MR Object Identification and Interaction: Fusing Object Situation Information from Heterogeneous Sources

JANNIS STRECKER, University of St. Gallen, Switzerland
KHAKIM AKHUNOV, University of Trento, Italy
FEDERICO CARBONE, University of Trento, Italy
KIMBERLY GARCÍA, University of St. Gallen, Switzerland
KENAN BEKTAŞ, University of St. Gallen, Switzerland
ANDRES GOMEZ, TU Braunschweig, Germany
SIMON MAYER, University of St. Gallen, Switzerland
KASIM SINAN YILDIRIM, University of Trento, Italy

The increasing number of objects in ubiquitous computing environments creates a need for effective object detection and identification mechanisms that permit users to intuitively initiate interactions with these objects. While multiple approaches to such object detection – including through visual object detection, fiducial markers, relative localization, or absolute spatial referencing – are available, each of these suffers from drawbacks that limit their applicability. In this paper, we propose ODIF, an architecture that permits the fusion of object situation information from such heterogeneous sources and that remains vertically and horizontally modular to allow extending and upgrading systems that are constructed accordingly. We furthermore present BLEARVIS, a prototype system that builds on the proposed architecture and integrates computer-vision (CV) based object detection with radio-frequency (RF) angle of arrival (AoA) estimation to identify BLE-tagged objects. In our system, the front camera of a Mixed Reality (MR) head-mounted display (HMD) provides a live image stream to a vision-based object detection module, while an antenna array that is mounted on the HMD collects AoA information from ambient devices. In this way, BLEARVIS is able to differentiate between visually identical objects in the same environment and can provide an MR overlay of information (data and controls) that relates to them. We include experimental evaluations of both, the CV-based object detection and the RF-based AoA estimation, and discuss the applicability of the combined RF and CV pipelines in different ubiquitous computing scenarios. This research can form a starting point to spawn the integration of diverse object detection, identification, and interaction approaches that function across the electromagnetic spectrum, and beyond.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; **Ubiquitous and mobile computing systems and tools**; • **Hardware → Radio frequency and wireless interconnect**.

Additional Key Words and Phrases: mixed reality, detection, identification, computer vision

ACM Reference Format:
Jannis Strecker, Khakim Akhunov, Federico Carbone, Kimberly García, Kenan Bektaş, Andres Gomez, Simon Mayer, and Kasım Sinan Yıldırım. 2023. MR Object Identification and Interaction: Fusing Object Situation Information from Heterogeneous

Authors' addresses: Jannis Strecker, jannisrene.strecker@unisg.ch, University of St. Gallen, St. Gallen, Switzerland; Khakim Akhunov, khakim.akhunov@unitn.it, University of Trento, Trento, Italy; Federico Carbone, federico.carbone20@gmail.com, University of Trento, Trento, Italy; Kimberly García, kimberley.garcia@unisg.ch, University of St. Gallen, St. Gallen, Switzerland; Kenan Bektaş, kenan.bektas@unisg.ch, University of St. Gallen, St. Gallen, Switzerland; Andres Gomez, andres.gomez@tu-bs.de, TU Braunschweig, Braunschweig, Germany, 38106; Simon Mayer, simon.mayer@unisg.ch, University of St. Gallen, St. Gallen, Switzerland; Kasım Sinan Yıldırım, kasimsinan.yildirim@unitn.it, University of Trento, Trento, Italy.

2474-9567/2023/9-ART124
https://doi.org/10.1145/3610879

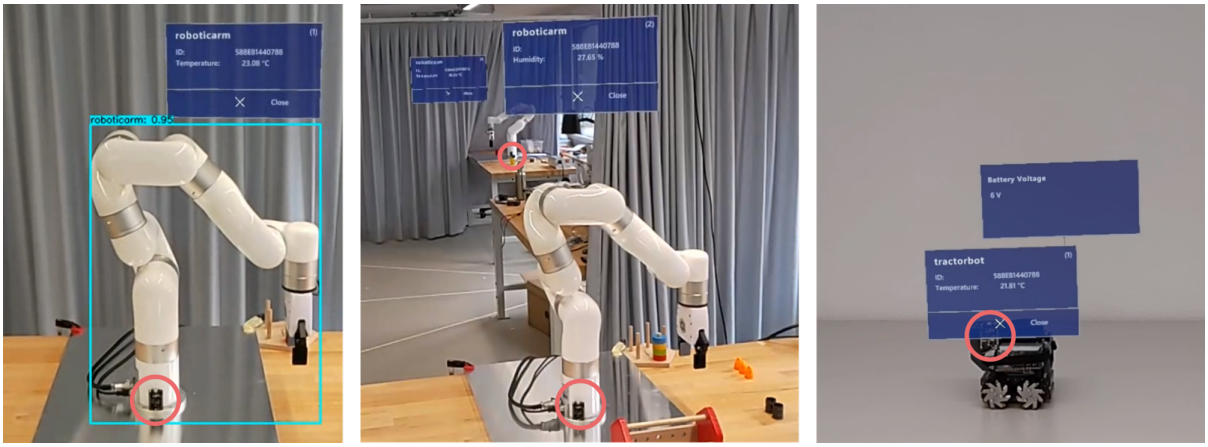


Fig. 1. BLEARVIS can identify objects (left) while differentiating between visually identical devices (center), and create interaction possibilities for users by accessing real-time data from these devices (right). Red circles indicate the positions of the Bluetooth Low Energy (BLE) tags.

Sources. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 124 (September 2023), 26 pages. https://doi.org/10.1145/3610879

1 INTRODUCTION

With the intensifying trend towards increasingly populated ubiquitous computing environments, we require widely applicable mechanisms that allow people to efficiently interact with objects in their surroundings. In environments like, e.g., smart factories [22], agriculture [3], or smart buildings [82] such interaction possibilities would be beneficial, e.g, for supervising and controlling the execution of robots' workflows, or checking the maintenance status of tractors. In these environments, objects (e.g., industrial robots) are often visually identical and therefore need to be manually differentiated by a human operator when they want to interact with them.

Interaction in this context commonly requires, first, the identification of the object to interact with and, second, the rendering of an (implicit or explicit) interface to provide users with object-specific information and means of interaction. For the first step of *identifying* interactable resources in the user's environment, researchers and practitioners have proposed to use visual object detection [13, 16, 51, 77]. Although this approach does not actually permit to *identify* objects but rather to visually *classify* them, it produced viable results [17, 41, 50, 52] already before the recent performance boost of vision-based object classification mechanisms that was delivered through the advent of Convolutional Neural Networks with access to vast amounts of training data; well-known examples of such systems are YOLO [68] and the Common Objects in Context (COCO) dataset [49]. Regarding the *rendering* of (explicit) user interfaces for successfully recognized interactable devices in a user's surroundings, a lot of recent research [4, 35, 36, 55–57, 61, 77] has focused on the use of Augmented Reality (AR) or, broader, Mixed Reality (MR) devices. Specifically, head-worn MR permits placing such interfaces at appropriate locations close to the respective physical objects within a user's field of view, and allows for a variety of interaction modalities (e.g., through gestures, voice, or traditional visual menus) while keeping the user's hands free.

Since MR devices typically feature forward-facing cameras in addition to their see-through user interface, they provide a plausible platform for combining vision-based object classification with the rendering of an MR interface for interacting with ubiquitous computing objects and the services they offer. However, using an only vision-based mechanism for object detection has several limitations. First, even though there has been

tremendous progress in object classification, existing approaches are still only able to differentiate between several hundred classes. While this problem might be overcome with further advances in Computer Vision (CV), a more fundamental problem remains since, while these mechanisms are able to *classify* objects, they cannot in general *identify* them from visual features only – this is immediately clear when considering scenes with multiple visually identical objects. This poses relevant limitations in cluttered scenes with many similar objects, such as in supermarkets (e.g., packaging of coffee beans with different intensity levels) or in industrial environments (e.g., cutting heads for CNC milling machines that visually differ only very slightly). To enable human users to interact intuitively with *specific individual objects* in such environments, a method is hence needed to *detect and identify* individual objects at the same time. The identification is especially important because only if the visualized information can be clearly assigned to a specific device does it become helpful for users' interaction with this device; this can, for instance, be seen if a user is interested in obtaining readings from several visually similar sensor nodes, or the user intends to control different visually similar robots.

Many different solutions have been proposed to address the object identification problem. Objects of interest may be tagged, e.g., with fiducial markers [23, 24, 39, 71, 72]. Such markers may occur in the form of human-visible 1D or 2D barcodes, such as Quick Response (QR) codes [38], might be elegantly embedded in the design of objects [30], or might even be completely invisible to the human eye, e.g., by using infrared signatures [19]; such codes may also be interactive [33, 74]. Although the use of fiducial markers for object identification is an inexpensive passive solution, marker-based approaches are typically susceptible to changes in illumination and affine transforms, have strong requirements on the size and clarity of markers in the camera image, require a clear line of sight, and tend to visually clutter scenes [30]. Furthermore, they are typically static: Once printed, the data can no longer be changed. This can limit their use to storing unique identifiers, since they cannot self-adjust their data to dynamic changes in their local context (e.g., environment, device orientation, user preferences, etc.).

Conversely, *active tags* do not suffer from these limitations since they can transmit any local information via an RF channel. This requires more expensive components and higher energy consumption, although these limitations are shrinking with new generations of energy-driven devices [28]. On the positive side, active tags can locally gather sensor data, process it locally, and transmit relevant data independently from any other device. They can furthermore be used to provide *out-of-band contextual information* to other systems. For instance, if a receiver is equipped with a customized antenna array, it could estimate the angle of arrival (AoA) of specialized RF packets, and thereby find the direction towards the origin tag. Such customized antenna arrays can be designed small enough to fit in MR glasses [87], allowing users to estimate the RF AoA based on their dynamic field of view, even when the tag is visually obstructed.

Since each of these detection and identification technologies has its respective weaknesses, we argue that for the *detection*, *identification*, *localization* of, and *interaction* with objects in a user's environment, it is beneficial to integrate information on the situation of an object from heterogeneous sources. Therefore, in this work we present the *object identification and detection framework (ODIF)*, which permits the vertically and horizontally modular fusion of object situation information. To demonstrate our approach, we built BLEARVIS, a system that enables intuitive interaction with objects in MR by including two information modules, CV-based object detection and RF-based direction finding and identification[1]. By combining those two sources, our system can provide the best of both worlds: it can yield high-accuracy in-scene object bounding, provide identifiable information for (visually identical) objects, and display dynamic sensor data or actuator controls to users.

In the remainder of this paper, we first introduce relevant related research (Section 2), then detail our approach and implementation (Section 3), and evaluate and demonstrate the functionality of our system (Section 4). We conclude this work by discussing our implemented system, stating its limitations, and outlining possible future work (Section 5 and Section 6).

[1]The source code of our prototype is publicly available in the following repository: https://github.com/Interactions-HSG/blearvis/.

2 RELATED WORK

Our work stands at the intersection of Mixed Reality, Ubiquitous Computing, Computer Vision (specifically object detection), and object identification using tags, which we combine to yield the BLEARVIS prototype. In the following, we survey relevant work that serves as the foundation of our proposed approach.

2.1 Mixed Reality in Ubiquitous Computing Environments

Our real-life experiences rely on our abilities to sense visual, audio, olfactory, taste, and tactile signals. However, an MR experience is often considered to appeal to our *visual* perception that can be created with fully virtual graphical elements, such as in Virtual Reality (VR) environments or with overlaying virtual elements onto real environments such as in AR. Hence, MR allows users to see and interact with *artificial* or *augmented* versions of real objects and scenes. This description of MR suits well to Milgram's virtuality continuum [54] (i.e., focusing mainly on visual experiences in real, AR, and VR environments), which presented an exhaustive taxonomy, conceptual dimensions, and various classes of visual displays. The early prototypes of MR devices had many limitations and were difficult to operate for users [2, 80], but today's MR devices and recent prototypes aim to induce immersive and multimodal experiences that include electromuscular [9], vestibular [78], tactile [63], auditory [86], tastable [67], and olfactory [47] stimulation. Desktop, hand-held, or wearable (e.g., head-mounted or near-eye) MR devices [32] can provide users with convincing and useful experiences by sensing, computing, and displaying information about the users and objects in their surroundings. For example, current MR head-mounted displays (HMDs) can track 3D movements of their user's head, hands, and eyes [8, 58] with the help of embedded sensors that can measure inertial, time-of-flight, and electromagnetic (e.g., infrared or visible light) signals. Furthermore, with auxiliary sensors, they can provide 360° live videos [81] to multiple users, or haptic and thermal stimulation for individuals, e.g., in special needs situations [60]. Most importantly, with wireless connectivity, they can be used in the detection and contextualization of objects [77] and text [79]. Thus, in many MR applications, users can freely move in a real or virtual environment and maintain explicit, implicit, collaborative, and even hands-free interaction [5, 62] with physical devices and virtual objects.

In the near future, MR technologies hold the potential to become pervasively available and a valuable part of many training, education, and business activities and remotely performed (or tele-) operations [58]. However, many challenges exist, such as providing spatial, temporal, and visual realism [32], and overcoming limited interoperability and user acceptance of MR devices in ubiquitous computing environments [6, 58]. In the following section, we give an overview of current object detection and identification solutions and then discuss which of these solutions can be suitably integrated into an MR-HMD that allows users to have access to relevant information in ubiquitous computing scenarios.

2.2 Computer Vision for Object Detection

The usage of CV techniques in MR applications has been explored since early works for identifying and locating scene objects [72]. Many contributions have focused on the usage of fiducial markers, which are patterns placed on a physical environment accompanied by an algorithm capable of detecting and identifying such patterns. Simple markers are printed 1-D or 2-D codes, which typically require visible markers to be used. Famously, the ARToolkit [39] proposes a video conferencing system that uses markers for positioning virtual elements in the real world using AR. Webtag [24] is a marker based on ARTags [23] that adds a second tier of information to the markers. Webtag adds a set of 32 additional bits of data to encode a custom URL that provides users with further information about an object. In [38], Quick Response (QR) codes are introduced as an alternative to traditional MR markers; with a total of 2953 bytes, these have been standardized to hold large amounts of information. Other recent efforts on fiducial markers in AR applications include Chromatag [18], which uses red and green regions to increase detection speed. As a major drawback of using fiducial markers in MR, the camera often needs to

be close and at an appropriate angle to the marker to be recognized. Furthermore, fiducial markers are often perceived as unwanted clutter in a visual scene [30] – leading researchers to turn to visual object classification as a means to interact with ambient objects [31, 51].

Object detection (without the aid of markers) consists of accurately locating objects of interest in an image and solving a classification problem. Thus, the output of an object detection algorithm is typically a label indicating a type of object and the location coordinates of such an object. Early object detection algorithms relied on bottom-up features such as corners or edges [15]. In recent years, thanks to the processing power of machines and the availability of training data, Convolutional Neural Networks (CNNs) started gaining a lot of attention [89]. In 2014, *RCNN* was proposed as the first CNN for object detection [27]. RCNN takes a two-stage approach: In the first stage, region proposals of candidate objects are extracted; then, through a large CNN features for each region are computed. In the second stage, each region is classified using linear SVMs. *Fast RCNN* [26] and *Faster RCNN* [73] are evolutions of RCNN that keep the two-stage approach. These and other two-stage approaches offer very good detection accuracy but compromise on speed [89]. Parallel to this work, one-stage algorithms were proposed; these extract region proposals and classify them at the same time, reducing computation time significantly. *You Only Look Once* (YOLO) [68] was the first algorithm that proposed an architecture in which a single neural network predicts bounding boxes and class probabilities at the same time. Even though YOLO works significantly faster than two-stage approaches, it struggles with the detection of small objects as well as objects at unfamiliar angles, and its object location estimation is lacking. YOLOv3 [69] improves the algorithm's performance with small objects, and YOLOv4 [12] offered an even faster and more accurate implementation. Finally, YOLOv7 [83], improves on YOLOv4 by requiring 75% fewer parameters, 36% less computation time, and an average precision increase of 1.5%.

Similar to fiducial markers, object detection has been used together with AR and MR applications. In [20] an application for visually impaired people is proposed. On a Microsoft HoloLens that records a user's surroundings, a YOLO algorithm is used to detect objects and provide feedback to a user wearing the HoloLens through spatial audio. In [77], digital companions for working environments are proposed. Here, YOLOv4 is used to evaluate the camera feed of a Microsoft HoloLens 2 (HL2); once objects in the user's surroundings are detected, users receive meaningful assistance to interact with them.

Even though deep learning approaches produce very good results for object detection, they require a lot of time and manual effort to prepare representative datasets for training. Often, available models that have been trained on large datasets cannot be used in applications that require detecting non-everyday objects (e.g., industrial robots). Thus, they can only be used for transfer learning tasks, requiring application developers to prepare scenario-specific datasets and re-training the model. Also, computer-vision-based deep learning approaches can be sensitive to changes in lighting conditions as well as to affine transforms (including changes in the distance to the object of interest). Most importantly, for the purposes of our research, consisting of initiating interactions with specific objects, vision-based object detection algorithms only provide detection at the class level: These approaches are hence not capable of *identifying* one of the multiple instances of a type of object.

2.3 Object Identification Using Tags

Apart from placing fiducial markers, one way to facilitate the *identification* of environment objects is to decorate the physical object with active or passive radio-frequency (RF) tags. Active tags have their own power source and can thus transmit information at predetermined intervals. Passive tags do not have their own power source and, hence, require an additional device (often referred to as a *reader*) that supplies them with sufficient energy for data transmission (e.g., through electromagnetic coupling or RF backscattering). Both types of tags have been used in combination with CV-based systems to identify devices, as summarized below.

A prominent class of passive tags, Radio-frequency identification (RFID), was used by Xie et al. [85] together with two receiver antennas to estimate a transmitting tag's phase difference and signal strength. Then, by combining the RF information with data from a depth camera, an image with identifiable objects can be derived. Other works have also utilized the combination of passive RFID and computer vision in robotic arm applications. Boroushaki et al. presented the RFusion system [14] that is used in a setup where a robotic arm can grab RFID-tagged objects, even if these are occluded. This is done by estimating the relative position of the tags using an RF-visual reinforcement learning approach. These systems need to work around the low localization accuracy with the used passive tags, for instance by actively moving the antenna to measurement points – for instance, RFusion requires at least three such vantage points.

While low-power *active RF tags* and radios have been used for many years in wireless sensor networks, active tags have fewer constraints (e.g. energy, form factor, etc.) than passive systems and are thus better suited for advanced sensing, local processing, user interaction, and transmitting dynamic data. The vast majority of applications with active tags use a single omnidirectional antenna for both the receiver and the transmitter; however, since the Bluetooth 5.1 standard [10] was published, manufacturers are now integrating antenna arrays for direction-finding applications in which either the AoA or the Angle-of-Departure (AoD) can be calculated. Recent works have focused on utilizing this new direction-finding feature in different contexts. Park et al. [61], for instance, join a receiving antenna array with a handheld device to visualize the origin of the wireless signal on the handheld device. To enable a more interactive user experience, the authors use the device's IMU to keep track of its orientation and correctly display the signal's origin on the display. Zhang et al. [87] propose using an AoA-based system for gesture detection. By mounting an antenna array on a smart-glass frame, the user is able to point to other active tags or nod in a particular tag's direction, and the system is able to detect these signals by analyzing the RF signal response. Neither of these AoA-based systems uses any vision-based information to correlate with the RF-based signals. In this regard, the closest comparable system to our work is VisBLE by Jiang et al. [34], which is composed of a receiver antenna array and a smartphone. The AoA data is then combined with the computer vision pipeline that recognizes the BLE transmitter and visualizes it on the smartphone's display.

Compared to VisBLE, our proposed system BLEARVIS focuses on MR applications and can detect a wide variety of objects and correlate it to a (potentially non-visible) BLE tag, thus placing the object's information in the correct position of the HMD. SpotBLE [7] is another similar system that correlates RF information from a BLE beacon with vision-based data, but the authors do not provide any details as to whether SpotBLE uses an antenna array to estimate the AoA. While there are existing MR-based systems for displaying sensor data from various devices [25, 53, 55], these solutions require the system to know beforehand where the devices are located and may fail to display the data when these devices move or get occluded.

We argue that an MR system that integrates RF-based direction finding and vision-based object detection is useful in many ubiquitous computing scenarios where the technologies' respective strengths compensate for each other's weaknesses. While vision-based object detection can only classify objects but not reliably identify them, the RF signal provides a *unique identifier* for each object. On the other hand, the CV subsystem permits to *accurately position* MR content on top of close-by objects, while this is not possible in general using the RF signal only (e.g., because of its accuracy, or when two devices share the same AoA); still, for occluded or distant objects – which both cannot be detected using CV – the RF signal *provides a fallback* for the provisioning of user interfaces that are positioned usefully close to the (invisible or faraway) object of interest.

3 ODIF: FUSION OF OBJECT SITUATION INFORMATION

In this section, we derive and present the ODIF architecture which permits the vertically and horizontally modular fusion of object situation information (see Figure 2). Demonstrating the applicability of this architecture, we then

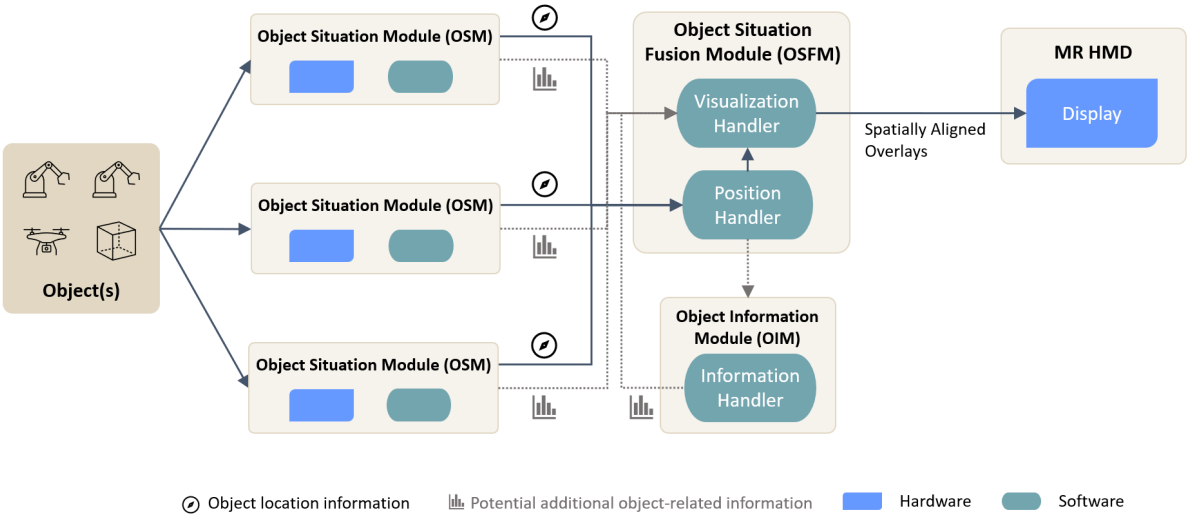
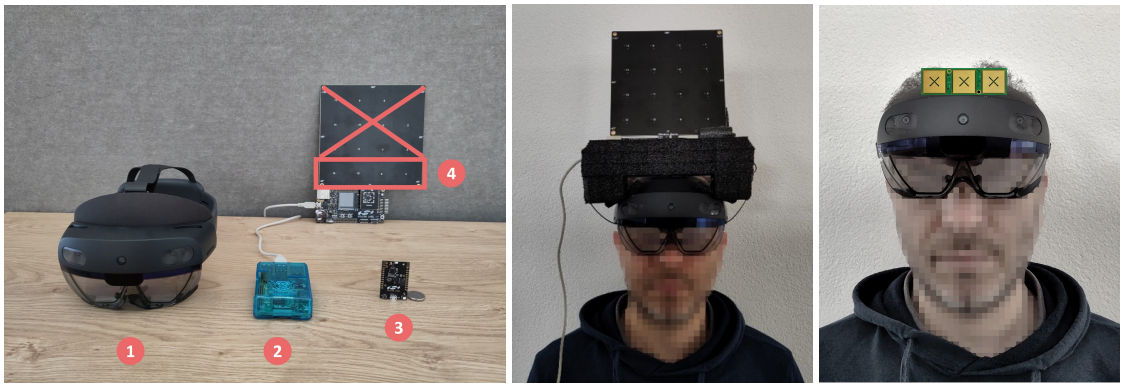


Fig. 2. The ODIF architecture: Several OSMs provide location information (i.e., a directional vector or coordinates of a bounding box) regarding one or multiple objects to the OSFM. The OSFM's *Position Handler* transforms these data into coordinates suitable for the MR HMD's Display. The Visualization Handler then guarantees the correct display of the information in MR. Optionally, the OSMs or OIM(s) can provide additional data from or about the objects and their context, such as an object ID or sensor data, or may provide interaction interfaces; these possibilities are added to the displayed MR content by the Visualization Handler.

report on the BLEARVIS system, which applies ODIF to fuse object situation information from a visual object detector and a BLE-based direction-finding and identification module.

Our goal in the conceptualization of ODIF was to create an architecture for the integration of information about an object's situation from diverse sources. Our architecture hence strongly emphasizes modularity: It supports *vertical modularity* regarding the (very conceivable) replacement of individual system components. This is required since, over time, more capable versions of individual components (e.g., improved object detectors) will become available and since we hence intend to facilitate the integration of such components with any system that follows ODIF. ODIF furthermore supports *horizontal modularity*, that is, the extension of the system with further sources that supply it with information about the location and identity of physical (or virtual) objects. This is important from a conceptual perspective because we want to keep our architecture open to permit the integration of diverse approaches and technologies that provide information about an object's current situation. These range from services that provide an object's GPS location over visual object detection systems such as YOLO and include RF-based relative localization systems such as BLE's AoA estimation as well as localization approaches that are based on ambient audio signals or on audio/video combinations [66]; they furthermore include approaches that have today only been demonstrated in the laboratory, such as [88] which can reconstruct human body meshes through walls, as well as AoA estimation systems for long-range communication networks such as LoRay [11]. We refer to such information sources about the (relative or absolute) location of an object and/or its identity, as *Object Situation Modules* (OSM). In addition, ODIF foresees the optional integration of *Object Information Modules* (OIM) that are able to fetch additional information from the identified devices (e.g., to display current sensor data) or enable interaction possibilities with these devices, e.g. through W3C WoT



(a) (b) (c)

Fig. 3. The devices used in BLEARVIS are depicted in (a): (1) MR HMD (Microsoft HoloLens 2), (2) Host Device (Raspberry 3 Model B), (3) BLE Sensor Tag (EFR32BG22 Thunderboard), and (4) Antenna Array (we use only the lowest row of antennas as indicated by the red box in the figure). Figure (b) shows a user wearing the MR HMD with the integrated antenna array in its current size. Figure (c) sketches the envisioned future prototype with a smaller antenna array (U-Blox ANT-B11[3]).

Thing Descriptions (TD)[2]. To guarantee vertical and horizontal modularity, the ODIF architecture does not make assumptions about the internals of OSMs and OIMs nor about the data that these modules require themselves. Instead, we reduce the OSM and OIM interface to their interaction with a single *Object Situation Fusion Module* (OSFM). The OSFM hence contains all logic required to fuse different sources of object situation information that it obtains from heterogeneous OSMs in the form of absolute or relative locations and universally unique or locally qualified identities. When fusing object situation information in the OSFM, systems that are based on our architecture can adapt this integration to the strengths and weaknesses of the utilized OSMs as well as their current context. Our software-based integration approach also implies that ODIF is compatible with both, commercial off-the-shelf as well as custom-made hardware – however, in case the added hardware module does not support a standard means of wired or wireless communication with the rest of the system, a custom-made gateway might be required.

3.1 BLEARVIS: Fusing CV and BLE

We have applied the ODIF architecture to the fusion of information from a visual object detector and a BLE-based object direction finding and identification service. Regarding the ODIF architecture, BLEARVIS comprises two OSMs: The first OSM (*CV Module*) on an MR HMD is used to find and classify objects in the view of the device. This information is fused with the output of the second OSM (*BLE Module*) that uses the RF signal of devices in the user's surroundings to identify them uniquely and to calculate the estimated AoA of the device's RF signal. To enable this, BLEARVIS uses a specialized antenna receiver array that we have integrated with an MR HMD (see Figure 3). BLEARVIS facilitates user interaction with ubiquitous computing devices that support RF-based relative localization where through our system's OSFM we take advantage of their individual strengths to compensate for each other's weaknesses. This mutual compensation allows BLEARVIS to render spatially aligned MR interfaces for objects, even if these objects have identical visual appearance – here, the accurate spatial alignment is enabled by the *CV Module* while the identification capability is added by the *BLE Module*.

[2]https://www.w3.org/2019/wot/td
[3]Image source: https://www.u-blox.com/en/product/ant-b11-antenna-board

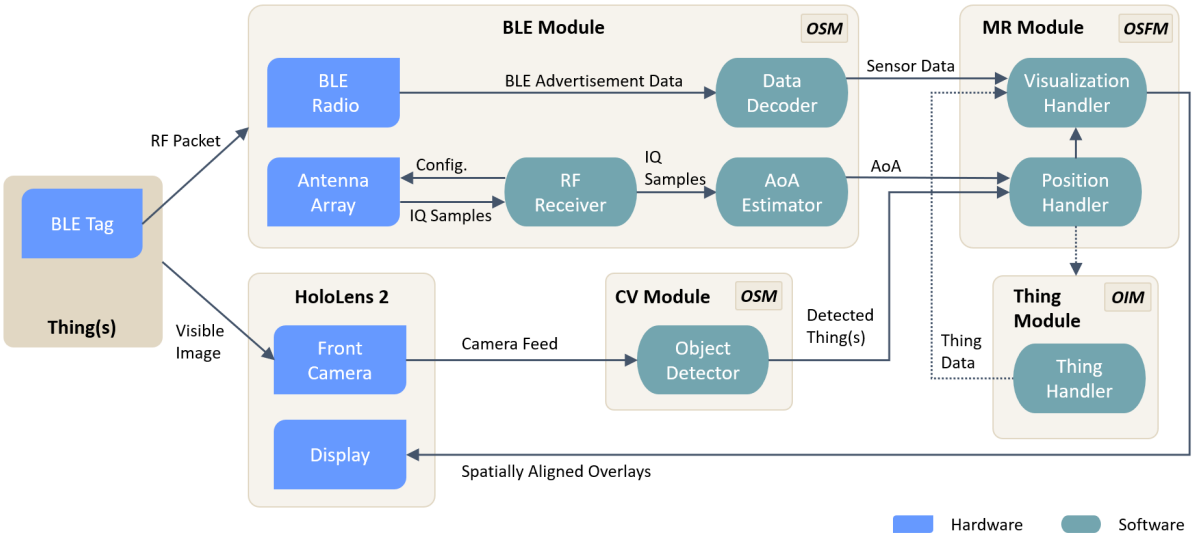


Fig. 4. The BLEARVIS system contains two OSMs (*BLE Module* and *CV Module*), one OIM (*Thing Module*) and one OSFM (*MR Module*).

Thanks to the *BLE Module*'s AoA, the alignment even remains robust to partial occlusions where the *CV Module* might lose its ability to classify an object, and MR interfaces stay loosely aligned (through the *BLE Module*) even when the objects are completely hidden from sight. Furthermore, there is an additional optional OIM (*Thing Module*) that retrieves additional information from the identified devices and their context.

In the following, we discuss the individual modules of BLEARVIS and their interactions, and share relevant implementation details, while referring the reader to the overview of BLEARVIS's architecture in Figure 4.

3.2 BLE Module

The first OSM in the BLEARVIS system is the *BLE Module* which uses BLE sensor tags (see Sect. 3.2.1) that are attached to objects in the system's surroundings. These tags allow for the collection of sensor data, and an AoA-based localization through the system's HMD-attached antenna array (see Sect. 3.2.2). Location-relevant data is passed to the RF Receiver component in the form of IQ (*In-phase* and *Quadrature*) samples, while the RF Receiver is responsible for the configuration of the antenna array (e.g., setting the number of active antenna cells). The AoA Estimator then uses the IQ samples to estimate the signal's AoA which it sends to the *MR Module*. In addition to supplying a signal for AoA estimation, the BLE sensor tag's raw sensor data (e.g., temperature and humidity) is conveyed to the *MR Module* along with the tag's MAC address, which is used as a unique identifier by the system.

3.2.1 BLE Sensor Tag. In our implementation, the BLE sensor tag is a low-power *EFR32BG22 Thunderboard* attached to the target device. The method that allows us to perform AoA localization while excluding interference with other devices was introduced in BLE version 5.1 to permit AoA localization for low-power devices [45] and we further explained it in Section 3.2.4. To transmit sensor data, the tag uses iBeacon [46] protocol packets in which beacon minor and major number fields are hijacked and substituted by actual data from requested tag sensors. Depending on the system configuration, the packets can be sent periodically or upon a predefined event.

The BLE Sensor Tag is powered by a small battery or energy-harvesting module collecting power from ambient sources (e.g., sunlight, RF waves, or vibration). The power autonomy of the tag enables portability and eases the integration of the *BLE Module*, keeping BLEARVIS independent of the application scenario. However, each powering strategy has its own trade-offs, which need to be considered at the system design stage. An available battery introduces more power stability but is heavy, occupies more space, necessitates periodic charging and changing, and must be properly disposed of at the end of its lifecycle. Contrarily, a transiently powered tag requires minimum maintenance, but the module's performance is highly dependent on environmental power availability [28]. However, this unpredictability can be relaxed by supporting a wider range of ambient power sources.

3.2.2 Antenna Array. The BLE sensor tag sends BLE packets to an antenna array, which is mounted on the MR HMD (see Figure 3b above). The antenna array in our prototype consists of 16 patch antennas that are arranged in a 4×4 pattern. Each antenna in the array observes phase differences that occur due to the different distances of the respective antenna with the BLE sensor tag. This is achieved by switching the antennas during the reception of the BLE packets, resulting in IQ samples that contain phase information for each antenna. The IQ Samples are then fed to the AoA estimator. Since we are only interested in 1-dimensional AoA measurements, our prototype uses only the lowermost row of four antennas of the array. While an antenna *array* naturally occupies more space than an individual antenna, the size of the array (see Figure 3a) could be reduced considerably while sustaining the same localization accuracy if using a more compact BLE AoA antenna and sensor board, such as the U-Blox ANT-B11 which occupies 29.5×93.5 mm and consumes on average only 0.8 mA and 0.14 mA of current at active and standby modes, respectively. Such an array would also weigh considerably less than our prototype's antenna array and could hence be integrated with an MR HMD (see Figure 3c), yielding a form factor that is comparable to currently available MR HMDs.

3.2.3 AoA Estimator. The AoA Estimator's task is to estimate the AoA of an emitted signal arriving at the antenna array, for whose calculation there are several possible solutions.

The *trivial solution* is trigonometry-based: If the BLE signal arrives at multiple antennas with an oblique incident angle, it results in a phase difference because the wavefront needs more time to reach the further antennas. This phase shift can be translated into a distance between the wavefront and the farther antenna (Δ). In this case, if the transmitter is far enough from the receiver, the angle of incidence can be easily estimated from that distance by using the general trigonometric equation: $\Theta = \arccos(\frac{\Delta}{d})$, where d is the distance between the antennas. While simple, this approach does not take into account the reflection of the signal in space, the interference of which can lead to incorrect results.

The main idea behind the *classical beamformer approach* [42] is to maximize the output power as a function of the angle. The maximized power, $P(\Theta) = \frac{\alpha^H(\Theta)R_{xx}\alpha(\Theta)}{\alpha^H(\Theta)\alpha(\Theta)}$, is calculated using the steering vector of the antenna array (α), the Hermitian transpose of a matrix (H), and an approximation of the so-called sample covariance matrix (R_{xx}). To find the AoA, one needs to loop through the arrival angles Θ and find the maximum peak power, P. The angle producing the maximum power corresponds to the desired AoA.

Another technique for arrival angle estimation is *subspace estimation*. The popular representative of this algorithm category is MUSIC (MUlti SIgnal Classification) [75]. MUSIC estimates the eigendecomposition on the covariance matrix as follows: $R_{xx} = VAV^{-1}$, where A is a diagonal matrix containing eigenvalues and V is a matrix containing the corresponding eigenvectors of R_{xx}. Then, the AoA can be found by looping through the desired values of Θ and finding the location of the maximum peak value of P, i.e., $P(\Theta) = \frac{1}{\alpha^H(\Theta)VV^H\alpha(\Theta)}$.

For this implementation, we select the MUSIC algorithm, since, compared to the classical beamformer approach, its angle estimation technique is more tolerant to receiving a single packet at a time, which is limited by the Bluetooth collision avoidance mechanism. To overcome interference due to signal reflection on environmental

obstacles, we apply spatial smoothing to the signal correlation matrix [43]. This technique solves multipath problems in suboptimal environments by decorrelating the signal and allowing for more accurate computation. Since we have to deal with packets without a fixed frequency, the estimator must perform a calculation even with only a single packet. To take advantage of cases in which we have a high packet frequency, the calculated angles are saved in a buffer, and at a constant frequency, the average value is calculated. For an accurate analysis, the signal on the various antennas furthermore needs to be sampled at the same time, which is impossible due to the hardware limitations, as explained above. Therefore, to make the IQ samples homogeneous it is necessary to transform them, which is possible because the first eight samples are purposely sampled from the same antenna. To achieve this, the antennas receive short continuous waves (CW) from the sensor tag, where the transmitter and receiver are synchronized to know when the CW signal is sent. The CW signal is sent as a Constant Tone Extension (CTE) of a normal BLE packet [45] with a length between 16 μs and 160 μs. The CTE starts with a guard period (4 μs) followed by a reference period (8 μs). After the reference period, the CTE comprises a sequence of alternating switch slots and sample slots, and the receiver samples the IQ components of the baseband signal with its native sampling rate. The samples are then downsampled to 1 sample/μs rate. The first four samples (taken in the guard period) are discarded, then eight samples (taken in the reference period) are stored in the sample buffer. Finally, every sample taken in switching slots is discarded and every sample taken in sample slots is stored in the sample buffer. Calculating the phase shift between these packets and applying this correction to subsequent samples allows us to obtain simultaneous samples.

3.2.4 Host Device. The Host Device is a compact human-wearable hardware platform on a Raspberry Pi 3 Model B (see Figure 3a) that configures the antenna array, obtains IQ samples and sensor data, and estimates the AoA in real-time. In our prototype, the host device is attached to the antenna array via USB and is kept by the user along with a power source. This is not a fundamental limitation of BLEARVIS but is due to the impossibility of connecting the antenna array directly to the HoloLens 2 device (incompatible connectors). The Host Device's Data Decoder receives iBeacon packets from the BLE sensor tag, parses them to obtain the sensor data, and transmits this data to the *MR Module*. The RF Receiver on the Host Device specifies the number of active antennas in the antenna array and collects IQ Samples. When BLEARVIS is running, the Host Device uses the MQTT protocol to deliver AoA data, sensor data, and the MAC addresses of the respective BLE sensor tag to the *MR Module*.

3.3 CV Module

The second OSM in the BLEARVIS system is the *CV Module* in charge of visual object detection. In our prototype, this module streams the HL2's camera feed to an object detector via the HL2's Device Portal API's *Mixed Reality Streaming*[4]; MR holograms are excluded from this stream. The object detector returns the class names of the detected objects in an image and sends this data to the *MR Module*, along with their respective bounding boxes on the camera feed and the confidence for each object. For our prototype, we have trained the YOLOv7 algorithm[5] to be able to recognize objects of interest in our laboratory context and we run the object detector on an external computer with a frame rate of around 35fps on a NVIDIA RTX 3060 GPU. YOLOv7 was chosen, given that it is a state-of-the-art object detector that offers fast detection performance and is suitable for mobile and other devices with limited capabilities.

[4] https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/device-portal-api-reference#mixed-reality-streaming
[5] https://github.com/WongKinYiu/yolov7

3.4 Thing Module

The *Thing Module* is an optional OIM that can fetch additional information from the detected Things. Our prototype was deployed in a ubiquitous computing environment where physical devices are connected to W3C WoT Thing Descriptions (TD)[6]. W3C WoT TDs are machine-readable and machine-understandable descriptions of the programming interface of a Thing. They enable protocol-independent interaction with these devices via *Property Affordances* (e.g., the current measurement of a temperature sensor) or *Action Affordances* (e.g., to move the gripper of a robot arm). In our setup, the MAC addresses of each BLE sensor tag are associated with the URL of the respective object's W3C WoT TD. Once BLEARVIS has identified any TD-connected object in our environment, it uses its *Thing Module* to load the object's W3C WoT TD. If it is able to parse the object's TD successfully, it then sends corresponding requests to achieve a desired interaction with the identified object and forwards the received information from the Thing to the *MR Module*. In the case of our prototype, we are using the *Thing Module* to obtain information from identified objects and display this information spatially referenced to the object via the MR HMD.

3.5 MR Module

In the BLEARVIS system, object situation information from the *BLE Module* and the *CV Module* is fused in the *MR Module*; if the detected object(s) feature W3C WoT TDs, this is enriched with information from the *Thing Module*. To this end, we implemented the *MR Module* as an MR application for the HL2[7] using the Unity Game Engine (2020.3) and the Mixed Reality Toolkit v2.8 (MRTK)[8].

To use BLEARVIS, a user wears the HL2 with the antenna array mounted on top of it (see Figure 3b). When they start the BLEARVIS application, they are greeted with a menu that lets them add one or multiple devices. The user first chooses which OSMs are available (*BLE Module*, *CV Module*, or both) and then clicks a button to add the device(s). While this happens, a loading screen tells the user not to move their head while the devices are added. This is necessary because a newly added object's holographic panel is added with respect to the user's head position (i.e., the camera of the HL2). Once the system has calculated the panel's position for each device, it is shown to the user next to the physical object in MR. If the *CV Module* is active, the panel shows the class name of the detected object. If the *BLE Module* is active, the user in addition sees the device's ID (i.e. the BLE sensor tag's MAC address), values from the BLE sensor tag's sensors (e.g., humidity, or temperature). If the *Thing Module* is active, then – in addition – real-time data from the device is visualized in the panel. In the following, we detail how BLEARVIS calculates a device panel's position when a new device is added.

3.5.1 AoA-based Position Calculation. The *MR Module* receives the AoA from the *BLE Module* via MQTT which was integrated into the application using *M2MQTT for Unity*[9]. After the AoA is received, the X- and Y-values in the Unity World Space coordinate system[10] are calculated based on the trigonometric equations for calculating a point on a unit circle (see Figure 5):

$$X_{aoa} = \cos(AoA + 90),$$
$$Z_{aoa} = \sin(AoA + 90),$$

In both equations, 90° is added to the received AoA, because an AoA of −90° on the antenna array corresponds to 0° on the unit circle, and an AoA of +90° to 180°, respectively. This results in a 3D vector $v_{aoa} = (X_{aoa}, Y_{aoa}, Z_{aoa})$. The Y-coordinate is always set to 0 because it cannot be deduced from one-dimensional AoA alone.

[6] https://www.w3.org/2019/wot/td
[7] https://www.microsoft.com/en-US/hololens/
[8] https://github.com/microsoft/MixedRealityToolkit-Unity/
[9] https://github.com/gpvigano/M2MqttUnity
[10] The Unity World Space coordinate system is left-handed with Y up.

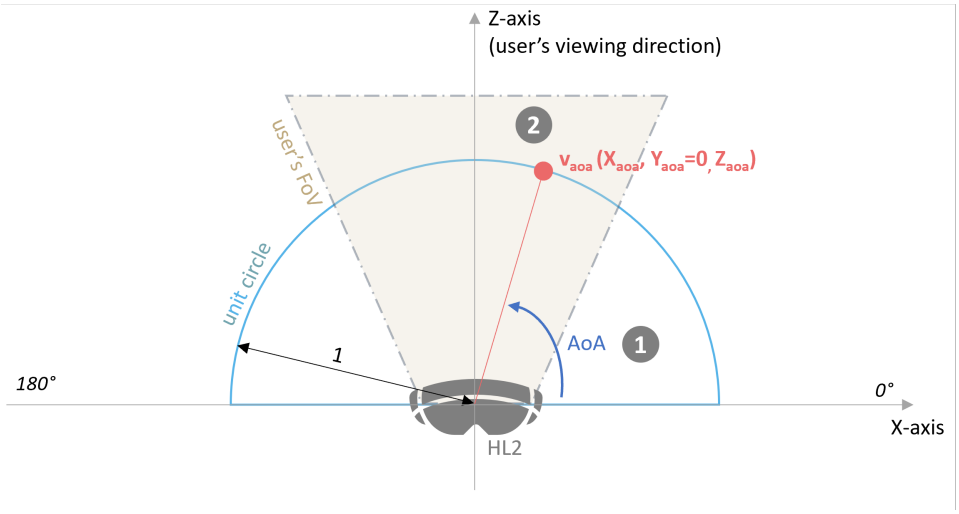


Fig. 5. The 3D-position calculation from the AoA in MR, shown from a bird's eye view of the HL2. (1) The AoA arrives from the *BLE Module*. (2) Then, the two coordinates X_{aoa} and Y_{aoa} are calculated. Y_{aoa} is set to 0. The resulting position v_{aoa} is on an imaginary unit circle of one meter radius centered on the user's head (i.e., the HL2).

3.5.2 CV-based Position Calculation. To calculate 3D positions in MR from the 2D bounding box coordinates that arrive from the *CV Module*, we use the function `Physics.Raycast(Ray direction, out Vector3 hitPosition)`[11] from Unity to cast rays against the Spatial Object Mesh (i.e. the geometry of the user's environment) created by MRTK's Spatial Awareness System[12]. We can compute the `Ray direction` from the 2D position, while `Vector3 hitPosition` is the resulting 3D position in MR. Before this function is used, the *CV Module* returns the coordinates of the detected object's bounding box as two pairs of coordinates: the top-left and the bottom-right corner of the bounding box. These are 2D vectors on the video frame (1280 × 720px) from the HL2's frontal camera that the *CV Module* gets as input. Then, also the other two corners and the center of the bounding box are calculated. The corner vectors are now linearly interpolated to be 40% closer to the center of the bounding box. This increases the chances that we get the 3D positions of these vectors, as we assume, that the object does not fill the complete bounding box, but the center of the bounding box likely corresponds to the physical object's location (see Figure 6). The MR app has a virtual camera object with a RenderTexture attached, where the RenderTexture has the exact dimensions as the video frame used in the *CV Module*. A Camera-GameObject in Unity has the function `Camera.ScreenPointToRay(Vector3 position)` which allows the transformation of a 2D vector on a RenderTexture to a ray in Unity World Space. With this function, we can create the rays we need for the aforementioned `Physics.Raycast()` function. We apply this function to each of the five points. The resulting rays can then be cast against Spatial Object Mesh with the raycasting function. We cast five rays, one for each corner and one for the center of the bounding box. Each Raycast returns a 3D Vector in Unity World Space, given that it returns a hit point. These hit points are then converted to the Camera's local space using the function `Camera.InverseTransformPoint(Vector3 position)`. This step is necessary so that we get a position that is relative to the user's head (i.e. the HL2's camera.) From the resulting (up to five) vectors, we select

[11] See https://docs.unity3d.com/ScriptReference/ for details about this and the other utilized methods.
[12] https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/spatial-awareness/spatial-awareness-getting-started

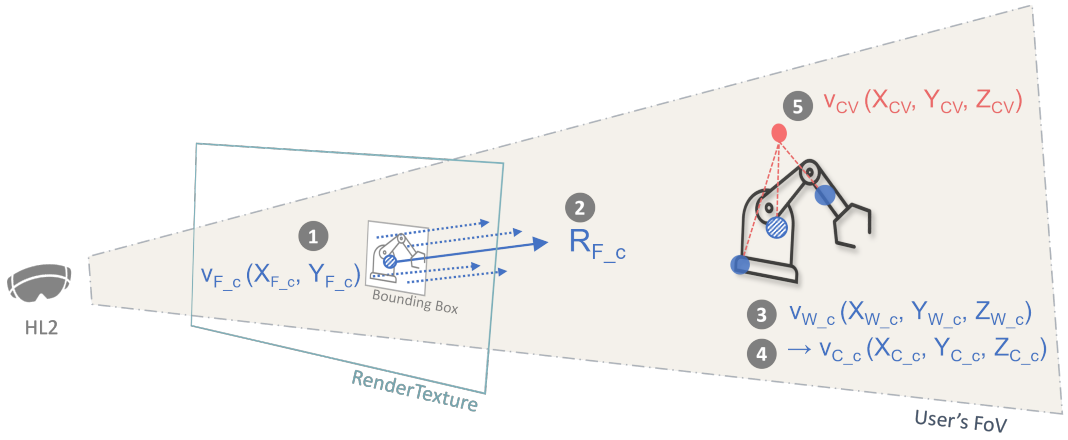


Fig. 6. The 3D-position calculation from the object detection's bounding box in MR. The user is looking from left to right with an approximate field of view of the cone-shaped area. The RenderTexture and the rays are not visible to the user, as they are only internal components of the application. Rather, the user only sees the robot and the panel, whose position in MR is determined as follows (exemplary for the center point of the bounding box in screen coordinates v_{F_c}, the striped blue dot in the bounding box): (1) The 2D vector v_{F_c} is converted into a ray R_{F_c} (solid blue line). (2) This ray is then cast via the Raycast-Method. (3) If R_{F_c} hits the surface it returns a hit point v_{W_c} (striped blue dot on the right) in Unity world coordinates. (4) v_{W_c} is then converted to v_{C_c} in the HL2's camera coordinates. (5) The first four steps are repeated for the corners of the bounding box. Then, v_{CV} (red dot) is calculated from the hit points. In this case, 3 out of 5 rays have hit the object.

the mean X-Value as X_{cv} and the highest Y-Value. In addition, we store the bounding box's height as Y_{cv}, and the Z-value that is closest to the user (but beyond a threshold of 0.5m away from the user) as Z_{cv}. This results in a 3-dimensional vector $v_{cv} = (X_{cv}, Y_{cv}, Z_{cv})$ that is slightly above the actual physical object so that the panel will not occlude the device when it is displayed.

3.5.3 Positioning Content in MR.. When new devices are added, the procedures described in Section 3.5.1, or Section 3.5.2, or both (depending on which modules are available) are repeated multiple times for each method per device resulting in a list of vectors. When multiple devices of the same class are added simultaneously, the calculated vectors must first be assigned to the respective devices. For the AoA-based vectors, this is trivial since the MAC addresses of each BLE tag are attached to each incoming AoA. Yet, for the CV-based vectors we need an extra step that takes all the vectors with the same class name attached as input and separates them by a spatial threshold. This function outputs a list of vectors per device with at least five vectors each. Then, an average of the list of vectors (v_{cv_avg} or v_{aoa_avg}) is calculated for each device to smooth eventual inaccuracies. When both, v_{cv_avg} and v_{aoa_avg}, are available, a new vector (v_{avg}) is linearly interpolated between the two. The assumption is that v_{cv_avg} is closer to the physical object than v_{aoa_avg} in most cases, so v_{avg} is interpolated 20% closer to o_{cv}. The resulting vector (v_{cv_avg}, v_{aoa_avg}, or v_{avg}) is then forwarded to the Position Handler, where a panel is created for each device and then positioned with the help of an MRTK Orbital Solver. Solvers in MRTK facilitate the calculation of an object's position and orientation according to a given algorithm[13]. The Orbital Solver allows to place the panel as if it was on an orbit around the user's head. The values of the previously calculated vector v_{cv_avg} or v_{aoa_avg} are then inserted as the Orbital Solver's *Local Offset* values and thereby placed relatively to the

[13] https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/solvers/solver

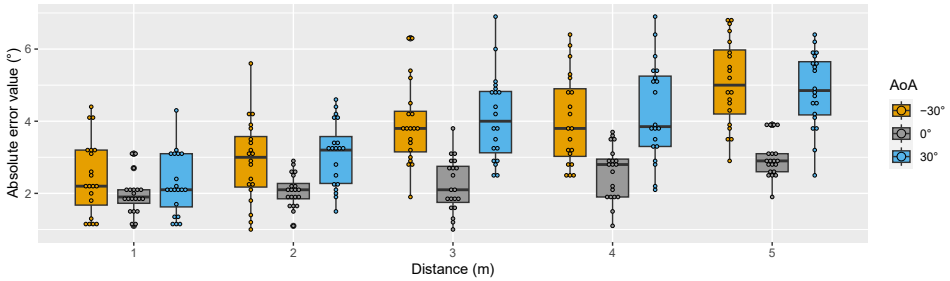


Fig. 7. Error estimation for the BLE-based AoA estimation at specific distances varying from 1 to 5 meters.

HL2's camera (or the user's head). After the panel is created and positioned, the Orbital Solver is deactivated, otherwise, the panel would move its position if the user moved their head. The described method allows us to position the panel correctly in front of the user regardless of their head's current orientation. To ensure that the panel's front is always facing the user, we use MRTK's *Billboard*[14] script which keeps a GameObject always oriented towards the user. Now, the panel will stay at this position independent of the user's movements. The panel displays the class name (if the CV Module is active) and the MAC address of the BLE tag as an ID (if the BLE Module is active). The user can now choose to add new devices.

4 EVALUATION & DEMONSTRATION

In the following, we report evaluation results on each of the modules of BLEARVIS. Subsequently, we demonstrate the applicability of BLEARVIS in two use case scenarios.

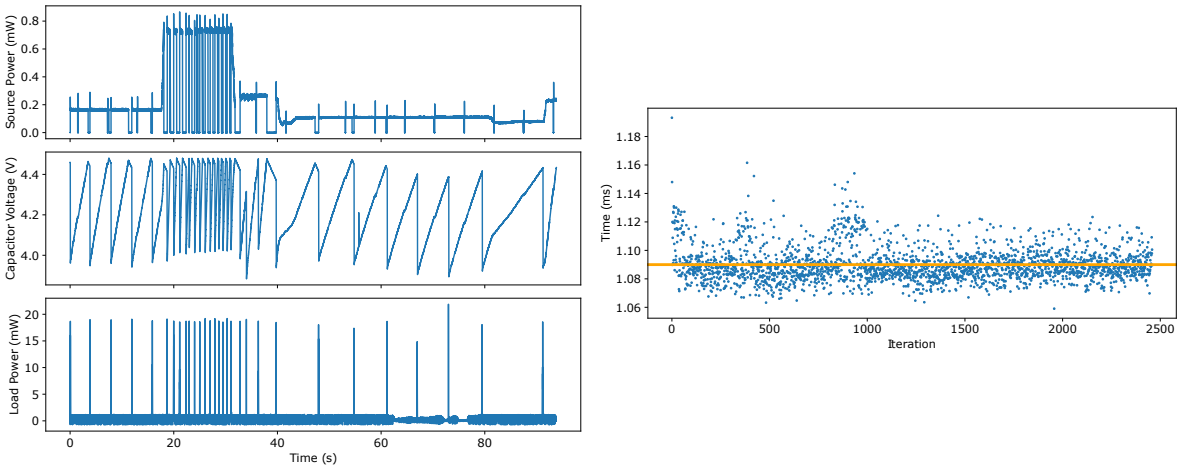
4.1 Evaluation of the BLE Module

4.1.1 Accuracy of the AoA Estimation. To evaluate the error of the *BLE Module*'s AoA estimation, we drew three lines diverging from a single point in space at three angles on a flat surface: -30°, 0°, 30°. On the point, we place the antenna array directed towards the lines and capture the estimated angle, positioning the tag at five different distances (1 to 5m) on each line. The tag is positioned vertically and always directed towards the antenna array. We collected 20 samples for each position and calculated the absolute errors, which are shown in Figure 7. It can be seen that the error grows with increasing distance for all angles. However, the average deviations stay at acceptable levels: 16% for 30°and -30°; and only 10% for 0°. In the same setting, the algorithm demonstrated the same accuracy results for multiple-tag experiments, successfully distinguishing different transmitters by their ID.

In this experiment, we did not observe issues related to spatial aliasing. While using BLEARVIS this is even less of a problem because of the non-static nature of the receiver (i.e., the antenna array is mounted on the user's head), and thus increases the probability for the sample to hit the antennas directly and not the space between them. With more dedicated antenna arrays (see Section 3.2.2), potential spatial aliasing issues can be mitigated even for more demanding applications. We refer readers to recent studies presenting details on how AoA estimation accuracy is affected by different parameters, such as the size and the number of antenna patches [64], the spacing between antenna patches [44], and the number of transmitters [1, 48].

4.1.2 Calculation Time Performance of the AoA Estimator. To measure the AoA calculation time, we executed 2500 estimations on the Raspberry Pi 3B platform. Figure 8b shows that the mean execution time is 1.09 ms and the maximum value can reach up to only 1.20 ms. This timing shows that given the transmission rate of 10Hz, the

[14] https://learn.microsoft.com/en-us/dotnet/api/microsoft.mixedreality.toolkit.ui.billboard



(a) (b)

Fig. 8. The power consumption and generated voltage of the BLE tag (a) and the calculation time the AoA estimator needed for each sample within 2500 iterations (b).

AoA Estimator has the potential to communicate with circa 90 BLE Tags simultaneously. With a more powerful host device, this value could be increased; however, this would affect the component's power consumption. As reference, with our current setup, the battery-powered antenna array and host device consume 0.2 A and 2 A of current, respectively.

Table 1. Power and energy consumption results of the BLE Tag.

Total transmissions	Transmit energy	LPM power	LPM current	Max power	Max current
66	87.963 µJ	8.887 µW	3.582 µA	21.883 mW	8.836 mA

4.1.3 Energy Consumption of the BLE Tag. Given the restricted energy budget allocated for the BLE Tag in the target applications, we evaluated the energy consumption of the BLE tag under an energy-harvesting scenario. We connected the EFR32BG22 Thunderboard to a standard Epishine solar cell [21] and a Mirocard [28] used as an electronic measuring unit with RocketLogger [76] as a data collector. As shown in Table 1, we activated 66 transmissions that consume 87.963 µJ in total. In the idle state, we transitioned the tag to a low-power mode (LPM) that costs only 8.887 µW and 3.582 µA of power and current, respectively. The maximum power and current consumption captured during the experiment were 21.883 mW and 8.836 mA, respectively. Figure 8a shows the dynamics of the source power (top), the capacitor voltage (center), and the load power (bottom) over time for the part of the experimental transmissions. As can be seen, the transmission from the tag is triggered when the capacitor voltage reaches 4.4 V followed by the transition to LPM and capacitor charging. When the source power is increased (19-35 s) (i.e., the solar cell receives more light), the transmission frequency also increases due to the decreased charging time. In essence, our experiment shows that the tag sampling rate depends on the availability of ambient power, whereas the energy consumption of a single transmission is only 1.35 µJ. With lower input power, the *BLE Module* might suffer from low accuracy. Conversely, higher input power would lead to an increase

in the tag sampling rate leading to higher accuracy of the AoA estimate. Thus, the tag's contribution to the energy consumption of the entire system can be estimated by counting the number of transmissions performed during the application lifecycle.

Table 2. Metrics resulting from testing our custom-trained YOLOv7 object detector with a validation set of 50 images per class for two classes

Class	Images	Precision	Recall	mAP@0.5	mAP@0.5:.95
all	100	0.99	0.982	0.995	0.831

4.2 Evaluation of the CV Module

To demonstrate our approach, we chose two types of robots while we also considered using everyday objects such as desk lamps and laptop computers. However, we are interested in objects that offer further information about their status and operation so that they can be integrated into industrial and smart farming use cases. Thus, we trained YOLOv7 to recognize two custom classes, namely small farming robots (i.e., tractorbots), which can be easily transported from one room to another, and articulated robotic arms, which are set up in our laboratory.

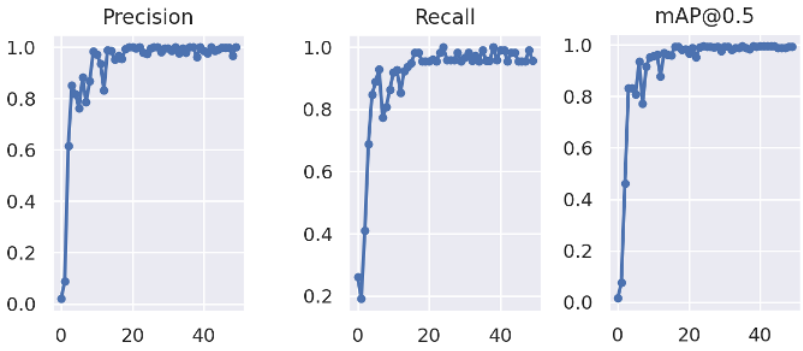


Fig. 9. Precision, Recall and mean Average Precision (mAP@0.5) of the training set over 50 epochs

For the training of the *CV Module*'s object detector, we took advantage of pre-trained weights on the COCO dataset[15]. The created training set for our two classes consists of 320 labeled images per class and a validation set of 50 labeled images per class. The training was done in an NVIDIA Tesla T4 GPU. Figure 9 shows the Precision, Recall, and mean Average Precision (mAP) on an Intersection over Union $IoU = 0.5$ metrics over the 50 epochs of training. The test over the validation set resulted in (see Table 2) a $mAP@.5 = 0.995$ over $IoU = 0.5$ and a $mAP@.5 : .95 = 0.831$ over $IoU = [0.5, 0.95]$. Once the training was completed, the weights were transformed from the output format of YOLOv7 (PyTorch) to the open standard Open Neural Network Exchange (ONNX) format. The ONNX weights were then used for real-time object detection in the *CV Module*.

[15] https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_training.pt

4.3 Demonstration of the MR Module

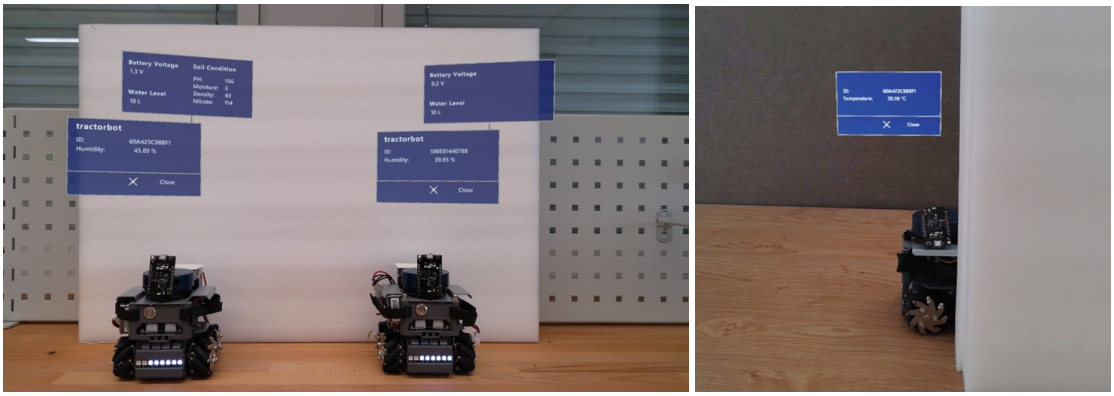
The accuracies of the *BLE Module* and the *CV Module* affect the accuracy of the MR content's placement in the *MR Module*. Yet, the accuracy of the MR panels' placement is also a rather subjective matter since it depends on users' characteristics and preferences (e.g., users' height and distance to the object, users' current task, or their personal preference) or their environment (e.g., illumination conditions, the number of devices around the user, or the size of the objects). Thus, the perception of an accurately placed panel in MR might differ from user to user. While it may be acceptable in certain situations that the panel is slightly next to a device when there is only a single object in the user's field of view, this might be problematic when there are multiple close-by objects in the scene. Therefore, an acceptable accuracy of the panel's placement in MR can only be studied for a specific user group, task, and environment. The following two scenarios, therefore, demonstrate the efficacy and features of our approach, which fuses object situation information from two *Object Situation Modules* (OSM), a visual object detection module (*CV module*) and a BLE-based direction finding and identification module (*BLE Module*). These scenarios show that BLEARVIS indeed provides benefits across multiple applications.

4.4 Smart Farming Scenario

In recent years, the smart farming field has gained the attention of researchers, practitioners, and governmental entities [3]. For instance, in the year 2019, 24 countries in the European Union signed the declaration on *A smart and sustainable digital future for European agriculture and rural areas*[16], in which robotics, artificial intelligence, and the Internet of Things (among others) were technologies highlighted as of special interest, given that such technologies can optimize agricultural activities, support decision making, and in consequence put Europe on the forefront of food production. In light of this interest and given the multiple activities that unmanned aerial vehicles (e.g., remote sensing for mapping soil properties, classification of species, and crop growth [59]) and unmanned ground vehicles (e.g., tilling and harvesting [65]) can perform in a farm context, we envision environments in which farmers work together with several vehicles (robots), and need to be kept in the loop about their status and operations, and about the status of the crops they monitor and work on. To this end and given the manual nature of farming, we propose to use BLEARVIS to provide farmers with hands-free identification of and interaction with robots that might be visually identical but are assigned to do different jobs.

Hence, in a smart farm equipped with autonomous tractors ('tractorbots') in charge of fertilizing the soil, monitoring its condition, and irrigating the crops, such tractorbots need to be regularly checked to verify their status and detect if they might need maintenance. Emilia, a farmer in charge of two identically-looking tractorbots (currently parked in the barn) needs to quickly make sure that the two tractorbots are sufficiently charged so they can be sent to perform a fertilizing task. Luckily, the tractorbots are W3C WoT enabled and their programming interfaces are described in their TDs. Thus, when Emilia enters the garage, wearing her MR HMD, she can use BLEARVIS to verify the tractorbots' status by simply looking at them (see Figure 10a). In this case, the *CV Module* and the *BLE Module* work together to identify each tractorbot; and the *Thing Module* fetches the current battery status of each robot. However, sometimes the tractorbots park behind objects, impeding the BLEARVIS's *CV Module* from accurately detecting them. This is not a problem for BLEARVIS, as through the *BLE Module*, the ID of the robot and its current battery status can still be shown to Emilia (see Figure 10b). In our current implementation, we use the tractorbot's Property Affordances to access its current sensor data (e.g., battery charge and water levels). However, this can be easily expanded to control the robot, by utilizing the tractorbot's Action Affordances. Hence, Emilia could, for instance, use BLEARVIS to send a tractorbot on a fertilizing task.

[16] https://digital-strategy.ec.europa.eu/en/news/eu-member-states-join-forces-digitalisation-european-agriculture-and-rural-areas



(a) (b)

Fig. 10. Demonstration of BLEARVIS in a Smart Farming Scenario. (a) Our system can identify two tractorbots that are visually identical and display information about the environment (e.g., room temperature) and the devices (e.g., battery voltage). (b) This is also possible for a (partially) occluded tractorbot thanks to the *BLE Module*.

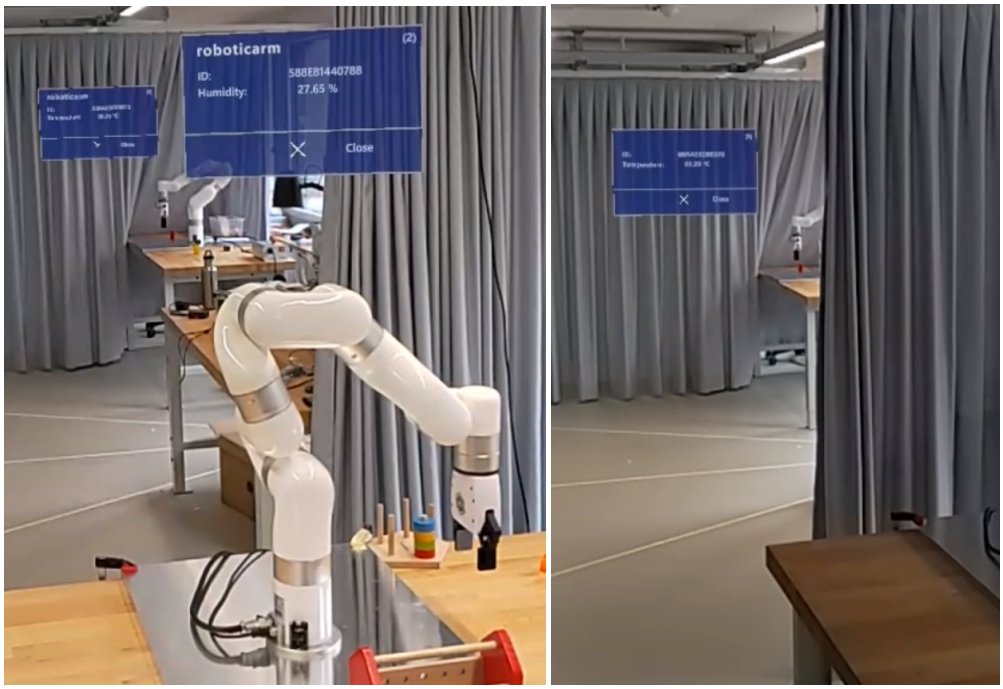
4.5 Smart Factory Scenario

In manufacturing, conflicts between human safety and productivity are still common, while human operators' roles have shifted towards collaborating with and utilizing new technologies [70]. These new technologies, such as the proposed BLEARVIS system, can help increase productivity without compromising workers' safety and health in the context of Industry 4.0.

Francis is a manager working in a smart factory together with collaborative robots (i.e., articulated robotic arms) and other human colleagues. In his daily tasks, Francis needs to be promptly informed about the current and past processes, as well as of the performance, of several identically looking robotic arms to make fast decisions regarding shop floor operations. Sometimes, Francis also needs to control the robotic arms, for instance, to adjust their speed. To make his job more efficient, Francis wears an MR HMD and uses BLEARVIS. Given the shop floor's physical layout, two robotic arms can be in the same line of sight, making the AoA the same for both robots. Yet, thanks to the distance estimation of the *MR Module*, which is computed based on the CV-aided object detection, BLEARVIS is capable of computing the placement of the panels close enough to each robotic arm such that Francis can visually distinguish which panel belongs to which robotic arm (see Figure 11a). Moreover, maintenance operations on such robotic arms are regularly performed by Francis' (human) colleagues. When this happens, the robot technician usually closes the curtain installed next to the robot so that he can focus on his task. In these cases, Francis might only see a small part of the robot. However, he likes to be informed about the type of maintenance and the remaining time the robot will be down. BLEARVIS's *BLE Module* and its *Thing Module* allow Francis to receive information about the current condition of the robot and even the environment (e.g., the hours of operation and room temperature; see Figure 11b). Since BLEARVIS allows Francis to have quick and effortless access to the robotic arms' status, his cognitive workload and stress level remains lower than if he would have to look up each robot's information on a tablet and would have to identify the robots to which the information belongs to manually.

4.6 Other Applications

The main strength of BLEARVIS is its object situation fusion capability, which makes it ideal for the identification of identical-looking objects. Aside from the scenarios presented in the preceding sections, BLEARVIS could be



(a) (b)

Fig. 11. Demonstration of BLEARVIS in a Smart Factory Scenario. (a) Our system enables users to distinguish information about two visually identical robots when both robots are within the line of sight. (b) BLEARVIS can identify a robot even if the robot is partially occluded and the BLE tag is fully occluded.

implemented in environments teemed with identically looking robots engaged in continuous tasks. Such is the case of robotic fulfillment centers [29], in which mobile robots transport movable shelves from storage locations to zones conveniently accessible by human packers, streamlining the packaging and delivery process. In such fulfillment centers, human workers responsible for robot maintenance could effortlessly inspect the status of robots using BLEARVIS with just a glance. A warehouse manager could get immediate information about the storage zones most visited by a specific robot by simply looking at the robot. Furthermore, BLEARVIS holds potential in monitoring equipment within smart cities or on a smaller scale in smart buildings. For example, leveraging its two OSMs, BLEARVIS enables users to "look through walls" to locate the spaces in which smart meters have been installed within a building [40]. Once these smart meters become visible to the BLEARVIS user, pertinent information regarding the equipment's health and performance could be displayed. Similarly, in large parking lots equipped with identical-looking electric vehicle chargers, BLEARVIS could streamline the process of identifying units requiring maintenance.

5 DISCUSSION

The ODIF architecture permits the vertically and horizontally modular fusion of object situation information from heterogeneous sources. We demonstrate this architecture through the implementation of a prototype, BLEARVIS, that can identify objects through a combination of vision-based object detection and RF-based AoA estimation. By integrating both technologies, our system can detect and identify objects at different distances and angles from the user. Furthermore, the system can still identify an occluded object, and it can differentiate two identically looking objects. BLEARVIS visualizes the class name and identifier for a detected object in MR in a way that is spatially aligned with the physical objects. The system is additionally capable of visualizing real-time data from the identified object and hence can render suitable user interfaces to permit users to interact with the devices. Our prototype makes use of well-established technologies (e.g., MUSIC, YOLOv7, and MQTT), yet these can be easily exchanged based on ODIF's *vertical modularity*. We evaluated the two OSMs of BLEARVIS and presented two scenarios that demonstrate the features of our system and possible applications. ODIF's *horizontal modularity* additionally allows the extension of the system with other OSMs so that it may satisfy the requirements of additional applications and scenarios.

Furthermore, our system is not without limitations. In the current version of BLEARVIS, the method we use to convert 2D points to 3D points in the *MR Module* (*Raycasting)* is susceptible to the target object's shape: If the object is box-shaped then our implemented raycasting method will most likely hit the object at least once. However, if the object features holes at the places that BLEARVIS samples with its raycasts, none of the rays might hit. This could be remedied by casting more rays to increase the chances of hitting the object's representation in the Spatial Object mesh, which, however, undermines the performance of our system. As a further limitation, to enable operation of the BLEARVIS *BLE Module* it is required that target objects are outfitted with suitable active tags, which increases the cost of these objects and where the tags might not fit into (or onto) the objects. While more and more target devices are already equipped with such tags, this conceptual limitation of course remains; however, we believe that BLEARVIS could motivate further research into AoA estimation for passive tags [84], which would overcome this limitation.

We only regarded static devices in the current implementation of BLEARVIS, as we encountered difficulties in updating a panel's position based on the *CV Module*'s information. As a purely CV-based solution cannot easily trace one object from one frame to another, when an object moves its position between two frames, this calls for incorporating CV methods capable of finding relationships between objects, such as Scene Graphs [37]. However, we plan to implement such a method in the future and thus extend the system to display MR content that moves along with mobile objects reliably. In our smart farming scenario, this would enable the farmer to not only use BLEARVIS while the tractorbots are parked but also when they are in movement, e.g., driving on a field. Emilia could then see real-time data from the tractorbots in an MR panel that moves with the physical object. Additionally, use cases involving other dynamic objects, such as drones could be unleashed. Specifically in the smart farming scenario, identifying drones and having access to the information their sensors (e.g., cameras) produce, could allow farmers to access the latest status of a specific patch of land, by just looking at the drone currently flying over it.

As future work, we also plan to include elements in the MR panels that allow users to interact with the object instead of only seeing the object's data visualized. This could, for instance, enable Francis in the Smart Factory scenario to also send basic commands to a robot through BLEARVIS, e.g., to move its gripper to a specific position. This could be easily achieved by implementing functions that take advantage of the W3C WoT Action Affordances (in addition to the implemented Property Affordances).

Furthermore, our current prototype has ergonomic constraints as it puts pressure on the user's head due to the antenna array's shape and weight. Enabling users to wear the MR HMD with the antenna array mounted for a

prolonged period of time would require a significantly lighter and smaller antenna array, such as the U-Blox ANT-B11 (see Sect. 3.2.2).

6 CONCLUSION

In this work, we proposed ODIF, an architecture that permits the fusion of object situation information from heterogeneous sources and that remains vertically and horizontally modular to allow extending and upgrading systems that are built according to this architecture. We furthermore presented BLEARVIS, a system built according to ODIF. BLEARVIS allows the identification and localization of objects based on two OSMs, an integration of RF-based AoA estimation and vision-based object detection. With this integration, BLEARVIS takes advantage of the strengths of both these approaches, while the technologies partially compensate for each other's weaknesses. BLEARVIS can be useful in situations where multiple visually identical devices in a ubiquitous computing environment need to be differentiated to enable users to interact efficiently with them, such as in a smart factory or smart farming scenario. The system furthermore enables the display of dynamic data from these devices in a way that is spatially aligned with the respective object in MR, and enables interaction possibilities with these devices through standard interface descriptions. Fundamentally, we see ODIF as a potential starting point to spawn the integration of diverse object detection, classification, and identification approaches that function across the electromagnetic (and possibly audio) spectrum. We argue that the specific benefits of different technologies, which are fundamentally based on the propagation properties of their underlying carriers, could be combined based on ODIF's horizontal modularity to yield fused object interaction systems that can be integrated into head-worn devices, as demonstrated with BLEARVIS.

ACKNOWLEDGMENTS

This project was supported by the Swiss Innovation Agency Innosuisse (Project #48342.1 IP-ICT) and the Basic Research Fund of the University of St.Gallen.

REFERENCES

[1] Amer, Mohammed M T and Atteya, Khadeeja Gamal Mohammed. 2020. Indoor Positioning Bluetooth Angle of Arrival. http://lup.lub.lu.se/student-papers/record/9005557

[2] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. 2001. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications* 21, 6 (2001), 34–47. https://doi.org/10.1109/38.963459

[3] Manlio Bacco, Paolo Barsocchi, Erina Ferro, Alberto Gotta, and Massimiliano Ruggeri. 2019. The Digitisation of Agriculture: a Survey of Research Activities on Smart Farming. *Array* 3-4 (Sept. 2019), 100009. https://doi.org/10.1016/j.array.2019.100009

[4] Michael Barz, Sebastian Kapp, Jochen Kuhn, and Daniel Sonntag. 2021. Automatic Recognition and Augmentation of Attended Objects in Real-Time Using Eye Tracking and a Head-Mounted Display. In *ACM Symposium on Eye Tracking Research and Applications* (Virtual Event, Germany) *(ETRA '21 Adjunct)*. ACM, New York, NY, USA, Article 3, 4 pages. https://doi.org/10.1145/3450341.3458766

[5] Kenan Bektas. 2020. Toward A Pervasive Gaze-Contingent Assistance System: Attention and Context-Awareness in Augmented Reality. In *ACM Symposium on Eye Tracking Research and Applications* (Stuttgart, Germany) *(ETRA '20 Adjunct)*. ACM, New York, NY, USA, Article 36, 3 pages. https://doi.org/10.1145/3379157.3391657

[6] Kenan Bektas, Jannis Rene Strecker, Simon Mayer, and Markus Stolze. 2022. EToS-1: Eye Tracking on Shopfloors for User Engagement with Automation. In *AutomationXP22: Engaging with Automation, CHI'22*. CEUR Workshop Proceedings. http://www.alexandria.unisg.ch/266339/

[7] Md Fazlay Rabbi Masum Billah, Md Mofijul Islam, Nurani Saoda, Tariq Iqbal, and Bradford Campbell. 2022. Fusing Computer Vision and Wireless Signal for Accurate Sensor Localization in AR View. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems* (Boston, Massachusetts) *(SenSys '22)*. Association for Computing Machinery, New York, NY, USA, 823–824. https://doi.org/10.1145/3560905.3568095

[8] Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Foundations and Trends® in Human–Computer Interaction* 8, 2-3 (2015), 73–272. https://doi.org/10.1561/1100000049

[9] Camille M Blondin, Ekaterina Ivanova, Jonathan Eden, and Etienne Burdet. 2021. Perception and Performance of Electrical Stimulation for Proprioception. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE,

4550–4554. https://doi.org/10.1109/EMBC46164.2021.9630186
[10] Bluetooth Special Interest Group. 2019. Bluetooth Specification Version 5.1. https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/.
[11] Noori BniLam, Dennis Joosens, Michiel Aernouts, Jan Steckel, and Maarten Weyn. 2021. LoRay: AoA Estimation System for Long Range Communication Networks. *IEEE Transactions on Wireless Communications* 20, 3 (Mar 2021), 2005–2018. https://doi.org/10.1109/TWC.2020.3038565
[12] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR* abs/2004.10934 (2020). arXiv:2004.10934 https://arxiv.org/abs/2004.10934
[13] Roger Boldu, Alexandru Dancu, Denys J.C. Matthies, Thisum Buddhika, Shamane Siriwardhana, and Suranga Nanayakkara. 2018. FingerReader2.0: Designing and Evaluating a Wearable Finger-Worn Camera to Assist People with Visual Impairments While Shopping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 94 (sep 2018), 19 pages. https://doi.org/10.1145/3264904
[14] Tara Boroushaki, Isaac Perper, Mergen Nachin, Alberto Rodriguez, and Fadel Adib. 2021. RFusion: Robotic Grasping via RF-Visual Sensing and Learning. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (Coimbra Portugal, 2021-11-15). ACM, 192–205. https://doi.org/10.1145/3485730.3485944
[15] John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6 (1986), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851
[16] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 129 (jan 2018), 27 pages. https://doi.org/10.1145/3161173
[17] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177
[18] Joseph DeGol, Timothy Bretl, and Derek Hoiem. 2017. ChromaTag: A Colored Marker and Fast Detection Algorithm. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 1481–1490. https://doi.org/10.1109/ICCV.2017.164
[19] Mustafa Doga Dogan, Ahmad Taka, Michael Lu, Yunyi Zhu, Akshat Kumar, Aakar Gupta, and Stefanie Mueller. 2022. InfraredTags: Embedding Invisible AR Markers and Barcodes Using Low-Cost, Infrared-Based 3D Printing and Imaging Tools. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. ACM, New York, NY, USA, Article 269, 12 pages. https://doi.org/10.1145/3491102.3501951
[20] Martin Eckert, Matthias Blex, and C. Friedrich. 2018. Object Detection Featuring 3D Audio Localization for Microsoft HoloLens - A Deep Learning based Sensor Substitution Approach for the Blind. In *International Conference on Health Informatics*.
[21] epishine. 2023. Indoor solar cell. https://www.epishine.com/product.
[22] Linn D. Evjemo, Tone Gjerstad, Esten I. Grøtli, and Gabor Sziebig. 2020. Trends in smart manufacturing: Role of humans and industrial robots in Smart Factories. *Current Robotics Reports* 1, 2 (2020), 35–41. https://doi.org/10.1007/s43154-020-00006-5
[23] M. Fiala. 2005. ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. 590–596 vol. 2. https://doi.org/10.1109/CVPR.2005.74
[24] Mark Fiala. 2007. Webtag: A World Wide Internet Based AR System. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 263–264. https://doi.org/10.1109/ISMAR.2007.4538858
[25] Philipp Fleck, Aimee Sousa Calepso, Sebastian Hubenschmid, Michael Sedlmair, and Dieter Schmalstieg. 2022. RagRug: A Toolkit for Situated Analytics. *IEEE Transactions on Visualization and Computer Graphics* (2022). https://doi.org/10.1109/TVCG.2022.3157058
[26] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, USA, 1440–1448. https://doi.org/10.1109/ICCV.2015.169
[27] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*. IEEE Computer Society, USA, 580–587. https://doi.org/10.1109/CVPR.2014.81
[28] Andres Gomez. 2020. On-Demand Communication with the Batteryless MiroCard: Demo Abstract. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (Virtual Event, Japan) *(SenSys '20)*. ACM, New York, NY, USA, 629–630. https://doi.org/10.1145/3384419.3430440
[29] Yeming Gong, Mingzhou Jin, and Zhe Yuan. 2020. Robotic mobile fulfilment systems considering customer classes. *International Journal of Production Research* (Dec. 2020), 1–18. https://doi.org/10.1080/00207543.2020.1779370
[30] Valentin Heun, Eva Stern-Rodriguez, Marc Teyssier, and Pattie Maes. 2016. Reality Editor. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. ACM, New York, NY, USA, 4. https://doi.org/10.1145/2851581.2889431
[31] Jie Hua, Sangsu Lee, Gruia-Catalin Roman, and Christine Julien. 2021. ArcIoT: Enabling Intuitive Device Control in the Internet of Things through Augmented Reality. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and*

other Affiliated Events (PerCom Workshops). 558–564. https://doi.org/10.1109/PerComWorkshops51409.2021.9431115
[32] Yuta Itoh, Tobias Langlotz, Jonathan Sutton, and Alexander Plopski. 2021. Towards Indistinguishable Augmented Reality: A Survey on Optical See-through Head-Mounted Displays. *ACM Comput. Surv.* 54, 6, Article 120 (2021), 36 pages. https://doi.org/10.1145/3453157
[33] Kay Erik Jenss and Simon Mayer. 2023. QRUco: Interactive QR Codes Through Thermoresponsive Embeddings. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 468, 4 pages. https://doi.org/10.1145/3544549.3583923
[34] Wenchao Jiang, Feng Li, Luoyu Mei, Ruofeng Liu, and Shuai Wang. 2022. VisBLE: Vision-Enhanced BLE Device Tracking. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 217–225. https://doi.org/10.1109/SECON55815.2022.9918581
[35] Dongsik Jo and Gerard Jounghyun Kim. 2019. AR Enabled IoT for a Smart and Interactive Environment: A Survey and Future Directions. *Sensors* 19, 19 (2019), 4330. https://doi.org/10.3390/s19194330
[36] Dongsik Jo and Gerard Jounghyun Kim. 2019. IoT + AR: Pervasive and Augmented Environments for "Digi-log" Shopping Experience. *Human-centric Computing and Information Sciences* 9, 1 (2019), 1. https://doi.org/10.1186/s13673-018-0162-5
[37] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3668–3678.
[38] Tai-Wei Kan, Chin-Hung Teng, and Wen-Shou Chou. 2009. Applying QR Code in Augmented Reality Applications. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and Its Applications in Industry* (Yokohama, Japan) *(VRCAI '09)*. ACM, New York, NY, USA, 253–257. https://doi.org/10.1145/1670252.1670305
[39] H. Kato and M. Billinghurst. 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. 85–94. https://doi.org/10.1109/IWAR.1999.803809
[40] Hakpyeong Kim, Heeju Choi, Hyuna Kang, Jongbaek An, Seungkeun Yeom, and Taehoon Hong. 2021. A systematic review of the smart energy conservation system: From smart homes to sustainable smart cities. *Renewable and Sustainable Energy Reviews* 140 (2021), 110755. https://doi.org/10.1016/j.rser.2021.110755
[41] Jung-Hwa Kim, Seung-June Choi, and Jin-Woo Jeong. 2019. Watch & Do: A Smart IoT Interaction System with Object Detection and Gaze Estimation. *IEEE Transactions on Consumer Electronics* 65, 2 (2019), 195–204. https://doi.org/10.1109/TCE.2019.2897758
[42] Hamid Krim and Mats Viberg. 1996. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine* 13, 4 (1996), 67–94. https://doi.org/10.1109/79.526899
[43] Debasis Kundu. 1996. Modified MUSIC Algorithm for Estimation DOA of Signals. *Signal Process.* 48, 1 (Jan 1996), 85–90. https://doi.org/10.1016/0165-1684(95)00126-3
[44] Långberg, Felix and Thurborg, Jonas. 2020. Design of a Size Reduced Antenna Array for Angle of Arrival (AoA) Estimation for BLE 5.1. https://lup.lub.lu.se/student-papers/search/publication/9031557
[45] Silicon Laboratories. 2021. *Custom Direction-Finding Solutions using the Silicon Labs Bluetooth Stack*. Technical Report. Silicon Laboratories. https://www.silabs.com/documents/public/application-notes/an1297-custom-direction-finding-solutions-silicon-labs-bluetooth.pdf
[46] Silicon Laboratories. 2022. iBeacon. https://docs.silabs.com/rs9116-wiseconnect/2.6/wifibt-wc-snippet-examples/ble-ble-ibeacon-readme#i-beacon
[47] Mei-Kei Lai and Yan Yan Cao. 2019. Designing Interactive Olfactory Experience in Real Context and Applications. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, Tempe Arizona USA, 703–706. https://doi.org/10.1145/3294109.3295659
[48] Qianfeng Lin, Jooyoung Son, and Hyeongseol Shin. 2023. A self-learning mean optimization filter to improve bluetooth 5.1 AoA indoor positioning accuracy for ship environments. *Journal of King Saud University - Computer and Information Sciences* 35, 3 (2023), 59–73. https://doi.org/10.1016/j.jksuci.2023.01.019
[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
[50] Simon Mayer, Yassin N. Hassan, and Gábor Sörös. 2014. A Magic Lens for Revealing Device Interactions in Smart Environments. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications* (Shenzhen, China) *(SA '14)*. ACM, New York, NY, USA, Article 9, 6 pages. https://doi.org/10.1145/2669062.2669077
[51] Simon Mayer, Markus Schalch, Marian George, and Gábor Sörös. 2013. Device Recognition for Intuitive Interaction with the Web of Things. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication* (Zurich, Switzerland) *(UbiComp '13 Adjunct)*. ACM, New York, NY, USA, 239–242. https://doi.org/10.1145/2494091.2494168
[52] Simon Mayer and Gábor Sörös. 2014. User Interface Beaming – Seamless Interaction with Smart Things Using Personal Wearable Computers. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops*. 46–49. https://doi.org/10.1109/BSN.Workshops.2014.17
[53] Konstantinos Michalakis, John Aliprantis, and George Caridakis. 2018. Visualizing the Internet of Things: Naturalizing Human-Computer Interaction by Incorporating AR Features. *IEEE Consumer Electronics Magazine* 7, 3 (2018), 64–72. https://doi.org/10.1109/MCE.2018.

2797638
[54] Paul Milgram and Fumio Kishino. 1994. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems* 77, 12 (1994), 1321–1329.
[55] Alexis Morris, Jie Guan, and Amna Azhar. 2021. An XRI Mixed-Reality Internet-of-Things Architectural Framework Toward Immersive and Adaptive Smart Environments. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (2021-10). 68–74. https://doi.org/10.1109/ISMAR-Adjunct54149.2021.00024
[56] Nahal Norouzi, Gerd Bruder, Brandon Belna, Stefanie Mutter, Damla Turgut, and Greg Welch. 2019. A Systematic Review of the Convergence of Augmented Reality, Intelligent Virtual Agents, and the Internet of Things. In *Artificial Intelligence in IoT*, Fadi Al-Turjman (Ed.). Springer International Publishing, 1–24. https://doi.org/10.1007/978-3-030-04110-6_1
[57] Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. 2015. Virtual Replicas for Remote Assistance in Virtual and Augmented Reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. ACM, New York, NY, USA, 405–415. https://doi.org/10.1145/2807442.2807497
[58] Jason Orlosky, Misha Sra, Kenan Bektaş, Huaishu Peng, Jeeeun Kim, Nataliya Kos'myna, Tobias Höllerer, Anthony Steed, Kiyoshi Kiyokawa, and Kaan Akşit. 2021. Telelife: The Future of Remote Living. *Frontiers in Virtual Reality* 2 (Nov. 2021), 763340. https://doi.org/10.3389/frvir.2021.763340
[59] Lucas Prado Osco, José Marcato Junior, Ana Paula Marques Ramos, Lúcio André de Castro Jorge, Sarah Narges Fatholahi, Jonathan de Andrade Silva, Edson Takashi Matsubara, Hemerson Pistori, Wesley Nunes Gonçalves, and Jonathan Li. 2021. A review on deep learning in UAV remote sensing. *International Journal of Applied Earth Observation and Geoinformation* 102 (Oct. 2021), 102456. https://doi.org/10.1016/j.jag.2021.102456
[60] Kaushik Parida, Hyunwoo Bark, and Pooi See Lee. 2021. Emerging thermal technology enabled augmented reality. *Advanced Functional Materials* 31, 39 (2021), 2007952. https://doi.org/10.1002/adfm.202007952
[61] Yongtae Park, Sangki Yun, and Kyu-Han Kim. 2019. When IoT Met Augmented Reality: Visualizing the Source of the Wireless Signal in AR View. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul Republic of Korea, 2019-06-12). ACM, 117–129. https://doi.org/10.1145/3307334.3326079
[62] Alexander Plopski, Teresa Hirzle, Nahal Norouzi, Long Qian, Gerd Bruder, and Tobias Langlotz. 2022. The Eye in Extended Reality: A Survey on Gaze Interaction and Eye Tracking in Head-worn Extended Reality. *Comput. Surveys* 55, 3 (March 2022), 1–39. https://doi.org/10.1145/3491207
[63] Dominic Potts, Martynas Dabravalskis, and Steven Houben. 2022. TangibleTouch: A Toolkit for Designing Surface-based Gestures for Tangible Interfaces. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3490149.3502263
[64] Theodoros Prokic. 2019. *Antenna Design for Angle of Arrival Measurement in Access Control Applications*. Ph. D. Dissertation. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-247508
[65] E. Fantin Irudaya Raj, M. Appadurai, and K. Athiappan. 2021. Precision Farming in Modern Agriculture. In *Smart Agriculture Automation Using Advanced Technologies: Data Analytics and Machine Learning, Cloud Architecture, Automation and IoT*, Amitava Choudhury, Arindam Biswas, T. P. Singh, and Santanu Kumar Ghosh (Eds.). Springer, Singapore, 61–87. https://doi.org/10.1007/978-981-16-6124-2_4
[66] Janani Ramaswamy. 2020. What Makes the Sound?: A Dual-Modality Interacting Network for Audio-Visual Event Localization. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4372–4376. https://doi.org/10.1109/ICASSP40776.2020.9053895
[67] Nimesha Ranasinghe and Ellen Yi-Luen Do. 2016. Virtual Sweet: Simulating Sweet Sensation Using Thermal Stimulation on the Tip of the Tongue. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16 Adjunct)*. ACM, New York, NY, USA, 127–128. https://doi.org/10.1145/2984751.2985729
[68] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. https://doi.org/10.1109/CVPR.2016.91
[69] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767
[70] Arto Reiman, Jari Kaivo-oja, Elina Parviainen, Esa-Pekka Takala, and Theresa Lauraeus. 2021. Human Factors and Ergonomics in Manufacturing in the Industry 4.0 Context – A Scoping Review. *Technology in Society* 65 (May 2021), 101572. https://doi.org/10.1016/j.techsoc.2021.101572
[71] Jun Rekimoto and Yuji Ayatsuka. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags *(DARE '00)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/354666.354667
[72] Jun Rekimoto and Katashi Nagao. 1995. The World through the Computer: Computer Augmented Interaction with Real World Environments. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology* (Pittsburgh, Pennsylvania, USA) *(UIST '95)*. ACM, New York, NY, USA, 29–36. https://doi.org/10.1145/215585.215639
[73] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems* 28 (2015).

[74] Cassandra Scheirer and Chris Harrison. 2022. DynaTags: Low-Cost Fiducial Marker Mechanisms. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (Bengaluru, India) *(ICMI '22)*. ACM, New York, NY, USA, 432–443. https://doi.org/10.1145/3536221.3556591

[75] Ralph Schmidt. 1986. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (1986), 276–280. https://doi.org/10.1109/TAP.1986.1143830

[76] Lukas Sigrist, Andres Gomez, Roman Lim, Stefan Lippuner, Matthias Leubin, and Lothar Thiele. 2016. RocketLogger: Mobile Power Logger for Prototyping IoT Devices: Demo Abstract. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM* (Stanford, CA, USA) *(SenSys '16)*. ACM, New York, NY, USA, 288–289. https://doi.org/10.1145/2994551.2996526

[77] Janick Spirig, Kimberly Garcia, and Simon Mayer. 2021. An Expert Digital Companion for Working Environments. In *Proceedings of the 11th International Conference on the Internet of Things* (St.Gallen, Switzerland) *(IoT '21)*. ACM, New York, NY, USA, 25–32. https://doi.org/10.1145/3494322.3494326

[78] Misha Sra, Abhinandan Jain, and Pattie Maes. 2019. Adding Proprioceptive Feedback to Virtual Reality Experiences Using Galvanic Vestibular Stimulation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300905

[79] Jannis Strecker, Kimberly García, Kenan Bektaş, Simon Mayer, and Ganesh Ramanathan. 2022. SOCRAR: Semantic OCR through Augmented Reality. In *Proceedings of the 12th International Conference on the Internet of Things*. ACM, Delft Netherlands, 25–32. https://doi.org/10.1145/3567445.3567453

[80] Ivan E. Sutherland. 1968. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. ACM Press, San Francisco, California, 757–764. https://doi.org/10.1145/1476589.1476686

[81] Theophilus Teo, Louise Lawrence, Gun A. Lee, Mark Billinghurst, and Matt Adcock. 2019. Mixed Reality Remote Collaboration Combining 360 Video and 3D Reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300431

[82] Anurag Verma, Surya Prakash, Vishal Srivastava, Anuj Kumar, and Subhas Chandra Mukhopadhyay. 2019. Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review. *IEEE Sensors Journal* 19, 20 (2019), 9036–9046. https://doi.org/10.1109/JSEN.2019.2922409

[83] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv:2207.02696 http://arxiv.org/abs/2207.02696

[84] Zhongqin Wang, J. Andrew Zhang, Fu Xiao, and Min Xu. 2022. Accurate AoA Estimation for RFID Tag Array With Mutual Coupling. *IEEE Internet of Things Journal* 9, 15 (2022), 12954–12972. https://doi.org/10.1109/JIOT.2022.3169328

[85] Lei Xie, Chuyu Wang, Yanling Bu, Jianqiang Sun, Qingliang Cai, Jie Wu, and Sanglu Lu. 2019. TaggedAR: An RFID-Based Approach for Recognition of Multiple Tagged Objects in Augmented Reality Systems. *IEEE Transactions on Mobile Computing* 18, 5 (2019), 1188–1202. https://doi.org/10.1109/TMC.2018.2857812

[86] Jing Yang, Amit Barde, and Mark Billinghurst. 2022. Audio Augmented Reality: A Systematic Review of Technologies, Applications, and Future Research Directions. *Journal of the Audio Engineering Society* 70, 10 (Nov. 2022), 788–809. https://doi.org/10.17743/jaes.2022.0048

[87] Tengxiang Zhang, Zitong Lan, Chenren Xu, Yanrong Li, and Yiqiang Chen. 2023. BLEselect: Gestural IoT Device Selection via Bluetooth Angle of Arrival Estimation from Smart Glasses. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 198 (Jan 2023), 28 pages. https://doi.org/10.1145/3569482

[88] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Hang Zhao, Tianhong Li, Antonio Torralba, and Dina Katabi. 2019. Through-Wall Human Mesh Recovery Using Radio Signals. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 10112–10121. https://doi.org/10.1109/ICCV.2019.01021

[89] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2023. Object Detection in 20 Years: A Survey. *Proc. IEEE* 111, 3 (2023), 257–276. https://doi.org/10.1109/JPROC.2023.3238524