution point clouds can be utilized to represent human subjects. However, such three-dimensional data is computationally expensive to process. In the realm of audio, sound source separation requires prior knowledge of the subjects' locations. However, it may take unacceptably long for sound source localization algorithms to provide this knowledge, which can still be error-prone, especially with moving objects. These challenges make it difficult for AR systems to produce real-time, high-fidelity representations of human subjects for applications such as AR/VR conferencing that mandate negligible system latency. We present Acuity, a real-time system capable of creating high-fidelity representations of human subjects in a virtual scene both visually and aurally. Acuity isolates subjects from high-resolution input point clouds. It reduces the processing overhead by performing background subtraction at a coarse resolution, then applying the detected bounding boxes to fine-grained point clouds. Meanwhile, Acuity leverages an audiovisual sensor fusion approach to expedite sound source separation. The estimated object location in the visual domain guides the acoustic pipeline to isolate the subjects' voices without running sound source localization. Our results demonstrate that Acuity can isolate multiple subjects' high-quality point clouds with a maximum latency of 70 ms and average throughput of over 25 fps, while separating audio in less than 30 ms. We provide the source code of Acuity at: https://github.com/nesl/Acuity.

IoTDI '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05.
https://doi.org/10.1145/3576842.3582363

ACM Reference Format:
Jason Wu, Ziqi Wang, Ankur Sarker, and Mani Srivastava. 2023. Acuity: Creating Realistic Digital Twins Through Multi-resolution Pointcloud Processing and Audiovisual Sensor Fusion. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3576842.3582363

1 INTRODUCTION

I. Motivation.[1] Digital twins, the replica of physical subjects in a virtual environment, have found their way in industrial manufacturing [36, 44], agriculture [49], and healthcare [2, 11]. Beyond the realm of industrial productions, they also have numerous personal applications in AR and VR to construct immersive spaces. Various leaders in the technology industry have invested heavily in this space, with Microsoft's Hololens [30] or Meta's Metaverse [21] being among countless others. One of the killer applications is AR/VR conferencing, where people in different physical spaces can seamlessly interact and collaborate. For an immersive AR/VR conferencing experience, we must focus on the problem of **creating real-time authentic digital twins of human subjects in the virtual space**. AR/VR conferencing users mainly perceive the scene audiovisually. Thus, the creation of human digital twins consists of two goals: (1) creating vivid three-dimensional visual representations of human bodies isolated from their original background and (2) streaming separated audio of individual's speech free of any interference from external sound sources. Also, the real-time constraint associated with virtual meetings imposes additional latency bounds when considering data processing.

II. Existing Solutions and Challenges. There are primarily two methods of acquiring sensor data to craft a comprehensive model of the subject: *instrumenting the subject*, and *instrumenting the environment*. Headsets are the primary choice when instrumenting a human subject, as they simultaneously acquire data and render the end scene for viewing [17]. Despite this, the headset solution

[1]Authors' emails: jaysunwu@g.ucla.edu, wangzq312@g.ucla.edu, as4mz@virginia.edu, mbs@ucla.edu

suffers from a few critical flaws: (1) individuals are represented with avatars, breaking immersion and detracting from a realistic experience, (2) individual units must be purchased for every participant rather than having a single system for multiple individuals in a meeting, and (3) some users reportedly suffer from motion sickness [3]. Therefore, we must turn to instrumented environments despite the increased difficulty in working with data requiring heavy post-processing. We instrument the environment with complementary sensing modalities to achieve the two goals (visual and audio) for human digital twin creation. We discuss the data and their corresponding challenges in the following paragraphs.

In the visual realm, the utilization of *point clouds* generated from Light Detecting And Ranging (LiDAR) sensors allow for a detailed portrayal of the subject. LiDAR sensors emit millions of short laser pulses and measure the time of flight to determine the distance to surrounding objects [48]. Aggregating all distance measurements forms a three-dimensional point cloud containing all objects within the field of view. If the points of interest can be colorized with a video stream and efficiently separated from the background, it will be possible to render a lifelike representation of a subject in a virtual environment.

Processing these aforementioned point clouds is a well-researched topic, typically utilizing neural networks for segmentation and classification [33, 35, 58]. While these networks perform effectively on these tasks, they are heavily reliant on a large amount of labeled training data such as the KITTI dataset [13]. These datasets are often targeted towards autonomous vehicle operations, and little research has been undertaken with a focus on human perception. As a result, the resolution of point clouds has been largely overlooked. These neural networks, while working well on sparse point clouds, generalize poorly on indoor dense point clouds. While some research on teleoperations have tried to incorporate dense point clouds [53], such systems do not perform point cloud perception tasks like segmentation. Additionally, processing high-resolution point clouds also incurs a significant computation delay as point cloud size increases [35]. **Real-time processing and separation of high-resolution point clouds for human subjects remains a challenge**.

Efficiently processing visual information only comprises half of the challenge, where audio must likewise be streamed in real-time to connect virtual users. While it is simple to transmit raw audio in real-time, complications arise when users are in noisy environments containing a lot of background noise, necessitating additional processing to isolate their audio. Algorithms such as background noise suppression [45] are commonly used in many modern video conferencing applications. However, they struggle when the background noise is of human speech or if the target audio is too quiet. Such situations may occur in hybrid meetings where multiple people talk simultaneously. Sound source separation (SSS) excels in such situations. Some learning-based methods [42, 43, 46] apply large neural networks to blindly separate speech sources. While they have achieved great success on ideal test data, these methods operate on large audio chunks, causing unacceptable buffering delays for real-time operations. Meanwhile, other signal processing SSS algorithms rely upon a microphone array with known spacing between microphones, leveraging the *Time Difference of Arrival* between microphones to isolate sound targets via beamforming [1, 31, 34] . However, these beamforming algorithms require knowledge of the location and number of the sound sources, resulting in the need to perform expensive Sound Source Localization (SSL).

SSL is primarily done through one of two algorithms: Multiple Signal Classification *(MUSIC)* or Steered Response Power Phase Transform *(SRP-PHAT)*. *MUSIC* calculates the correlation matrix between the microphones in the array, obtaining a power spectrum with eigenvalue decomposition where the peaks are associated with a unique direction of arrival. *SRP-PHAT*, on the other hand, calculates a general cross-correlation given by the phase difference between each microphone pair and defines a steered response power function for a given location $l = [x, y, z]^T$ [5]. [22] performs a comparison of the two methods and finds that the computational load for *SRP-PHAT* is 3-15 times heavier than *MUSIC* in the *best* case. However, when computing the correlation matrix for the *MUSIC* algorithm, an average across multiple consecutive audio frames must be taken to ensure stability, which incurs a latency of up to half a second. Thus, even *MUSIC*, the fastest running algorithm, is unfit for real-time SSL. Moreover, these SSL algorithms sometimes perform poorly with an indefinite number of speakers and complex sound environments, especially if the subjects are moving. **Real-time sound source localization and separation comprise the second half of the challenges.**

III. Proposed Solutions. We present Acuity, a real-time system creating high-fidelity digital twins, jointly tackling the two challenges mentioned above. **In the visual realm**, Acuity leverages multi-resolution point cloud processing to meet real-time constraints on segmenting human subjects from high-resolution point clouds. **In the acoustic realm**, Acuity achieves real-time sound source separation by fusing the localization results (subject azimuths) from the vision pipeline with state-of-the-art SSS algorithms.

The **vision pipeline** of Acuity firstly generates colorized point clouds using multiple LiDAR cameras viewing the scene from different perspectives. Next, it segments out the point cloud of each human subject. Our key observation is that while a dense point cloud is necessary for realistic human perception, clustering and background subtraction algorithms are capable of efficiently extracting vital localization data (bounding box and centroid) from a coarse-grained point cloud. Thus, we propose a double background-subtraction pipeline (detailed in Section 3.3). The pipeline aggressively downsamples the input point clouds and then applies clustering and background subtraction to estimate a course boundary for each human subject. The pipeline then filters the original point cloud with these boundaries, drastically reducing the input space, and performs background subtraction again to isolate a fine-grained point cloud for each individual. We obtain these point clouds from multiple angles and aggregate them to create a comprehensive point cloud encompassing the entirety of the subject. The visual components of Acuity are capable of running at 25 frames per second while processing far denser input than learning-based methods. With a maximum latency of around 70 ms, these processed point clouds are well suited for applications in conferencing or presenting.

Acuity leverages *audiovisual sensor fusion* to address the challenges **in the acoustic realm**. We notice that SSL is the critical path in the audio processing pipeline. This module is inserted only to provide priors about the angle-of-arrival of the voices so that

we can perform the separation. Our intuition is that we can bypass SSL by leveraging the vision pipeline to guide the acoustic pipeline. As the by-product of the visual "double-subtraction" pipeline, we can obtain objects' precise locations at very low latency. With a tracker to ensure stability, the visual pipeline informs the acoustic pipeline about the number of subjects and their azimuths, which guides the beamforming of the microphone array to achieve clear sound source separation. Thus, the combination of LiDAR with directional audio enables SSS to be performed in real time while simultaneously enriching the scene by providing both audio and visual information. Our evaluations show that Acuity can stream separated audio from multiple individuals with a latency of less than 70ms and an average throughput of over 25 fps while achieving a superior voice quality compared with state-of-the-art separation pipelines. Our key contributions are summarised as follows:

- ***Efficient Processing of Detailed Point Clouds:*** Through the utilization of a double-background subtraction pipeline that processes the point cloud at multiple resolutions, high-fidelity point clouds can be isolated in real-time. Avoiding the use of neural networks allows for high-resolution output from lightweight systems lacking discrete GPUs.
- ***Real-Time Sound Source Separation:*** Utilizing *multimodal sensor fusion* with the LiDAR localization information as the prior circumvents the need for costly SSL, allowing directional audio to be streamed in real-time. Directional audio is key in isolating speakers, especially in situations where multiple individuals speak at similar volumes.
- ***Open-source Implementation:*** To boost reproducible research, the source code of Acuity is available at https://github.com/nesl/Acuity.

2 RELATED WORK

Point Cloud Segmentation and Classification: Object detection and localization within point clouds is popularly done through the use of neural networks, as knowledge of the location and class of obstacles is paramount in the target application of autonomous vehicle navigation [56]. These neural networks represent point clouds in the form of points [33, 39, 55], voxels (or pillars) [24, 27, 54, 58], projected 2D views [4, 15], and graphs [41]. These works have proposed effective neural architectures for feature extraction, region proposal, and point cloud classification. Ultimately, they output the bounding box and the classification of each identified subject in the scene. However, hyper-realistic digital twins have different requirements in terms of point cloud processing compared with autonomous driving. The former focuses more on acquiring dense point clouds that are *fully* isolated from the background while the latter emphasizes the ability to know the class and location of a subject within a sparse point cloud. Often trained upon sparse point clouds like the KITTI dataset [14], the aforementioned neural models may not generalize well on a dense, close-range point cloud for digital twin creations, owing to the distribution discrepancy between the training and the testing data (also known as the domain shift problem). Furthermore, these networks may struggle with inference time when applied on dense point clouds. For example, PointNet [33] is evaluated to process approximately 1 million points per second [33], which if constrained to a system operating at 30 frames per second, results in each frame containing approximately 30,000 points. In comparison, the Intel Realsense Camera outputs over 300,000 points per frame when running at the lowest 640×480 resolution. Some works such as [37] propose using skeleton estimation and avatar fitting to generate a representation of human subjects in the virtual space, but these avatars are confined to predefined models that lack details.

Sound Source Localization: Since computationally intensive SSL becomes a bottleneck to performing real-time SSS, many efforts have been made to enhance the speed of SSL. [23] broke with the traditional approach of signal processing methods, proposing a real-time CNN that can run on resource-constrained systems. The model was trained on a dataset of simulated room impulse responses. [29] uses the unique frequency characteristics of a voiceless syllable to filter out the noise and perform real-time localization. [16] and [32] both optimize traditional signal processing algorithms to achieve real-time localization. However, these algorithms encounter difficulties where the number of sound sources is unknown. *Liu et al.* proposed utilizing multimodal fusion of camera and microphone data to perform sound source localization in real-time for the purpose of improving hearing aids [25]. The algorithm used for sound source separation utilizes the angle of arrival and distance of the speaker to isolate subject sound via beamforming. However, the subject is localized in *Liu et al.* by analyzing certain features in the subject's face and extrapolating that to a 3D position. While the study accurately performed sound source separation, even on mobile subjects, it cannot handle multiple sound sources. Furthermore, the requirement that all subjects must face the camera also constrains the system's robustness.

3 SYSTEM DESIGN

3.1 General Overview

Acuity consists of three main stages, *initialization & point cloud generation, point cloud processing, and audio processing*. Figure 1 depicts the overall flow of the system and its constituent components.

(1) ***Initialization and Point Cloud Generation:*** The startup code initializes the camera and depth streams, beginning data transfer from the LiDAR to the edge server. Upon receiving the data from both streams, the color pixels are mapped to the depth values, building an aligned colored point cloud.
(2) ***Point Cloud Processing:*** The colored point cloud built in the previous step then passes through the "double background-subtraction" pipeline. During this pipeline, we localize the centroid of the subject and transform the (x, y, z) values of the point cloud from their default LiDAR frame coordinates to a set of global coordinates. The source localization information is transmitted through a Publisher/Subscriber model. Once the transformed isolated point cloud is obtained, it likewise broadcast through the same model.
(3) ***Audio Processing:*** The audio processing stage involves both the acquisition of sound data from a microphone array and the subsequent audio processing. The audio processing pipeline receives the localization data from the Point Cloud Processing stage via the Publisher/Subscriber mechanism mentioned previously, utilizing it to perform SSS without running expensive SSL.

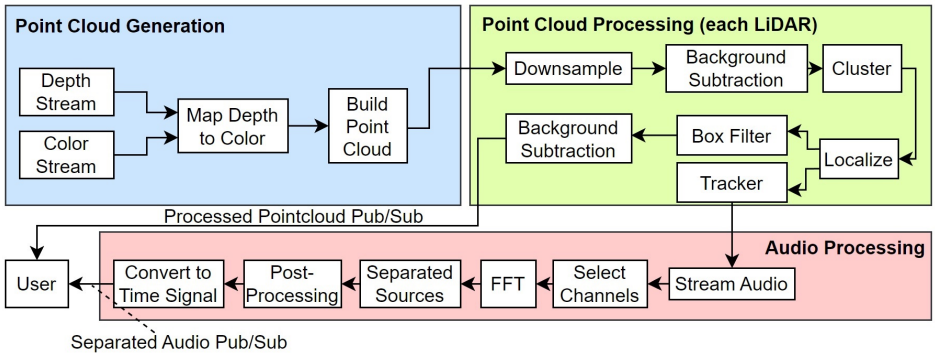

Figure 1: Acuity system overview

3.2 Initialization and Point Cloud Generation

3.2.1 ***Initialization.*** In Acuity, all the sensor drivers and data processing services run on an edge server, which is a mini computer without any special hardware accelerators. Before the system begins to produce output, several factors must be initialized for proper functionality. *First*, Acuity initializes the infrastructure for computation and sensor information exchange. Acuity employs a Publisher-Subscriber (pub-sub) model where the sensor drivers and the processing algorithms are implemented as nodes, and the input/output communications between these nodes are implemented as topics. Nodes can publish raw sensor data or processed output into a number of topics, which can then be subscribed by other nodes as input. This pub-sub model ensures the scalability of Acuity where a network of multimodal sensors and computers can communicate with each other easily to form a comprehensive view of the scene. Thus, Acuity first initializes the topic message broker (which handles the pub-sub mechanism) and all computation nodes. *Next*, the camera and depth streams of the LiDAR camera are initialized with proper parameters, beginning the streaming of sensor data. The camera stream transmits at a resolution of 960×540 while the depth stream runs at a resolution of 1024×768, both with a framerate of 30 fps. The resolution was experimentally adjusted to allow for the fastest runtime with reasonable pixel density.

3.2.2 ***Camera and Depth Stream Alignment.*** The depth sensor generates a depth frame, an image where each pixel is represented by a 16-bit depth value. The depth frame can be processed to form a 3D point cloud. For vision sensors like LiDAR/cameras, projection from a world coordinate (x, y, z) to pixel (u, v) is given by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & ppx \\ 0 & f_y & ppy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (1)$$

In the equation above, r represents the rotation, and t represents the translation. These extrinsic parameters are decided by the placement of the camera. f_x, f_y refer to the image focal lengths, s the axis skew (usually $s = 1$), and ppx, ppy refer to the principle axis offsets. All of these intrinsic parameters are pre-determined by the manufacturing of the camera. Equipped with Equation 1, it becomes trivial to map a known pixel $[u, v]^T$ with depth z to a point $[x, y, z]^T$ in the 3D space.

However, the point cloud has to be properly colorized to generate a vivid representation of a 3D human body. In Acuity, the intrinsic parameters and the relative location (i.e., relative rotation and translation) between the LiDAR and camera are all known. Thus, with Equation 1 and nearest interpolation, we can colorize the points with the RGB frame from the camera. At the end of this step, we obtained a colored high-resolution 3D point cloud

$$P_i^{high\_res}(n) = \{(x_1, y_1, z_1, r_1, g_1, b_1), (x_2, y_2, z_2, r_2, g_2, b_2), ..., (x_n, y_n, z_n, r_n, g_n, b_n).\} \quad (2)$$

3.2.3 ***Establishing a Global Coordinate Frame.*** Transforming to common global coordinates via a transformation matrix not only allows for a seamless combination of point clouds from different sensor angles but also simplifies audio processing by ensuring all source coordinates are in the same reference frame.

To combine point clouds from multiple LiDAR cameras and calculate the direction of arrival for the microphone array, we must establish mappings between the LiDAR local coordinates and a global coordinate system. We utilize *AprilTags* [52] for this purpose, which consist of patterned black and white squares that allow a camera to localize itself within the AprilTag's pre-established coordinate system. This system is often referred to as the "world" or "global" coordinate frame. Obtaining the transformation matrix between the camera and world frames requires only the intrinsic parameters of the camera and the dimensions of the printed tag. In Equation 1, the intrinsic parameters are given by the camera manufacturer, and all we need to estimate are the extrinsic parameters (r's and t's) which have 6 degrees of freedom (3 for rotations and 3 for translations). When the AprilTag is placed within the scene, we take a single frame from the video stream and detect the tag ((u, v)'s) in this image. With the dimension of the tag (x, y, z)'s known, Acuity forms a group of simultaneous equations and solves for the extrinsic parameters. For the first few iterations of the system, the camera resolution will be increased to 1920×1080 to capture the best shot of the AprilTag, after which it is lowered to the standard 960×540 resolution after the matrix is computed. This transformation matrix is essentially another extrinsic parameter matrix as it performs an identical task of mapping between two reference frames. At the end of this process, we acquire the transform matrix that maps a given point in the LiDAR coordinate system to the global system established by the AprilTag. If multiple LiDAR cameras are utilized, this process is repeated for each camera.

3.3 Point Cloud Processing

Volumetric capture is critical in order to generate visual digital twins of human bodies that look realistic from all perspectives in the virtual scene. In Acuity, we apply a point-based method, i.e., the vision-based pipeline takes in a point cloud from the LiDAR as input, outputting a point cloud in global coordinates containing only the points that belong to the human subjects. While some end-to-end neural structure [33, 58] has been proposed to perform this task, such networks have high computational overhead and cannot process high-quality, dense point clouds in real-time, especially on edge servers lacking neural accelerators.

Since we are instrumenting the environment with sensors in Acuity, we can reduce the computation overhead by collecting a reference point cloud without human subjects and applying point-cloud subtraction. However, a trivial direct subtraction does not work well since LiDAR point clouds are noisy with the amount and location of detected points varying frame-to-frame. Thus it is necessary to perform clustering and identify clusters with a reasonable amount of points as surrogates of human bodies. This subtraction and clustering, however, can still be computationally expensive given the large number of points generated by the indoor LiDAR.

To address this challenge, we propose a "double subtraction" pipeline that processes the point cloud with different resolutions. The key idea is that, while realistic human perception requires a high-resolution point cloud, subtraction and clustering can achieve successful results with point clouds of a much lower resolution. We can perform these expensive operations on low-resolution point clouds and apply the inferred object bounding boxes to the high-resolution point cloud and remove irrelevant points. With the search space drastically shrunk, and we can now apply the background subtraction again to obtain an isolated high-quality point cloud for each individual. The aforementioned pipeline also outputs the centroid of each subject in global coordinates. Algorithm 1 depicts the overall structure of the pipeline, while Figure 2 provides a visual depiction of the point cloud at each step. We will provide more detailed elaboration in the next few paragraphs.

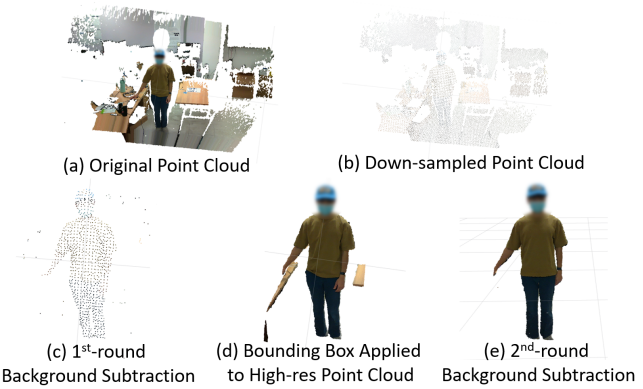

Figure 2: Visualization of the point cloud at each stage in the point cloud processing pipeline

3.3.1 ***Background Subtraction Basics.*** The key functionality used to separate the subject from the background is the background subtraction algorithm, which works as follows: A reference point cloud $P_r^{high\_res}$ at a particular time instance is stored in the memory. The reference point cloud captures the scene without human subjects, containing static objects such as walls and furniture. During runtime, when a new LiDAR point cloud arrives, the background subtraction function returns all points that differ between the two, effectively "subtracting" all points that remain the same. Thus, once a subject enters the scene, the function perceives distinct points from the background and returns only the subject's point cloud. However, background subtraction is only a piece to the puzzle, as not only does it leave some residual background points, but the efficiency drops significantly with larger point clouds (see Sec. 5.2.2 for results). This drawback drives the need for preprocessing for the system to perform in real-time.

Algorithm 1: Point Cloud Processing Pipeline

Input: colored point cloud

$$P_i^{high\_res}(n) = \{(x_1, y_1, z_1, r_1, g_1, b_1), (x_2, y_2, z_2, r_2, g_2, b_2), ..., (x_n, y_n, z_n, r_n, g_n, b_n)\}$$

Reference point cloud $P_r^{high\_res}(n)$, Transform matrix $\boldsymbol{M}$

Output: isolated colored point cloud of subjects

$$P_o(n) = \{(x_1, y_1, z_1, r_1, g_1, b_1), (x_2, y_2, z_2, r_2, g_2, b_2)...(x_n, y_n, z_n, r_n, g_n, b_n)\}, \text{ with } P_o(n) \subset P_i^{high\_res}(n)$$

1 $P_i^{low\_res}(n) \leftarrow DownSample(P_i^{high\_res}(n))$
2 $P_r^{low\_res}(n) \leftarrow DownSample(P_r^{high\_res}(n))$
3 $P_s \leftarrow BSB(P_i^{low\_res}, P_r^{low\_res})$; /* Background Subtraction */
4 $\boldsymbol{CL_l} = \{Cl_0, Cl_1...Cl_j\} \leftarrow Cluster\{P_s\}$; /* Extract j clusters for human subjects in P_s */
5 $\boldsymbol{CL_w} \leftarrow M \times CL_l$; /* Convert coordinates of each cluster to world coordinates with transform matrix M */
6 $\boldsymbol{C_w} = \{C_0, C_1...C_j\} \leftarrow Centroid(\boldsymbol{CL_w})$; /* Obtain centroid of each cluster */
7 $\boldsymbol{BB_w} = \{BB_{w0}, BB_{w1}, ...BBw_j\} \leftarrow BBox\{\boldsymbol{CL_w}\}$; /* Find bounding box of each cluster */
8 $\boldsymbol{ID} \leftarrow KalmanTracker(BB_w, C_w)$; /* Track the bounding boxes */
9 $\boldsymbol{Obj} \leftarrow ID \oplus BB_w \oplus C_w \oplus Angle(C_w)$
10 $Publish(\boldsymbol{Obj})$; /* Inform the audio processing pipeline of the locations of human subjects */
11 $P_f = \{(x, y, z, r, g, b) \in P_i^{high\_res} | (x, y, z) \in \boldsymbol{BB_w}\}$; /* Filter input point cloud extracting all points in bounding boxes */
12 $P_f^{ref} = \{(x, y, z, r, g, b) \in P_r^{high\_res} | (x, y, z) \in \boldsymbol{BB_w}\}$; /* Filter reference point cloud */
13 $P_b \leftarrow BSB(P_f, P_f^{ref})$; /* Perform second round of Background Subtraction */
14 $P_o \leftarrow M \times P_b$; /* Transform pointcloud to world coordinates */

3.3.2 ***Downsampling and the First Background Subtraction.*** Given that background subtraction performs poorly on large input point clouds, we must reduce the point cloud size to meet real-time constraints. After designating a reference point cloud, we store it for future use in background subtraction. We also create and store an aggressively downsampled version of that point cloud $P_r^{low\_res}$. The high-resolution input point cloud $P_i^{high\_res}$ is also aggressively downsampled to become $P_i^{low\_res}$. After this step, we have two versions (high-resolution, low-resolution) of the reference point clouds and two versions of the current input (Alg. 1 Line 1-2).

The first round of background subtraction takes in the *sparse* version of the current input point cloud and the downsampled reference (Alg. 1 Line 3). Performing background subtraction between the two point clouds provides a very coarse-grained representation of the subject, but with minimal latency. However, while all the background points are largely removed, we lack knowledge about the location and number of subjects in the scene. To provide this information, Acuity then performs clustering on the subtracted point cloud. The clustering algorithm segments our point cloud into distinct subjects where only clusters with sufficiently large numbers of points are kept. The clustering step helps us eliminate any residual background points as outliers. Nextly, the clusters CL_l are all transformed into the world coordinate system established by the AprilTag. The transformed clusters are denoted as CL_w. Afterwards, we can compute the centroid C_w and bounding box BB_w of each subject (Alg. 1 Line 4-7). While this stage does not yield point clouds of satisfactory quality, it provides critical information about the location of the subject that we can exploit in the acoustic pipeline.

3.3.3 ***Bounding Box Tracking.*** To consistently associate an identified source in the visual domain to a sound source in the audio domain, there must be a concept of *persistence*, where the same subject across multiple frames must be recognized as such. Tracking allows for applications such as transcription or selectively adjusting the volume of a given individual. Furthermore, tracking also introduces greater stability where subjects who disappear briefly continue to broadcast information for a short period. The tracking process works in two stages: *First*, a *Kalman Tracker* tracks the history of the bounding box locations and makes predictions. To improve the efficiency of the Kalman-based tracking algorithm, we first reduce the complexity to a two-dimensional case by taking the bird's-eye view. The Kalman Filter predicts the locations of future bounding boxes from existing position measurements and velocity calculation, effectively tracking the locations of the existing bounding boxes in the scene. *Then*, the *Hungarian Algorithm* associates the predicted bounding boxes with the detected bounding boxes from the current LiDAR point cloud, creating the concept of *persistence* between frames. The tracked individuals are assigned ID's through the Hungarian Algorithm that persist until the individual leaves the scene. In the event of a frame drop or brief occlusion of the subject, the tracker will persist for a maximum of 15 frames before deleting the subject from memory. Now, we can concatenate the object ID, bounding boxes, centroids, and calculated object azimuth angle, publishing them as a topic (Alg. 1 Line 8-9). This information will be subscribed by the audio processing pipeline to guide the sound separation algorithm.

3.3.4 ***Filtering and the Second Background Subtraction.*** Given the bounding box around the subject, we can utilize a *box filter* on the *original, high resolution* point cloud. A box filter passes though only the points within a specified x, y, z range, which is defined by the bounding boxes obtained in the first background subtraction (Alg. 1 Line 11-12). This isolates the subject with the exception of a few background points that are within the bounds of the box filter. Employing a *second round* of background subtraction eliminates those residual points (Line 13), ultimately resulting in only the subject's points remaining. Since there are only a small number of background points due to filtering, the artifact points from background subtraction noise are still present but not numerous enough to be noticeable. Thus no clustering is needed for the second round of background subtraction. This filtering and background subtraction process is repeated for each subject identified, after which their point clouds are merged into one complete cloud P_b, transformed to world coordinates as P_o, and is finally output for visualization (Line 14).

The novelty of the system lies in the ability to produce *high quality* point clouds at a *high speed*. Since background subtraction and clustering both scale incredibly poorly with size, performing them on a heavily downsampled point cloud allows us to extract key information without sacrificing time. Then, the information obtained from the downsampled input can be utilized to extract a subset of points primarily around the subject from the original point cloud. This point cloud, while high quality, contains far fewer points due to the constraint imposed by the bounding box, so the search space is vastly reduced for the second stage of background subtraction.

3.4 Audio Processing

In the previous section, we discuss how we achieve high-quality human subject point cloud segmentation in real-time. Authentic digital twins also require transmitting high-quality audio of the target subject with minimum interference from other sound sources. To achieve this goal, we present the audiovisual sound source separation algorithm depicted in Algorithm 2.

Algorithm 2: Audiovisual Sound Source Separation Pipeline

Input: $C \times T$ sized short audio frame
$$\boldsymbol{x}(t) = [x_1(t), x_2(t), ..., x_C(t)]^T$$
N sized Object Information $\boldsymbol{Obj}$, where
$$Obj_i = \{ID, (x, y, z), Angle, Elevation\}$$
Output: $N \times T$ sized separated audio for each subject
$$\boldsymbol{y}(t) = [x_1(t), x_2(t), ..., x_m(t)]^T$$

1 $\boldsymbol{X}(k) \in \mathbb{C}^{C \times T} \leftarrow MultiFFT(\boldsymbol{x}(t))$
2 $\boldsymbol{\theta} \leftarrow ExtractAngle(Obj)$; /* Extract the azimuth angle from subscribed object */
3 $\boldsymbol{Y}(K) \in \mathbb{C}^{N \times T} \leftarrow GHDSS(\boldsymbol{X}(k), \boldsymbol{\theta})$; /* Separate the audio for human subjects based on their signal AoA */
4 $\boldsymbol{N}(k) \leftarrow HRLE(\boldsymbol{Y}(K))$; /* Noise estimation */
5 $\tilde{\boldsymbol{Y}}(k) \leftarrow SpecSub(\boldsymbol{Y}(K), \boldsymbol{N}(k))$; /* Spectral subtraction */
6 $\boldsymbol{y}(t) \leftarrow Synthesize(\tilde{\boldsymbol{Y}}(k))$; /* convert the signal from the frequency to time domain */

The audio processing pipeline begins with obtaining information from the microphone in the form of frames. Each frame has the form of a $\mathbb{R}^{C \times T}$ matrix, where C is the number of channels and T is the number of samples per frame. After a frame is generated and fed into the pipeline, a Fast Fourier Transform (FFT) is taken (Alg. 2 Line 1). The audio pipeline also subscribes to the sound source information provided by the point cloud processing pipeline(Alg. 2 Line 2). With this subscription, we fuse the visual information with the audio sensor so that the audio pipeline is informed of the location (azimuth) of human subjects. Then we can perform SSS without the expensive SSL.

With the frequency domain frame and the directions of arrival for each audio source, the pipeline performs sound source separation and outputs single channel (mono) audio for each source in the frequency domain (Alg. 2 Line 3). We adopt the state-of-the-art Geometric High-order Dicorrelation-based Source Separation (GHDSS) algorithm depicted in [8] for this SSS task. For clarity, we give a brief summary of the GHDSS algorithm here. The received multichannel microphone array signal $\boldsymbol{X}(k) \in \mathbb{C}^{C\times T}$ in the frequency domain is decided by the N source sources $\boldsymbol{S}(k) \in \mathbb{C}^{N\times T}$, channel response $\boldsymbol{H}(k) \in \mathbb{C}^{C\times N}$, and noise $\boldsymbol{N}(k) \in \mathbb{C}^{C\times T}$, i.e.,

$$\boldsymbol{X}(k) = \boldsymbol{H}(k)\boldsymbol{S}(k) + \boldsymbol{N}(k). \tag{3}$$

The goal of GHDSS is to find an optimal separation matrix $\boldsymbol{W} \in \mathbb{C}^{N\times C}$ to separate the signal into

$$\boldsymbol{Y}(k) = \boldsymbol{W}(k)\boldsymbol{X}(k), \tag{4}$$

where $\boldsymbol{Y}(k) \in \mathbb{C}^{N\times T}$ approximates $\boldsymbol{S}(k)$. The signal separation problem is subject to two constraints: *first*, the separated signal $\boldsymbol{Y}(k)$ should be high-order decorrelated. More specifically,

$$diag(\boldsymbol{R}^{\phi(y)y}) \coloneqq diag(E[\phi(\boldsymbol{Y})\boldsymbol{Y}^{\boldsymbol{H}})]) = \boldsymbol{0}, \tag{5}$$

where E is the mathematical expectation over time, H is the hermitian transpose, and ϕ operator is defined by

$$\phi(\boldsymbol{Y}) = [\phi(Y_1), \phi(Y_2), ..., \phi(Y_N)]^T, \tag{6}$$

$$\phi(Y_m) = tanh(\sigma|Y_m|)e^{\angle(Y_m)}, \tag{7}$$

and σ is a scaling factor. *Second*, the line-of-sight (LoS) components should be separated with no distortions, i.e.,

$$\boldsymbol{W}\boldsymbol{H_D} = \boldsymbol{I}, \tag{8}$$

where $\boldsymbol{H_D}$ is the transfer function of the LoS sounds. With these two constraints, GHDSS optimizes $\mathbf{W}$ for the following problem using complex gradient descent,

$$\boldsymbol{W_{opt}} = \arg\min_{\boldsymbol{W}} \alpha \sum_{i\neq j} |R_{i,j}^{\phi(y)y}|^2 + \beta||\boldsymbol{W}\boldsymbol{H_D} - \boldsymbol{I}||^2. \tag{9}$$

The last piece of the puzzle is how to obtain the transfer function of the LoS sounds $\boldsymbol{H_D} \in \mathbb{C}^{C\times N}$. Prior to running the system, we measure these transfer functions by playing a time-stretched impulse (TSP) sound file using an external speaker at intervals of 5° around a 1 m radius circle and measure the microphone array's response. This measurement needs to be done only once in advance since the location of the microphone array is static. During the runtime, the point cloud pipeline informs the audio processing pipeline about the subjects' azimuth angles $\theta_1, \theta_2, ..., \theta_N$. We can then approximate $\boldsymbol{H_D}$ by combining the frequency response from these θ's. Finally, GHDSS outputs separated audio spectrograms $\boldsymbol{Y}(k) = \boldsymbol{W_{opt}}(k)\boldsymbol{X}(k)$.

This outputted audio spectrogram $\boldsymbol{Y} \in \mathbb{C}^{N\times T}$ can be further enhanced through *post processing* (Alg. 2 Line 4-5), where we remove background noise and suppress interference between sound sources using spectral subtraction [47]. From a high level, this step estimates the spectrogram of the noise and interferences, subtracting it from the spectrogram of the separated sound. The pipeline ends after converting the signal back to the time domain (Alg. 2 Line 6) and publishing the separated audio stream for each person.

The key contribution of this pipeline lies in the *multimodal fusion* of audio and visual data with the localization provided by LiDAR acting as the prior to this system. The LiDAR localization is provided at a rate of 25 times per second from the point cloud processing pipeline, allowing knowledge of the precise location of the individual and results in accurate beamforming even in conditions where the angle rapidly changes. For example, in the case where a person is moving around, an SSL+SSS-based method has to perform the expensive SSL multiple times (at the cost of high latency), causing significant delay in the audio streaming. The method used in Acuity, however, can keep track of the azimuth of the target speech without noticeable overhead.

At this point, Acuity is well-equipped to provide a hyper-realistic experience in conferencing. Acuity outputs the combined colored point clouds from multiple LiDAR cameras containing only the human subjects with the separated audio of each individual. It shines in situations where an individual is joining a hybrid remotely, as the remote person can both see and hear all physical participants, even when multiple discussions occur at once.

4 IMPLEMENTATION

4.1 Hardware

4.1.1 ***Sensing Hardware.*** To construct a *colored* point cloud, we choose to use an *Intel Realsense L515* LiDAR camera, which contains both an RGB camera and a LiDAR depth sensor. It can deliver a maximum depth resolution of 1024×768 and a color resolution of 1920×1080, both at 30 frames per second [20]. This Intel LiDAR camera specializes in outputting high-density point clouds over a narrow field of view, contrasting with LiDAR sensors from Ouster or Velodyne that provide 360° coverage at hundreds of meters at the cost of resolution. Another key feature is the provided *Librealsense SDK* [19] contains in-built functionality for aligning the color and depth streams, creating a well-aligned colored point cloud with relative ease. Note that Acuity's algorithms are capable of operating upon any colored pointcloud, independent of the type of LiDAR sensor used.

Microphone arrays are the ideal sensor to capture audio for SSS purposes, as the circular arrangement of the microphones allows for accurate estimation from all angles. We will be utilizing the *ReSpeaker Mic Array v2.0* manufactured by Seeed Studio, which contains 4 microphones each separated by 90 degrees. The ReSepeaker is an affordable microphone array that interfaces well with many devices. Once again, Acuity is capable of operating correctly with a wide variety of microphone arrays, including those with more microphones and greater precision.

4.1.2 ***Computing Hardware.*** To test the effectiveness of Acuity on edge servers lacking a GPU, the system was controlled by a *Intel NUC 12 Extreme NUC12DCMi9* Mini Desktop. While larger than traditional NUC mini PCs, its compact form factor still allows for portability. The system lacks any special hardware acceleration, and all the computation occurs on the CPU. Both LiDAR cameras were connected to the NUC, as well as the ReSpeaker microphone array.

4.2 Software

4.2.1 ***Sensor Information Exchange.*** We utilized *Robot Operating System (ROS)* as the Publisher/Subscriber Model. The visual

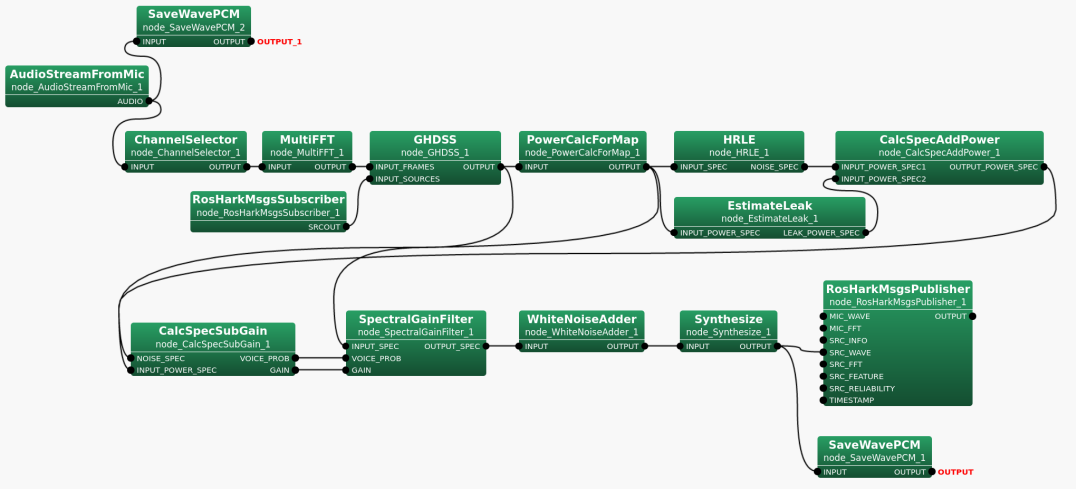

Figure 3: Depiction of the structure of the SSS pipeline

pipeline's two cameras publish the separated clouds P_o to their individual topics (/cameraX/subtracted/clouds), along with the information of detected subjects Obj (/cameraX/sources). The audio pipeline subscribes to the localization information Obj, outputting the separated audio to the topic of /Hark/Audio. An end user of this system can subscribe to the separated audio and isolated point clouds via these topics and use the ID information to associate a given point cloud with its audio.

4.2.2 ***Point cloud computation.*** We leveraged the rich functionality of the Point Cloud Library (PCL) to process our LiDAR point clouds. Not only does it provide convenient methods to alter point clouds, but it also seamlessly integrates with both ROS and the Realsense Library. The conversion between ROS point cloud objects and PCL point clouds is trivial. The key PCL functions used in the system involve background subtraction, clustering, and filtering.

Background subtraction is implemented in PCL through the use of two octrees acting as buffers. Octrees are a tree-like structure where each parent node has 8 children nodes, allowing it to recursively partition a given 3-D space into 8 regions until a minimum region size is attained. The octree structure stores the points in a manner conducive to efficient searching, as it exploits spatial locality by grouping similar points together. One buffer (octree) stores the reference point cloud, and the other stores the current point cloud. When background subtraction is performed, the corresponding points in the two octrees are compared, returning a vector consisting of the points that differ.

We utilize *Euclidean clustering* from the PCL Library to separate out distinct subjects. The process begins by storing the point cloud in a *k-d tree*, which is a binary search tree where each node is a k-dimensional point in space. The structure of a k-d tree allows for rapid nearest-neighbor searches. We establish a cluster tolerance distance of 15 cm, where neighboring points within 15 cm of the growing cluster will be added. Additionally, to address the issue of residual points arising from noise in background subtraction, we establish a minimum cluster size of 300 points. Clustering results in S independent point clouds, where S is the number of subjects present in the scene.

In order to filter the original point cloud using the bounding box, we use a *Box Filter*. The box filter simply iterates through all the points, returning a collection of only the ones that lie within the specified $(x_{min}, y_{min}, z_{min}) \rightarrow (x_{max}, y_{max}, z_{max})$ range.

4.2.3 ***Audio Computation.*** Audio Processing leverages the *Honda Research Institute Japan Audition for Robots with Kyoto University (HARK)* library for both acquisition of sound data from the ReSpeaker and subsequent audio processing.

HARK Overview: HARK is a highly modular audio processing library primarily focused on efficient audio processing in robotic applications. It is comprised of individual modules that each carry out a specific task, such as calculating an FFT or writing to a file. When chained together, they can perform a complete task such as SSS. The overall pipeline for SSS is depicted in Figure 3. Descriptions of its key nodes are as follows:

- ***AudioStreamFromMic:*** This node handles streaming multichannel audio from the ReSpeaker. The audio stream occurs at a rate of 16000 samples/second in frames of 512 samples, though each frame contains only 160 new samples. Each sample is a 16-bit integer value, and each frame is outputted every 10 ms.
- ***ChannelSelector:*** Selects the 4 channels corresponding to the 4 microphones, omitting two playback channels.
- ***MultiFFT:*** Performs a 512 point FFT on a given frame for each channel
- ***RosHarkMessageSubscriber:*** Subscribes to the source information published by the point cloud processing pipeline. The sources are in the form of *HarkSrc* objects, which contain the (x, y, z) position of the subject as well as the id, power, angle, and elevation.
- ***GHDSS:*** GHDSS performs geometric higher-order decorrelation among the signals, and isolates the sound signature from a specified direction of arrival. GHDSS takes two inputs: the frequency domain audio signal, and the audio sources received in the RosHarkMessageSubscriber node. To perform sound

source separation, GHDSS requires several transfer functions, which are found by analyzing the ReSpeaker Microphone's response to a time-stretched impulse (TSP) sound file played on an external speaker. To decrease the impact of noise on the system, the TSP signal is repeated 20 times, ensuring a high SNR by minimizing the effect of noise [10]. The repeated TSP signal is played at intervals of 5° around a 1 m radius circle on an external loudspeaker around the ReSpeaker Microphone. The audio at each angle is recorded through the ReSpeaker, and the transfer functions are computed through a *complex regression model* [9] using the provided *HarkTool5_GUI*. The computed transfer functions are used to initialize GHDSS parameters, allowing it to output the directional sound in the frequency domain for each source established by the LiDAR.

- ***Synthesize and RosHarkMsgsPublisher:*** Synthesize converts the signal from the frequency to time domain, and RosHarkMsgsPublisher publishes the audio streams to a ROS topic as a *HarkSrcWave* object. The object contains the ID and location of the source in the ReSpeaker coordinate frame, as well separated audio for the subject. An end user could subscribe to this message topic to receive the real-time audio of a given source.
- ***Post-Processing Nodes:*** Any node not explicitly mentioned is a post-processing node that enhances the quality of the main speaker and reduces background noise.

5 RESULTS

5.1 Experiment Setup

5.1.1 ***Sensor Placement:*** Figure 4 depicts the experimental setup. The Intel Realsense LiDAR cameras (1) and (2) are placed at opposite ends of the room on tripods, each connected to the edge server. The use of two LiDAR cameras allows the entire body to be captured, as each camera can only record from a fixed viewpoint. The point clouds from each will be transformed into a world coordinate system and combined by the edge server, creating a comprehensive point cloud of the entire subject. The global coordinate system is defined by an *AprilTag* (Sec. 3.2.3) placed upon the floor. The LiDAR cameras are viewing the scene at a downward angle to ensure individuals of all heights can be captured while also allowing a good shot of the AprilTag. The ReSpeaker microphone array (3) is placed at the center of the scene to ensure the best coverage. It is elevated on a table to be closer to sound sources and prevent interference from footsteps or reflected sound waves traveling across the floor.

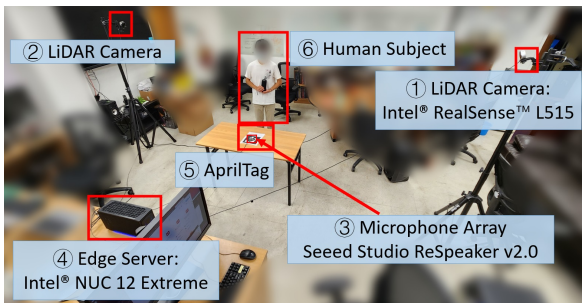

Figure 4: Experiment setup.

5.1.2 ***Experiment Scenarios:*** We choose to evaluate Acuity in two real-world situations: a conference meeting with three people seated around a table and a conversation between individuals as they walk in a predefined pattern (see Figure 5).

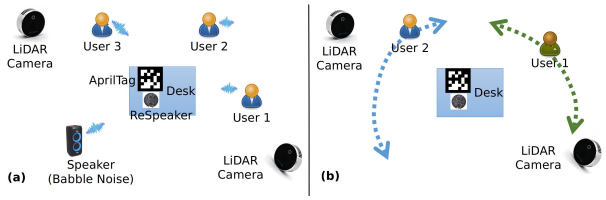

Figure 5: Experiment scenarios: (a) three people seated around a table (b) two people walking around.

- ***Scenario 1:*** This scenario simulated three individuals seated around a conference table, testing the ability of the point cloud processing pipeline to accurately isolate the point clouds despite occlusions from the chairs and tables. This setting frequently occurs in real life when a participant joins a physical group meeting remotely. Fig. 5(a) demonstrates the arrangement of the physical participants. We will test three situations: subjects speaking individually (Single Source), two of the three subjects speaking (Two Sources), and finally all three subjects (Three Sources) speaking at once. A speaker is placed near the table playing babble noise to simulate background noise.
- ***Scenario 2:*** Instead of placing the individuals in a fixed location, Scenario 2 tests the ability of the audio separation pipeline in conditions involving mobile sources. Fig. 5(b) depicts the experimental setup. During this setup, the azimuth of both speakers constantly changes. Source 1 is walking repeatedly along a line 1 meter away from the ReSpeaker at its closest point. Source 3 likewise performs a similar task, but on the other side of the ReSpeaker.

5.1.3 ***Reference Audio Clips:*** To ensure the consistency of audio between trials, we prerecorded 3 audio clips to represent the speech of 3 subjects. During the following experiments, human subjects carried portable speakers playing those clips to simulate talking. This controls for differences in speech volume between takes, and allows for the computation of SDR by providing an exact reference signal. The three clips consist of a 30 second recordings from Wikepedia articles detailing Blue Whales, Right Whales, and Dolphins. We will henceforth refer to the Blue Whale, Right Whale, and Dolphin audio clips as Sources 1, 2, and 3, respectively.

5.1.4 ***Evaluation Metrics:*** The following metrics are employed to evaluate the performance of Acuity.

- **Latency and Throughput Measurements**: The throughput of the system is determined by the number of LiDAR frames that can be processed in the point cloud subtraction pipeline in unit time. Thus, we measure the visual throughput by recording the time elapsed in the Point Cloud Processing Stage. The latency is obtained by adding the additional time needed to aggregate two point clouds from distinct angles. Note that the camera stream transmits at a resolution of 960 × 540 while the depth stream runs at a resolution of 1024 × 768 unless otherwise specified. On the

audio side, the latency is obtained by the time between an audio frame received from the microphone and the final separated audio.

- **Audio Separation Clarity**: While Signal to Noise ratio (SNR) is typically utilized to evaluate the strength of a given source, Signal to Distortion Ratio (SDR) is superior when dealing with blind sound source separation [51]. SDR (in decibels) is given by the following formula:

$$SDR = 10 \times log_{10} \frac{||s_{\text{target}}||^2}{||e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}||^2} \tag{10}$$

where $e_{\text{interf}}, e_{\text{noise}}, e_{\text{artif}}$ are the error terms for interference, noise, and artifacts, respectively. SDR provides a systematic way of quantifying the clarity of a reconstructed signal received from sound source separation. We will utilize the bss_eval Matlab toolkit to compute the SDR [12, 50].

- **Audio Separation Word Error Rate (WER)**: While SDR provides a metric for signal quality and clarity, it fails to provide any information regarding the *intelligibility* of the signal. To measure how easily the speech in the separated audio is understood, we run the separated audio stream through a text-to-speech neural network, computing the WER of the transcribed file. WER is measured as follows:

$$WER = \frac{S + D + I}{N} \tag{11}$$

where S is the number of substitutions, D the number of deletions, I the number of insertions, and N the total number of words. We utilized the *ConverseSmartly* speech-to-text service from Folio3 to obtain the transcripts of our audio files.

5.2 Point Cloud Pipeline Evaluations

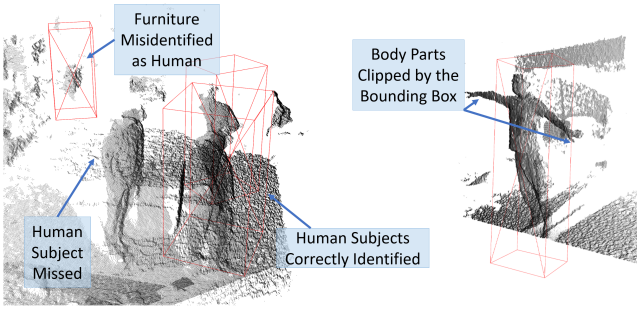

Figure 6: Failure cases of deep learning models on dense point cloud segmentation.

5.2.1 ***Failure Cases of Deep Learning Models:*** While some deep learning models are capable of running classification and localization on point clouds in real-time with GPU acceleration, their performance suffers when directly applied to dense indoor point clouds. We evaluated PartA2 [40], PointPillar [24], PV-RCNN++ [38], and SECOND [54] using their pre-trained weights. The first two models failed to generate meaningful results. While the latter two successfully identified a few human subjects, their performances were still unsatisfying. Figure 6 demonstrates the associated issues, with the neural network mistakenly classifying a desk as a human while passing over the actual human. Furthermore, the bounding box on the right not only includes the background points of the floor but also clips some of the subject's body parts. We will perform a detailed comparison of Acuity vs. learning-based methods in 5.2.3. These deep learning models were mostly trained on datasets collected with commercial LiDAR in open spaces, where the pointcloud is sparse. A high-quality dataset with tedious human labeling is necessary if we want to adapt these deep neural detectors for indoor dense point clouds. Acuity, on the other end, requires no model training at the cost of collecting a few background frames with no human subjects.

5.2.2 ***Latency Analysis:*** Figure 7 highlights the improvement of the Multi-Resolution Background Subtraction pipeline of Acuity compared to using a single resolution. In analyzing system throughput, we neglect the additional latency imposed by the final point cloud transformation and aggregation step, as it runs in parallel. With dense point clouds, background subtraction and clustering suffer greatly in run time, taking over 150 milliseconds (5 fps) for only a single subject and inflating to over 250 milliseconds (4 fps) for 3 subjects. In contrast, Acuity has a throughput of 30 frames per second (capped by LiDAR frame rate) for one subject, 27 fps for 2 subjects, and 20 frames per second for 3 subjects. The latency of Acuity includes an additional 15 millisecond delay from the combination of point clouds, resulting in a maximum latency of around 70 ms.

We also evaluate the performance of Acuity on lower end systems to demonstrate its viability over a wide range of systems. Since the visual pipeline of Acuity is the main bottleneck, we measured throughput in frames per second. We tested the visual pipeline on a *2015 Intel NUC i5-5250*, and a *2016 Intel NUC i7-6770HQ*. The *i5 NUC* processed each frame of a single-subject scene at 640x480 with an average time of 29.2 ms, and the *i7 NUC* achieved a processing time of 19.98 ms on a similar setup. Both machines achieve a single-subject throughput that is limited by the 30 fps limit of the LiDAR sensor, demonstrating Acuity's ability to run on lower-end systems.

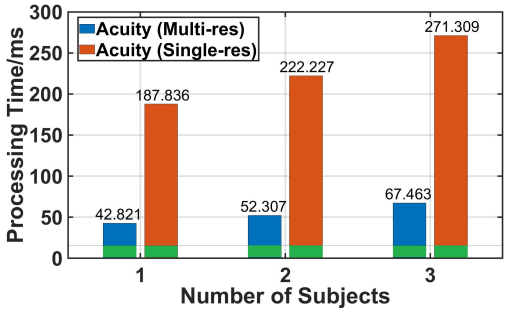

Figure 7: End-to-end latency of Acuity w/ and w/o multi-resolution pointcloud processing.

5.2.3 ***Comparison with Neural Networks.*** We showcase the advantages provided by Acuity by comparing it with two learning-based networks mentioned above, PV-RCNN++ and SECOND. We evaluate the *latency, average F1 score, and accuracy* from 1 to 5 subjects in the scene to test Acuity's performance in crowded scenes. Accuracy is given by $\frac{TP}{GT}$, and F1 Score is $\frac{TP}{TP+0.5\times(FP+FN)}$, where

	Acuity (Ours)			PV-RCNN++			SECOND		
Number of Subjects	Latency (ms)	Accuracy	Average F1 Score	Latency (ms)	Accuracy	Average F1 Score	Latency (ms)	Accuracy	Average F1 Score
1	34	100%	1.00	1584	53.30%	0.33	33	88.50%	0.6
2	54	100%	1.00	1635	60%	0.26	32	90.80%	0.78
3	30	100%	1.00	690	76%	0.38	31	94.40%	0.83
4	44	97.40%	0.99	691	68.80%	0.33	31	71.60%	0.63
5	61	93.30%	0.96	699	75%	0.37	31	92.80%	0.77
6	69	91.70%	0.95	693	70.80%	0.41	31	63.10%	0.61

Table 1: Latency and accuracy evaluations for the human subject detection of Acuity vs. PV-RCNN++ and SECOND. Note that the first rows were performed on the original resolution, while subjects three and more were performed on 640x480 resolution to measure the maximum throughput. Acuity outperforms the neural models in terms of detection accuracy within a reasonable latency budget.

TP, FP, FN, and GT are the True Postive, False Positive, False Negative, and Ground Truth bounding boxes.

Table 1 showcases that while Acuity's runtime increases with a greater number of subjects, the learning-based methods depend only on the density of the input point cloud. As a result, they perform the same regardless of the number of subjects. PV-RCNN++ operates over the raw point input, resulting in a latency of over half a second in all situations. This showcases the inability of some neural networks trained on sparse input to run efficiently on denser point clouds. SECOND, however, performs sparse convolution on the input data and runs at close to 30 frames a second even at at higher subject counts, outperforming Acuity. Acuity's framerate drops to 15 fps at 6 subjects, but outperforms SECOND in both average F1 score and average accuracy. In an application such as video conferencing, it is more vital to have the subjects be correctly identified and separated from the scene, making Acuity the best choice despite the diminished throughput with more subjects.

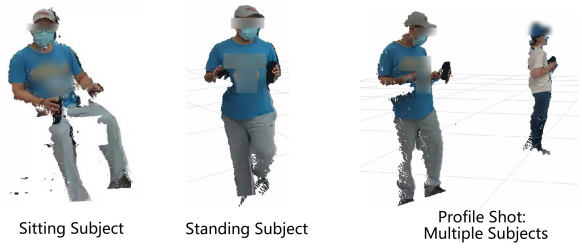

Figure 8: High-resolution point clouds separated by Acuity

5.2.4 ***Segmented Point Cloud Visualization:*** Figure 8 displays the separated point clouds by Acuity. Though the faces and identifying text on clothing are blurred for anonymity, one can observe that the point clouds are capable of capturing the realistic likeliness of the subject, from symbols on clothing to glasses. Furthermore, the set-up with multiple cameras allows good coverage of the subject, as shown by the profile shot where both subjects are viewed from the side.

5.3 Audio Pipeline Evaluations

5.3.1 ***Baselines:*** We evaluate the audio performance of Acuity against two state-of-the-art sound source separation models (SuDoRM-RF [46], DPRNN [28]) trained on the WSJ0-2 dataset. To handle the case of three speakers, we trained DPRNN on the VCTK dataset but were unable to do the same for SuDoRM-RF. Note that while we are comparing these neural networks with Acuity in terms of sound separation performance, they do not possess the full functionality of Acuity: apart from separating audios, the audiovisual fusion approach of Acuity also establishes a connection between people's point cloud and their speech, i.e., "who said what". This information is not available in those neural models.

5.3.2 ***Failure Case of SSL.*** Aside from execution latency issues, SSL also suffers from instability and robustness concerns. Figure 9 illustrates such complications, where despite there being two subjects continually speaking in the scene at $-90°$ and $90°$, SSL not only identifies "ghost subjects" at $0°$, but also loses track of the subject at $90°$. Even in the properly tracked source at $-90°$, the angle appears to vary by up to $20°$ from the baseline. In addition, using only audio to perform localization can falsely place an increased emphasis upon signal power, where only the top few sources with the highest power are considered. If the background noise is sufficiently loud enough, SSL will mistakenly associate the noise as a source.

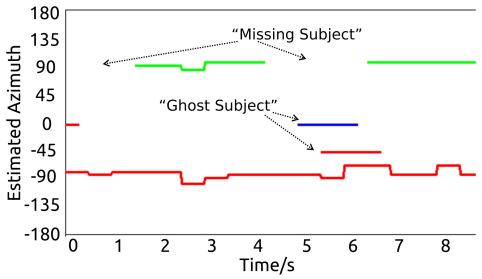

Figure 9: An example where SSL failed to reliably locate sound sources.

5.3.3 ***Scenario 1: Multiple Sources with Background Noise:*** We evaluated Acuity's ability to accurately reconstruct mutiple sound sources in a noisy environment by utilizing a speaker playing *babble noise*, which consists of overlapping human speech. We utilized the setup depicted Figure 5(a), placing our subject + speaker combinations around the ReSpeaker Microphone. The experiment consisted of three configurations of 12 trials each: Single Source

		SDR Gain (db)			Word Error Rate (%)			
		Acuity	DPRNN	SuDoRMRF	Raw	**Acuity**	DPRNN	SuDoRMRF
Single Source	Source 1	**1.92**	-1.803	-0.249	6.02	**1.95**	60.43	47.075
Two Sources	Source 1	**3.194**	-1.094	0.296	79.02	**5.08**	95.5	95.95
	Source 2	**10.822**	1.107	0.104	87.82	**5.34**	78.13	75.4
Three Sources	Source 1	**5.309**	0.2899	-	100	**15.49**	100	-
	Source 2	**7.025**	-0.192	-	100	**3.13**	92.1	-
	Source 3	**4.835**	-0.145	-	100	**28.425**	95.5	-

Table 2: Scenario 1 (Section 5.1.2) WER and SDR

(Source 1), Two Sources (Source 1, 2), and Three Sources (Sources 1, 2, 3). Each trial outputted the separated audio streams along with a raw, unseparated audio file directly from the microphone. The *SDR Gain* was obtained by taking the difference between the SDR values of the separated audio stream and the raw audio file. The WER was also measured for both the separated audio streams and the raw audio. The results are summarized in Table 2. We notice that Acuity not only provides the highest SDR gains but also yields considerable gains in WER across various configurations. Both DPRNN and SuDoRM-RF barely improve upon the raw microphone audio in SDR and WER, illustrating that current machine-learning SSS methods do not generalize well to real-life experiments where acoustic conditions and background noise vary greatly. Note that in the one source example, the raw microphone audio performs quite well as there is minimal interference from external sources, but it degraded rapidly as the number of speakers increased. For 3 speakers, the transcription service failed to provide any output, resulting in a 100% WER.

5.3.4 ***Scenario 1: Noise Suppression for silent subjects:*** In a real-world situation, not all the subjects within the scene will be talking at once. For example, if three subjects are present in the scene, but only one is talking, the two silent sources should attempt to minimize the audio of the vocalizing source. The visual-based source information of Acuity afford the unique advantage of conducting SSS on a silent subject, allowing for the suppression of undesired sound sources. To test this, we ran evaluations similar to Experiment 1, but always with three subjects regardless of the true number of speaking sources. Through this, we obtained a *negative SDR gain* signifying suppression of the external source. This was conducted with 1 vocalizing source (two silent sources), and 2 vocalizing sources (1 silent source), obtaining an average SDR interference suppression of **-4.285** and **-2.553**, respectively (the lower the better).

5.3.5 ***Scenario 2: Audio Separation for Moving Subjects:*** Scenario 2 (Figure 5 (b)) explored a situation involving two mobile sources, which adds an additional layer of complexity as the angle of arrival changes over time. Figure 10 illustrates that Acuity was the only system capable of handling mobile subjects, yielding positive SDR gain and maintaining a high word error rate gain. Both SuDoRM-RF and DPRNN recorded significant *negative* SDR gains, indicating a reduced audio quality. Additionally, the WER of SuDoRM-RF and DPRNN were likewise higher than the raw microphone audio, suggesting that the moving sound sources pose a challenge to existing learning-based systems, and these models distorted the audio without properly separating sound sources.

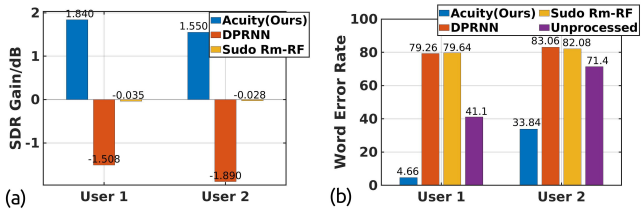

Figure 10: Audio Separation Results for Moving Subjects

5.3.6 ***Latency Analysis:*** The latency of Acuity is given by the time difference between the arrival of the audio data from the microphone and the outputted separated audio timestamp. Acuity processes 10 ms audio frames in a pipelined manner where the whole pipeline takes 30 ms to complete. Thus, Acuity is capable of real-time audio processing with a latency of 30 ms. The *Advanced Televisions Systems Committee* found that audio lagging video by up to 45 milliseconds is indiscernible, showing Acuity to be suitable for virtual conferencing [6]. However, if we attempt to provide the audio as chunks of the same duration (10 ms) into SuDoRM-RF or DPRNN, as we would if streaming from a microphone, the model fails to provide any meaningful result. This stems from the neural network requiring a certain amount of audio to extract useful features, as evidenced by DPRNN's minimum 8000 sample limit. We found that DPRNN and SuDoRM-RF both perform significantly faster than real-time, with DPRNN taking 0.75 seconds per 1 second of audio and SuDoRM-RF taking 0.3 seconds per 1 second of audio. Unfortunately, adopting these methods in a real-time system incurs an unpleasantly long buffering delay.

6 LIMITATIONS AND FUTURE WORK

In this section, we discuss the problems that have not been solved by Acuity and point out some future directions.

Real-time Point Cloud Streaming and Rendering. Acuity is a system on the sensing side of AR/VR conferencing. On the vision end, we mainly address the problem of real-time capturing and segmenting of high-resolution point clouds. However, such point clouds still need to be streamed over the Internet in real time. It is important to work on the compressing, representation, and streaming of the point clouds, where such things can be done in the form of points [26], meshes [7], or volumetric video [57]. Apart from streaming, the receiver end of AR/VR conferencing is also critical, where we need to efficiently render the representations of the subjects in real-time. One such way is using headsets or AR glasses. In recent years, 3D pads using light-field technologies [18] and holography have also been employed for AR/VR object display.

Collaborative efforts on the sensing, streaming, and rendering parts will constitute an immersive AR/VR conferencing experience.

Scaling Up Acuity. As a proof-of-concept prototype, Acuity employs one microphone array and two LiDAR cameras to capture a scene. To create a hyper-realistic digital twin of human subjects, the system needs to be scaled up where multiple microphone arrays and a distributed LiDAR camera system are employed. The scaled-up system will be a large sensor network with multiple edge servers to handle a larger amount of subjects. One of the open research questions on this front is the optimal placements of data and computation in distributed network settings. Also, it is possible to use reinforcement learning or other methods to investigate the best policy to decide when to conduct computations locally and when to offload the data to a server that is powerful and GPU-enabled.

Environmental Conditions. Though Acuity is capable of outperforming current state-of-the-art point cloud and audio separation algorithms, it comes with an additional cost of environmental dependence. The time stretched impulse response (Section 3.4) of a given room must be measured by analyzing the microphone signal at various positions to obtain a separation matrix. Acuity will still perform audio separation without this precise measurement, but will be limited in the quality of the separation. Further analysis can be undertaken to investigate the degradation of separation quality without a precise TSP response measurement. Furthermore, since Acuity relies on indoor LiDAR cameras for the visual pipeline, it is error prone in low-light situations while also suffering from interference in the presence of direct sunlight. Additional sensors can be introduced into Acuity to alleviate these errors.

7 CONCLUSIONS

We present Acuity, a system aiming at creating realistic digital twins of human subjects for AR/VR conferencing. We discovered that processing the point cloud in multiple resolutions can help reduce the computation overhead of point cloud background subtraction and clustering algorithms, based on which we implemented a "double subtraction" pipeline. Compared to deep learning-based methods that process the point cloud as a whole, we make the system more efficient and robust by paying the price of collecting a reference point cloud without human subjects. However, we decide this is worthwhile since this collection can be convenient and efficient. Another challenge we discovered for AR/VR conferencing is to stream separated audio for multiple human subjects. We found existing systems leveraging a sound source localization and separation pipeline suffer from unsatisfying performance and high buffering delay. Acuity proposes an audiovisual sensor fusion approach that uses visually estimated azimuths to guide the sound source separation algorithm and achieve success. The current version of Acuity is still a prototype that needs to be properly scaled up. The system mainly focuses on the data acquisition side of AR/VR conferencing and has to be combined with data streaming and rendering efforts to comprise an integrated application.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Hao Xiang at UCLA for helpful discussions. The research reported in this paper was sponsored in part by: the IoBT REIGN Collaborative Research Alliance funded by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF1720196; the National Science Foundation (NSF) under award # 2211301; and, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, DARPA, NSF, SRC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Jacek Dmochowski. 2007. On microphone-array beamforming from a MIMO acoustic signal processing perspective. *IEEE Transactions on Audio, Speech, and Language Processing* 15, 3 (2007), 1053–1065.

[2] Koen Bruynseels, Filippo Santoni de Sio, and Jeroen van den Hoven. 2018. Digital Twins in Health Care: Ethical Implications of an Emerging Engineering Paradigm. *Frontiers in Genetics* 9 (2018). https://doi.org/10.3389/fgene.2018.00031

[3] Anthony Capaccio. 2022. Microsoft (MSFT) US Army HoloLens Goggles Gave Soldiers Nausea, Headaches - Bloomberg. https://www.bloomberg.com/news/articles/2022-10-13/microsoft-s-us-army-version-of-hololens-goggles-gave-soldiers-nausea-headaches.

[4] X Chen, H Ma, J Wan, B Li, and T Xia. 2016. Multi-view 3d object detection network for autonomous driving. arXiv. *arXiv preprint arXiv:1611.07759* (2016).

[5] Maximo Cobos, Amparo Marti, and Jose J. Lopez. 2011. A Modified SRP-PHAT Functional for Robust Real-Time Sound Source Localization With Scalable Spatial Sampling. *IEEE Signal Processing Letters* 18, 1 (2011), 71–74. https://doi.org/10.1109/LSP.2010.2091502

[6] Advanced Television Systems Committee et al. 2003. ATSC Implementation Subcommittee Finding: Relative Timing of Sound and Vision for Broadcast Operations. *IS-191* 26 (2003).

[7] Alexandre Devaux and Mathieu Brédif. 2016. Realtime projective multi-texturing of pointclouds and meshes for a realistic street-view web navigation. In *Proceedings of the 21st International Conference on Web3D Technology*. 105–108.

[8] HARK development team. 2021. GHDSS. hark. https://www.hark.jp/document/hark-document-en/subsec-GHDSS.html

[9] HARK development team. 2021. HARKTOOL5GUI 3.3.0 Documentation. hark. https://www.hark.jp/document/packages/harktool5-gui-en/harktool5-gui.html{#}fd-conv-label

[10] HARK development team. 2021. How to Generate a Transfer Function Using HARKTOOL5. https://www.hark.jp/document/tf/generating_transfer_functions/Generating_a_Transfer_Function_Using_HARKTOOL5.html. (Accessed on 10/04/2022).

[11] Abdulmotaleb El Saddik, Fedwa Laamarti, and Mohammad Alja'Afreh. 2021. The Potential of Digital Twins. *IEEE Instrumentation Measurement Magazine* 24, 3 (2021), 36–41. https://doi.org/10.1109/MIM.2021.9436090

[12] Cédric Févotte, Rémi Gribonval, and Emmanuel Vincent. 2005. BSS_EVAL toolbox user guide–Revision 2.0. (2005).

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Jiaqi Gu, Zhiyu Xiang, Pan Zhao, Tingming Bai, Lingxuan Wang, and Zhiyuan Zhang. 2022. CVFNet: Real-time 3D Object Detection by Learning Cross View Features. *arXiv preprint arXiv:2203.06585* (2022).

[16] Yushi Hosokawa, Yasuharu Hirano, Daichi Kominami, Ikkyu Aihara, and Masayuki Murata. 2019. Implementation of a real-time sound source localization method for outdoor animal detection using wireless sensor networks. In *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*. 1–6. https://doi.org/10.1109/ICSPCS47537.2019.9008727

[17] Muhammad Huzaifa, Rishi Desai, Samuel Grayson, Xutao Jiang, Ying Jing, Jae Lee, Fang Lu, Yihan Pang, Joseph Ravichandran, Finn Sinclair, et al. 2020. Exploring extended reality with illixr: A new playground for architecture research. *arXiv preprint arXiv:2004.04643* (2020).

[18] Leia Inc. 2020. Leia Inc. – 3D Lightfield Experience Platform. https://www.leiainc.com/#Product. (Accessed on 10/30/2022).

[19] Intel. 2018. *Llibrealsense*. https://github.com/IntelRealSense/librealsense

[20] Intel. 2022. Intel Realsense Lidar L515. https://www.intelrealsense.com/lidar-camera-l515/

[21] Mike Isaac. 2022. Meta's Move to the Metaverse Drags Down Profit - The New York Times. https://www.nytimes.com/2022/02/02/technology/meta-facebook-earnings-metaverse.html. (Accessed on 10/20/2022).

[22] Anders Johansson, Greg Cook, and Sven Nordholm. 2004. Acoustic direction of arrival estimation, a comparison between root-music and SRP-PHAT. (01 2004). https://doi.org/10.1109/TENCON.2004.1414674

[23] Jungbeom Ko, Hyunchul Kim, and Jungsuk Kim. 2022. Real-Time Sound Source Localization for Low-Power IoT Devices Based on Multi-Stream CNN. In *Sensors*.

[24] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12697–12705.

[25] Ching-Feng Liu, Wei-Siang Ciou, Peng-Ting Chen, and Yi-Chun Du. 2020. A Real-Time Speech Separation Method Based on Camera and Microphone Array Sensors Fusion Approach. In *Selected Papers from IEEE ICKII 2020*.

[26] Zhi Liu, Qiyue Li, Xianfu Chen, Celimuge Wu, Susumu Ishihara, Jie Li, and Yusheng Ji. 2021. Point cloud video streaming: Challenges and solutions. *IEEE Network* 35, 5 (2021), 202–209.

[27] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. 2020. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11677–11684.

[28] Yi Luo, Zhuo Chen, and Takuya Yoshioka. 2019. Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. arXiv:arXiv:1910.06379

[29] Michal Mandlik, Zdenek Nemec, and Radovan Dolecek. 2012. Real-time sound source localization. In *2012 13th International Radar Symposium*. 322–325. https://doi.org/10.1109/IRS.2012.6233370

[30] Microsoft. 2022. Microsoft HoloLens | Mixed Reality Technology for Business. https://www.microsoft.com/en-us/hololens. (Accessed on 10/20/2022).

[31] Chao Pan, Jingdong Chen, and Jacob Benesty. 2015. Theoretical analysis of differential microphone array beamforming and an improved solution. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23, 11 (2015), 2093–2105.

[32] Despoina Pavlidi, Matthieu Puigt, Anthony Griffin, and Athanasios Mouchtaris. 2012. Real-time multiple sound source localization using a circular microphone array based on single-source confidence measures. *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on* (03 2012). https://doi.org/10.1109/ICASSP.2012.6288455

[33] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *31st Conference on Neural Information Processing Systems*.

[34] Boaz Rafaely, Yotam Peled, Morag Agmon, Dima Khaykin, and Etan Fisher. 2010. Spherical microphone array beamforming. *Speech Processing in Modern Communication* (2010), 281–305.

[35] Gernot Reigler, Ali Osman Ulusoy, and Andreas Geiger. 2017. OctNet: Learning Deep 3D Representations at High Resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[36] Roland Rosen, Georg Von Wichert, George Lo, and Kurt D Bettenhausen. 2015. About the importance of autonomy and digital twins for the future of manufacturing. *Ifac-papersonline* 48, 3 (2015), 567–572.

[37] Sudhakar Sengan, Kailash Kumar, V Subramaniyaswamy, and Logesh Ravi. 2022. Cost-effective and efficient 3D human model creation and re-identification application for human digital twins. *Multimedia Tools and Applications* 81, 19 (2022), 26839–26856.

[38] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. 2021. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *arXiv preprint arXiv:2102.00463* (2021).

[39] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 770–779.

[40] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. 2019. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *arXiv preprint arXiv:1907.03670* (2019).

[41] Weijing Shi and Raj Rajkumar. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1711–1719.

[42] Daniel Stoller, Sebastian Ewert, and Simon Dixon. 2018. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185* (2018).

[43] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 21–25.

[44] Fei Tao, He Zhang, Ang Liu, and A. Y. C. Nee. 2019. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics* 15, 4 (2019), 2405–2415. https://doi.org/10.1109/TII.2018.2873186

[45] Krisp Technologies. 2022. Noise Cancelling App & Echo Reduction Software | Krisp. https://krisp.ai/. (Accessed on 10/20/2022).

[46] Efthymios Tzinis, Zhepei Wang, and Paris Smaragdis. 2020. Sudo rm-rf: Efficient networks for universal audio source separation. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[47] Saeed V Vaseghi. 1996. Spectral subtraction. In *Advanced Signal Processing and Digital Noise Reduction*. Springer, 242–260.

[48] Velodyne. 2022. How does LiDAR Work? https://velodynelidar.com/what-is-lidar/

[49] Cor Verdouw, Bedir Tekinerdogan, Adrie Beulens, and Sjaak Wolfert. 2021. Digital twins in smart farming. *Agricultural Systems* 189 (2021), 103046.

[50] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. 2006. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing* 14, 4 (2006), 1462–1469.

[51] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. 2006. Performance measurement in blind audio source separation. In *IEEE Transactions on Audio, Speech and Language Processing, Institute of Electrical and Electronics Engineers*, Vol. 14. 1462–1469. https://doi.org/nria-00544230

[52] John Wang and Edwin Olson. 2016. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[53] Xiao Xu, Burak Cizmeci, Anas Al-Nuaimi, and Eckehard Steinbach. 2014. Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating. *IEEE Transactions on Instrumentation and Measurement* 63, 11 (2014), 2558–2569.

[54] Yan Yan, Yuxing Mao, and Bo Li. 2018. Second: Sparsely embedded convolutional detection. *Sensors* 18, 10 (2018), 3337.

[55] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 2020. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11040–11048.

[56] Georgios Zamanakos, Lazaros Tsochatzidis, Angelos Amanatiadis, and Ioannis Pratikakis. 2021. A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving. *Computers & Graphics* 99 (2021), 153–181.

[57] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2021. Efficient volumetric video streaming through super resolution. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*. 106–111.

[58] Yin Zhou and Oncel Tuzel. 2017. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *arXiv:1711.06396 [cs.CV]*.