



Frameworks Enabling Ubiquitous Mixed Reality Applications Across Dynamically Adaptable Device Configurations

Sandro Weber*, Linda Rudolph, Sven Liedtke, Christian Eichhorn, Daniel Dyrda, David A. Plecher and Gudrun Klinker

FAR Augmented Reality Research Group, TU Munich, Munich, Germany

If Mixed Reality applications are supposed to become truly ubiquitous, they face the challenge of an ever evolving set of hardware and software systems - each with their own standards and APIs—that need to work together and become part of the same shared environment (the application). A unified standard is unlikely so we can not rely on a single software development stack to incorporate all necessary parts. Instead we need frameworks that are modular and flexible enough to be adapted to the needs of the application at hand and are able to incorporate a wide range of setups for devices, services, etc. We identified a set of common questions that can be used to characterize and analyze Mixed Reality applications and use these same questions to identify challenges as well as present solutions in the form of three frameworks tackling the fields of tracking and inference (UbiTrack), interaction (Ubi-Interact) and visualization (UbiVis). Tracking and inference has been addressed for quite some time now while interaction is a current topic with existing solutions. Visualization will be focused more in the future. We present several applications in development together with their future vision and explain how the frameworks help realize these and other potential apps.

Keywords: mixed reality, tracking, visualization, interaction, frameworks

OPEN ACCESS

Edited by:

Thierry Duval,
IMT Atlantique Bretagne-Pays de la
Loire, France

Reviewed by:

Jérémy Lacoche,
Orange, France
Bruce Hunter Thomas,
University of South Australia, Australia
Jean-Yves Didier,
University of Évreux Val d'Essonne,
France

*Correspondence:

Sandro Weber
webers@in.tum.de

Specialty section:

This article was submitted to
Technologies for VR,
a section of the journal
Frontiers in Virtual Reality

Received: 27 August 2021

Accepted: 08 March 2022

Published: 04 April 2022

Citation:

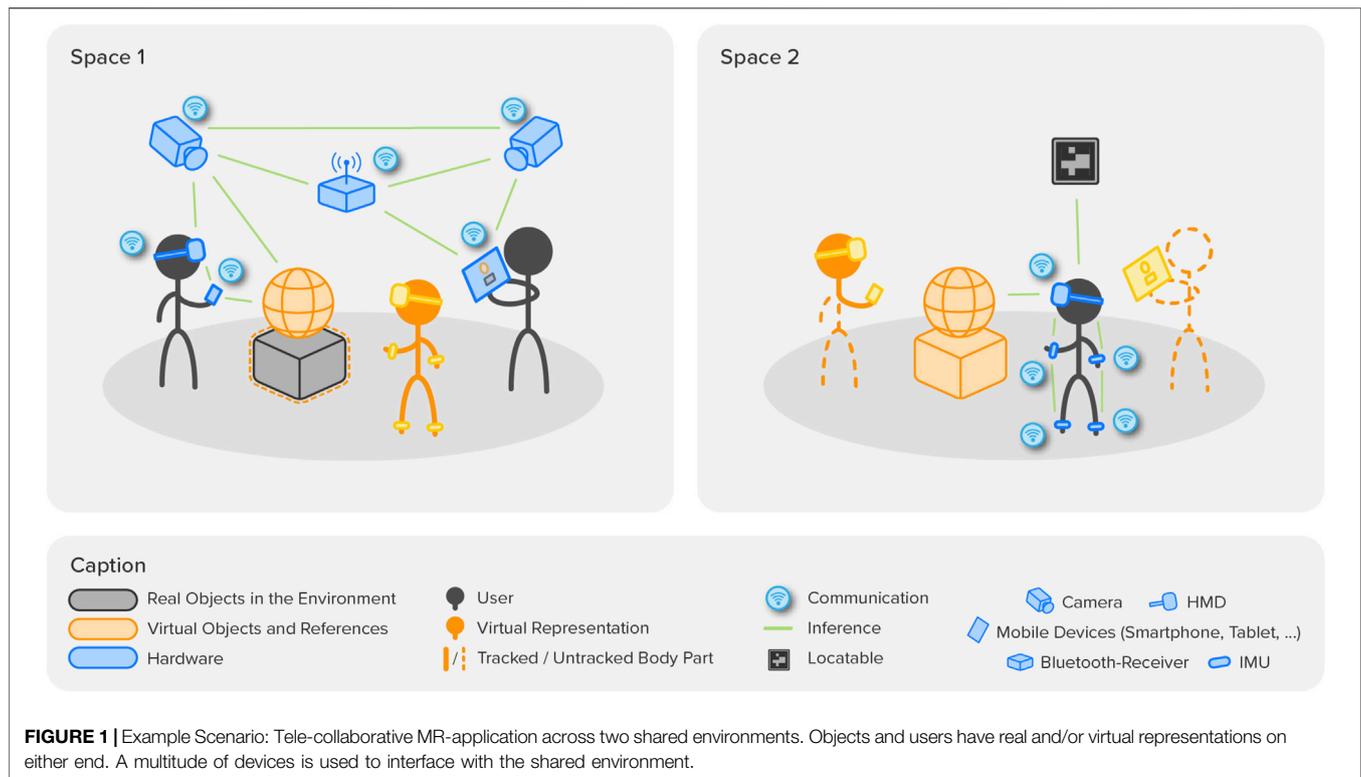
Weber S, Rudolph L, Liedtke S,
Eichhorn C, Dyrda D, Plecher DA and
Klinker G (2022) Frameworks Enabling
Ubiquitous Mixed Reality Applications
Across Dynamically Adaptable
Device Configurations.
Front. Virtual Real. 3:765959.
doi: 10.3389/frvir.2022.765959

1 INTRODUCTION

Contemporary futuristic media and imagination is very often inspired by a complete and permanent merging of real and virtual aspects of our lives. Slowly, it becomes technically conceivable to implement parts of this vision. Many technical aspects and requirements remain still unclear though.

Looking at the development of user interface technologies in the last years, we are faced with a rapid diversification and multiplication of hardware as well as software solutions undergoing continuous (r)evolution, each trying to become smarter and to standardize interfaces. Myers et al. (2000) characterized this situation at the start of the new millennium—and it has not changed since. It is particularly true for *Mixed Reality* (MR). It is safe to assume, that evolving standards will never completely align. Forward compatibility of system generations is not always considered or possible, either. Depending on how open or closed a system is, we have to rely on the interfaces provided to us if we want to use its capabilities.

Truly ubiquitous MR applications thus need to function in an ever-changing environment. They need to leverage the combined potential of these technologies with the “magic” happening in between their interactions, avoiding serious overhead in maintenance or being shoehorned into using only



one software stack/vendor guaranteed to be compatible. Indeed, users may want or be required to switch back and forth between hardware, depending on the time and place. Such applications will be context-aware, adapting to dynamically changing environments and become “smarter” themselves. The content needs to be accessible and interactive from different places, times and with differing hardware setups. For incremental evaluation of varying solutions, these need to be comparable over time side-by-side in the same context (Swan, 2018). This requires a more “black-boxed” approach to implemented solutions. Judging from the number of devices and areas of application, systems need to be more and more distributed while at the same time keeping hard timing boundaries for real-time interactivity. A framework that fulfills these needs must be able to incorporate new and upcoming developments and also integrate itself with other existing systems in order to truly embrace this new ubiquitous digital environment.

To advance in this direction, we present and discuss efforts tackling overarching ubiquitous problems in three domains: tracking, interaction and visualization. **Figure 1** illustrates one possible exemplary scenario where three users split into two separate locations view and interact with the same content using different hardware, each requiring some form of visualization of that content as well as the other two users.

Concerning ubiquitous *tracking* and *localization*, we revisit approaches including sensor fusion, data flow networks and spatial relationship patterns. We analyze these concepts, focusing on how to integrate them into a holistic ubiquitous MR environment.

Concerning *interaction*, we propose offering a network of nodes with common extendable message formats. These nodes can act as data producers/consumers, as well as hubs for data processing modules (edge computing) and binding endpoints for external systems. This allows decoupling of functionality and overall system behavior in these modules from the specific (number of) devices used during runtime.

For *content visualization* and visual representation of *3D user interfaces*, we discuss an automatic adoption mechanism for content displayed according to users’ visual and physical perception. In addition, we investigate how to consider environmental conditions for content placement. Furthermore, we address challenges of device limitations impacting the interface design to enhance user acceptance.

To evaluate and inform these efforts, we analyze a number of related applications. The applications cover various different domains like entertainment, serious games and collaborative work-spaces, requiring diverse and flexible use of hardware.

2 FUNDAMENTALS AND RELATED WORK

To define the vision and scope of the system we are talking about, we will first define the terminology. Then, we will investigate requirements and analyze related systems.

2.1 Users, Agents and Environments

Since current and future MR applications may well include somewhat intelligent (semi-) autonomous systems and virtual

assistants next to humans, we follow the definition of Russell and Norvig (2021) for agents. Human agents specifically are referred to as “users.” In artificial intelligence, agents are observed in relation to their environment, which in our case is understood as the rest of the MR application and its subsystems (some of which may actually be hidden to the agent). Important for considerations on the topics of inference, coordination and visibility is that agents usually bring with them or follow some form of intent or task. This intent may be brought into the system by a single agent and disappear with it, it might be formed and followed by multiple agents, or it might be inherent to the application/environment itself. All of these elements, possibly even the hardware and devices used by agents, form non-hierarchical relationships.

2.2 Components and Devices

In order to address the distributed and heterogeneous nature of MR apps and to be able to better reflect the mix between virtual and real elements, a more abstract interpretation of the nature of devices and their components has been adopted.

Components are regarded as the parts that provide or consume data, usually resulting in a change of system state. This includes analog as well as purely virtual input/output, e.g., the motor of a real robot vs. the same motor being simulated. Devices are used to hierarchically group components into meaningful entities. Agents trying to enact their intent can then use devices and their components to interact with the environment. Devices do not always relate to an agent, they can be “static” expressions of the environment as well. Devices are the conceptual groups expressing some form of contact with the rest of the MR app. To continue the robot example: structural elements, sensors, motors, manipulators, etc., are components that form the device “robot arm.” Whether an actual physical arm or the simulation of that arm, it is interfaced with the same way.

Devices and components can then be used to describe the capabilities they bring to the rest of the app. In turn, the app can be designed to (re-)use and replace devices with similar designation depending on the context. Works like Figueroa et al. (2002), Ohlenburg et al. (2007), Lacoche et al. (2015), Lacoche et al. (2016) and Krekhov et al. (2016) reflect the continuous need and effort to form abstraction layers making parts of the overall system more exchangeable and manageable.

2.3 The Vision of Ubiquitous Mixed Reality

Ubiquitous computing was defined by Weiser (1991). He described ubiquitous scenarios as work spaces, where tasks and users would be dynamically identified by so-called tabs to individually used hardware (pads) and shared multi-user hardware (boards). Weiser claimed that the individually used hardware is not bound to a user, but can be re-assigned anytime to a new user or a new task. This vision was adopted by Newman et al. (2007) for a definition of ubiquitous Mixed Reality applications, where tasks, users and hardware enabling applications of the Mixed Reality continuum [Milgram and Kishino (1994)] are coupled dynamically by an intelligent distributed system.

2.4 Framework Challenges and Requirements

Years of research on 3D UI, VR, AR and MR have identified and proposed solutions to quite a list of challenges. The following points are taken as a list of general requirements that the proposed frameworks have to respect and deal with in order to be viable. They can be roughly thought of in terms of functionality (1–5) and in terms of using the framework for development (6–9).

2.4.1 Plasticity, Adaptivity

Ongoing research efforts have been conducted to make MR applications dynamically adapt to changing circumstances. Browne (2016), Thevenin and Coutaz (1999), Lacoche et al. (2015) and Ohlenburg et al. (2007) concentrated on introducing systematic abstraction layers for devices and how they are interacting to make user interfaces more flexible.

In ubiquitous MR, an application should know enough about its runtime environment to adapt its own configuration (*Adaptivity*), but it must also be able to deal with changes in the hardware form-factors or the environment (*Plasticity*). The environment itself can be assumed as everything but static with users joining/leaving, users relying on individual hardware or hardware being exchanged, the composition of the physical surroundings changing, etc. All this requires the application to adjust to and incorporate the capabilities provided and available at the current point in time. A framework must then offer systematic approaches to developers dealing with these dynamic conditions.

2.4.2 Connectivity

As MR applications often rely on spatial relations and potentially run for extended periods of time while agents join and leave, methods of providing data to participants have special requirements apart from basic real-time interactivity. The publish-subscribe pattern has been adopted by Pereira et al. (2021), Waldow and Fuhrmann (2019), Blanco-Novoa et al. (2020) and Fleck et al. (2022) as an efficient message delivery system and is commonly used for IoT systems (MQTT). Furthermore, in MR it is important to consider spatial and temporal neighborhoods when connecting systems to minimize latency—Pereira et al. (2021) use a geospatial atlas of “realms” to connect to instances and provide content close by. These concepts may need to extend to all scales—from body-worn to room-scale, urban-scale and world-scale.

A certain amount of latency or irregularity in the data transmission is unavoidable, though. Huber et al. (2009) and Huber et al. (2014) investigated methods of synchronization and interpolation between data points.

2.4.3 Scalability

Bondi (2000) distinguishes between load scalability, space scalability, space-time scalability, and structural scalability of computer systems in general, relating the four concepts to varying numbers of objects in a system. In MR applications, such objects range across multiple dimensions: the number of

virtual and real objects of interest (content), the number/kind of interactive devices, as well as the number of parallel users that collaborate or compete. They affect the system and network load, as well the processing time.

2.4.4 Data Security, Privacy

De Guzman et al. (2019) have put together a comprehensive survey on the security and privacy challenges on several levels of MR architectures. Prominent concepts are very fine-grained data access controls as well as the introduction of intermediary protection and abstraction layers only providing data that is necessary to later stages of the processing pipeline—as opposed to feeding raw data into a monolithic system with unlimited and uncontrolled data access. Distributed architectures using, e.g., blockchain technology as demonstrated by Sosin and Itoh (2019) may be another solution to put control over data into the hands of the users.

As concerns may arise and change during use of an MR applications based on, e.g., the spatial constellation of users, users entering restricted areas or use of the application during certain hours of the day security and privacy evaluations definitely need to go beyond common user and device authorization—MR requires much more context awareness to recognize potential risks.

From a framework perspective, it seems quite difficult to consider all proposed and future security and privacy measures on all levels in a single framework. A split into several frameworks that are each focused on one aspect—for example, 3D rendering output and the accompanying risks—might be more suitable.

2.4.5 N-Dimensional Content Reasoning

Today it seems some of the more visionary concepts for MR are often hindered by the lack of contextual information necessary for advanced inferences based on spatial, temporal and/or semantic relations. If (networks of) applications are to become truly smart and user interfaces to be naturally integrated with the surroundings, more understanding and reasoning about spatial, temporal, semantic, etc., relations is required. This information may help solve but also bring additional concerns for *Data Security and Privacy*. For *Plasticity and Adaptivity* too, on higher levels of abstraction it may only happen with enough meta-information. For semantic understanding of 3D scenes and reconstructions Bowman et al. (2017), Dai et al. (2017), Fehr et al. (2017) and Tahara et al. (2020) have recently presented solutions. This type of annotations could be extended to agents and their devices to better understand intentions and how to interface optimally.

2.4.6 Interoperability, Extensibility

One time-consuming frustration as a developer for MR is a heterogeneity and a lack of established standards for hardware and software. The hardware brought into the application might follow different standards or provide different levels of performance, might evolve over time and become old or even outdated. In some cases, it might simply be unresponsive and/or broken. Throughout the lifecycle of an app, pieces of hardware

and/or software may become updated or replaced entirely because better or entirely new solutions arise. The more a framework can help a developer ease through these fluctuations without having to spend additional time and effort, the better it will serve its purpose.

2.4.7 Convenience

It is desirable for frameworks to keep entry hurdles low and hide complexity, making it accessible only if needed. Depending on the technical expertise of the developer—e.g., designers vs. software engineers—mechanisms to provide ready-to-use and auto-configuring building blocks that can be shared and improved on their own helps. Depending on the type of MR application, users might even become content creators themselves.

Just as important, frameworks should offer ways of introspection and tools for error analysis. Especially for distributed applications, the task of finding an error is arduous if not properly communicated or analyzed.

2.4.8 Quality Assurance

In order to produce viable and lasting results, reproduction, verification, comparison and testing are a must. Especially on the topic of reproduction and verification, Swan (2018) commented on the replication crisis and what it means for the Virtual Environments community, arguing for more acceptance of studies replicating and solidifying results. Casarin et al. (2018) too explain how validation is slowed down by an arduous process of re-developing with rapid changes in hardware. One way for frameworks to alleviate this problem is to design them allowing easy adoption and integration of systems, architectures or algorithms presented in other studies (in a modularized version). This helps test them under similar conditions without relying on the exact same software stack. For example, the design and implementation of interaction techniques as in Figueroa et al. (2002) and Casarin et al. (2018) could be investigated in such a fashion. To cut time-consuming re-implementations, frameworks could help integrating solutions using their original runtime environment. Conversely, a framework should motivate developers to implement solution in a decoupled and reusable form for quick and direct rewiring to other input/output.

2.4.9 Integrability

As we will try to demonstrate in chapter 4, not all of these requirements hold for every MR application, and it will be quite difficult to cover them all on every level in a single framework. This is why we propose to develop several frameworks that respect and deal with these requirements but are otherwise specialized to solve the challenges in one domain of expertise. This in turn will force the frameworks developed to think of themselves as one piece to the puzzle, reinforcing the notion of *Integrability*—i.e., besides the question of “How well can my framework integrate other systems?” it is just as important to think about the question “How well can my framework be integrated with other systems?”.

2.5 Recent Related Systems

In the last decade, a number of powerful solutions for MR experiences have started emerging. By now, there are a variety of vendors offering MR hardware that is rapidly evolving, as well as software to support development for this new hardware on a large range of platforms. And while standards are being established that abstract from lower hardware levels, the software stacks used to build MR applications each have their own strengths and weaknesses targeting their core domain, again each bringing a list of SDKs, APIs, Plugins and general software environments.

There are many current efforts to define device independent shared standards for Mixed Reality. On one side, we have **proprietary systems** like Apple's ARKit and Google's ARcore with ever-evolving functionalities for handheld devices—partially also for more than one operating system, but just for handhelds. With PTC's Vuforia engine and Unity's ARFoundation, there are also augmented reality frameworks, that enable bringing the same application logic to AR head-mounted displays and handheld displays across a variety of operating systems and hardware architectures. However, both systems are rather closed—it is hard to add individual features like self-made marker implementations to them, so we are bound by the (admittedly impressive) list of functionalities these frameworks offer and the devices these frameworks support.

2.6 Open Standards

More open standards currently appear as well. Here, large consortia formulate standards like the W3C WebXR device API, bringing Mixed Reality to browsers and the OpenXR¹ standard by the Khronos Group deploying the same application to many head-mounted Mixed Reality devices. However, the strength of these two frameworks is, either on Web Level or on Hardware level to build an abstraction layer upon used hardware—they do not help to connect many devices to the same application. Furthermore, since at least WebXR is in an early development state, a low entry-hurdle is not given yet, and we observe version breaks and missing compatibility with the change from WebVR to WebXR—the same is sometimes true for closed systems such as Google's Tango Platform that was replaced by ARCore after a few years, making the situation for developers unreliable and limiting the possibility to use these frameworks for MR research, where we need to be able to compare solutions developed over time.

A few platforms are emerging which combine Mixed Reality to other fields with three-dimensional requirements. Examples are the Open Spatial Computing Platform by OpenARCloud, a platform trying to mount AR experiences to the real world by the usage of Geoposes and the GL transmission format (glTF) by the Khronos Group as a tool for efficient 3D model exchange.

There are also research-based efforts which mostly produce new concepts but (in the domain of MR) rarely maintained and reusable code: There is the approach of adding a blockchain as a tool to reach serverless authentication by Sosin and Itoh (2019),

adding semantic scene understanding as a tool for ubiquitous interaction with the environment (Chen et al., 2018) or using game engines as a tool for hardware-independent MR applications (Piumsomboon et al., 2017).

Ideally, one would want to wrap these great ideas, findings and solutions into a module and then verify, compare and re-use them in different contexts. Systems like ROS—the Robot Operating System, Quigley et al. (2009)—did become a kind of standard because they were not designed for one specific context or environment and instead kept possibilities open for the developer to quickly iterate new ideas and integrate them with existing solutions.

3 PROPOSED FRAMEWORKS

There are two main approaches for MR frameworks. The first one tries to solve all issues of MR at the same time—e.g., Arena (Pereira et al., 2021), DWARF² (Bauer et al., 2001; Sandor and Klinker, 2005). An advantage of this is to have a complete, monolithic system. This is helpful to have a quick up-and-running system with a bunch of functionality. Moreover, if they are using open-source approaches and plug-in capabilities, these systems are even evolvable and more flexible than classical monolithic systems. The drawback is the complexity and the maintenance-heavy structure. The second approach is to focus heavily on a specific use case and device configuration. Such focused systems often offer better performance and better use of the possibilities of specific devices. We propose using three independent frameworks for interaction, visualization, and tracking as services for a broad field of Mixed Reality applications. This splitting offers the chance to handle optimized solutions like sensor fusion and error propagation and at the same time, hide this complexity concerning the interaction and the visualization layer. Furthermore, this splitting forces us to develop clear interfaces between these three functionalities that are (hopefully) also usable if one of the frameworks becomes unmaintainable or obsolete. We distinguish tracking, interaction and visualization since this taxonomy is close to the input, processing and output architecture of classical IT systems. Therefore, in our opinion, this split is very intuitive. This should not however imply that these three are understood to cover every single aspect of MR. They may very well be extended by additional frameworks in the future. Another reason for this split is that each of these frameworks builds upon the knowledge of other computer-science disciplines. UbiTrack uses sensing and computer vision approaches, Ubi-Interact uses knowledge about network architectures and efficient interprocess communication and UbiVis builds upon computer graphics and software ergonomic. By encapsulating these functionalities, experts in the respective fields can optimize parts they can handle. For the other parts, they can use standard configurations or less specialized alternatives. For example, if no spatial adaptation of

¹<https://www.khronos.org/openxr/>

²Distributed Wearable Augmented Reality Framework

UI elements is necessary, a tracking specialist could plug in a Unity rendering pipeline instead of UbiVis for the visualization.

3.1 UbiTrack

UbiTrack was built with the intent to combine several heterogeneous tracking and localization approaches dynamically and automatically. The system was developed and enhanced over more than a decade. Many physical principles can be used to determine the 6-DoF pose of objects and devices in an environment. Each modality comes with advantages and disadvantages (Welch and Foxlin, 2002).

MR systems need to know the spatial relationships between all mobile and stationary devices involved in an application: all stationary devices, as well as all rigidly combined components of mobile devices (Pustka and Klinker, 2008; Waechter et al., 2010; Benzina et al., 2012; Itoh and Klinker, 2014), must be registered while all mobile devices must be tracked. There may also be self-referencing, source-less sensors such as IMUs and inside-in optical tracking algorithms like SLAM (Davison, 2003), PTAM (Klein and Murray, 2007) and KinectFusion (Newcombe et al., 2011). When used in context with globally referenced information, these self-referencing systems also need to be registered (initialized) to the stationary environment. The result is a network of n:n spatial relationships, some fixed and some changing over time. Most MR and robotics systems model such spatial relationships internally. The tracking configurations are hard-wired in the system implementation and are thus not open to flexible rearrangements by users or application configurators when new devices are to be integrated.

UbiTrack externalizes spatial relationships on a declarative level into **Spatial Relationship Graphs (SRGs)** (Pustka et al., 2011). SRGs can be formally specified *via* the Ubiquitous Tracking Query Language (UTQL) (Pustka et al., 2007). Users can generate and edit them textually or graphically with an interactive tracking manager (Keitler et al., 2010b). SRGs contain device components as nodes and device transformations as edges. The spatial relationship between two objects or device components in an SRG is determined by a path from a source node representing the first object to a sink node representing the second object. Paths contain static (registered) as well as dynamic (tracked) edges, based on intermediate nodes that represent sensing devices. Edges along the path define a concatenation of spatial transformations. Any spatial relationship that is not directly measured can be derived by the concatenation of known (or previously derived) spatial relationships. Note that it is common practice in the robotics and computer vision communities to illustrate the internal sensing behavior of their systems with such graphs. Yet, there is no process to link these illustrations to the systems.

Using graph searching and subgraph matching techniques, UbiTrack can be used to turn the declarative spatial knowledge of SRGs into executable tracking systems by generating **Data Flow Networks (DFNs)** (Pustka et al., 2011). Users can interactively determine paths in an SRG from one or more source nodes (representing target objects relevant to an application) to a sink node (the application) (Keitler et al., 2010b). Parts of the path may consist of subgraphs rather than simple edges. Such

subgraphs describe complex robotics algorithms for tracking, registration (Horn, 1987; Daniilidis, 1999) and sensor fusion (Durrant-Whyte and Henderson, 2016) which require special spatial arrangements or several objects to derive a spatial relationship between a source and sink node of the subgraph. UbiTrack provides a large library of such SRG subgraphs, so-called **SRG Patterns** (Pustka et al., 2006). When a user indicates a path through an SRG, UbiTrack transforms the respective edges and sub-graphs into executable processes (implementations of algorithms to concatenate or invert transformations, as well as for sensor fusion and registration) and chains them together in data flow manner. MR apps (SRG sink nodes) thus receive the result of a chain of transformations, externally defined in an SRG and produced by Ubitrack. They do not need to specify such transformations within their own code, keeping it agnostic to spatial rearrangements.

Beyond graph searching and subgraph matching, UbiTrack also addresses issues inherent to robust, precise and efficient tracking. By including them in UbiTrack rather than into applications, they are readily integrated at the application level without requiring this kind of expertise from those concerned with the application content. UbiTrack explicitly handles **measurement errors** and their propagation along paths of the SRG (Bauer et al., 2006; Sielhorst et al., 2007; Pustka et al., 2010). If there are multiple paths involving different sensors, the amount of accumulated errors can vary significantly (Keitler et al., 2008; 2010a). With the AR4AR facility, users can see augmentations within their physical environment that show how different paths from a sink node (target object) through an SRG to the sink node representing an MR app yield different pose estimates (Pankratz and Klinker, 2015). Furthermore, UbiTrack handles measurement differences in **asynchronous sensor setups**. The involved sensors do not start measuring at the same instant and also depends on the processing stack that digitizes the raw data. UbiTrack aligns the timelines of all sensors *via* time delay estimation (Huber et al., 2009). It performs the required signal correlation process in 3D along the dominant motion direction, alternating between spatial and temporal calibration several times. It also temporally aligns measurements of asynchronously operating sensors. If one sensor pushes a new signal to the UbiTrack server, it pulls measurements from the other sensors for the same instant—interpolating between real measurements or extrapolating from recent measurements into the future (Pustka, 2006). Ubitrack provides such services, e.g., based on motion models and Kalman filtering. Finally, UbiTrack offers many options for **meta-level spatial reasoning** based on SRGs. Inspector apps can continuously watch the data flow generated from an SRG and use the declarative knowledge about the spatial relationships to detect a misregistered or dysfunctional sensor when multiple paths through an SRG yield strongly differing results. Akin to roaming in telecommunication, such apps can also dynamically activate or deactivate parts of a DFN when a user leaves the sensing range of some sensors and enters the range of others (Pustka and Klinker, 2008). In

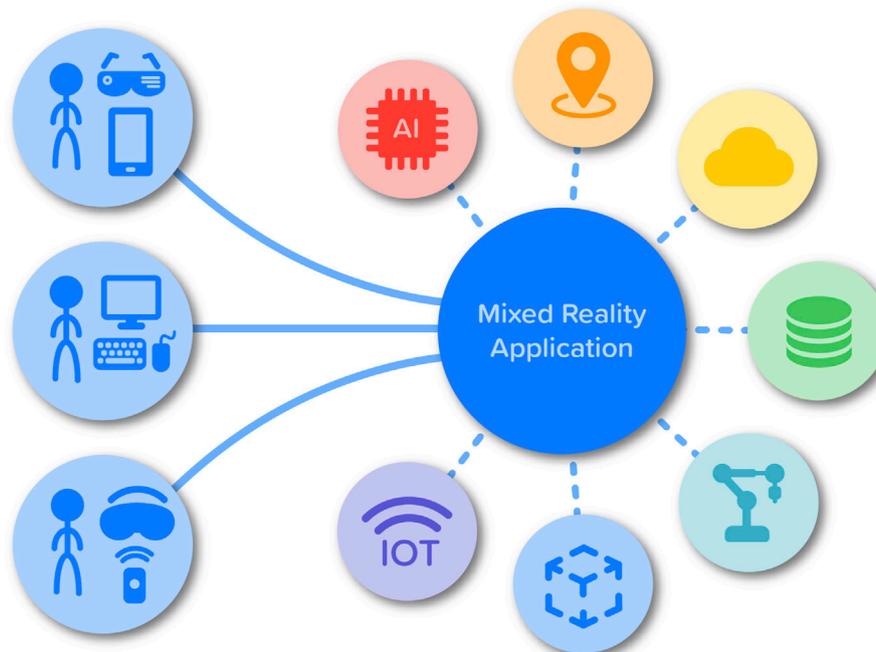


FIGURE 2 | Overview of systems that might play a role in MR applications and their combined potential (AI, location, cloud, databases, robotics, virtual models, IoT). Each circle typically represents a distinct system with its own API. Circles to the left (AR glasses + smart device, desktop PC, VR equipment) are examples of technologies used by humans to integrate with the rest of the virtual world (their “digital skin”).

combination with the time-enriched sensor data of the DFNs, the SRGs also provide opportunities to reason about events. When events are measured simultaneously by multiple sensors for which the SRG did not yet indicate a spatial relationship (disconnected subgraphs), Ubitrack can determine such new relationships and add them to the SRG (Waechter et al., 2009, Waechter et al., 2013).

In consequence, UbiTrack allows for the installation of **ubiquitous tracking services in AR-ready environments**. When a mobile MR app enters such an AR-ready environment, it registers to a server, announcing all spatial relationships between its own sensors and targets that are installed on its mobile device(s). The server merges such app-related SRGs with its own ubiquitous SRG, which represents all stationary sensors and targets in the environment. Depending on the current pose of the mobile MR app, it delivers a DFN to the app that represents a suitable current match between mobile and stationary sensors and targets (Huber et al., 2007). Note that the DFN can contain shared tracking services between multiple mobile MR apps (given by several mobile SRGs).

During the configuration phase, in the current UbiTrack implementation³ an expert user has to conceptualize the DFN out of the algorithms and specific sensor streams manually. This makes UbiTrack hardly manageable for novices. In the future, UbiTrack could integrate automatic inference or supporting semi-automatrical suggestion systems to increase *Convenience*.

³<https://github.com/Ubitrack>

3.2 Ubi-Interact

Ubi-Interact (Weber et al., 2021) is an effort to connect and orchestrate arbitrary systems and devices into one distributed application. It is therefore mainly concerned with network communication, message formats and being able to integrate with external existing ecosystems. It can be used to define a common language and implement system behavior for a complete MR application, but could just as well be used to bundle a set of devices that integrates into a bigger context.

Originally, Ubi-Interact was envisioned as a tool for highly personalized setups directly worn or held by users. To the left in **Figure 2** are examples (AR glasses + smartphone, desktop PC, VR equipment) that could be described as the “digital skin” or “digital suit,” i.e., the combination of devices used to get in touch with the digital world and which embody a human in their new digital environment. The expectation is that further technology (wearables, etc.) and personal intelligent virtual assistants evolve, the better these systems will “fit” a user and hopefully expand their digital agency. Its systematic however can be equally applied to higher scales of deployment, with any circle of **Figure 2** potentially being represented by Ubi-Interact nodes. While one Ubi-Interact arrangement is centralized around a master node, all Ubi-Interact nodes are expected to form networks between each other and/or external systems. This follows the ideas outlined in chapter **Section 2.4** for *Scalability* and *Integrability*.

Like related efforts (Pereira et al., 2021; Blanco-Novoa et al., 2020; Fleck et al., 2022), Ubi-Interact makes use of a publish-subscribe message broker to cover requirements of *Connectivity* and *Plasticity*. In contrast, Ubi-Interact does not build upon the

MQTT protocol. Usually, to build upon an established and standardized solution is a good decision as it provides trust in a proven and tested set of features and is usually maintained and already supported by many adjacent technologies. For ubiquitous MR however and starting Ubi-Interact's development, the to-date topic-based message brokers seemed limited in their expressiveness when it comes to the desired *N-Dimensional Content Reasoning* for discovering, examining and navigating more dynamic environments. Wildcards or regular expressions applied to topic strings are a known concept and have proven useful. We think the range of approaches needs to be extended to encompass much richer information impractical or performance-affecting to be encoded in the topic string and/or the associated data format alone. Examples for meta-info include performance and reliability metrics like latency or data accuracy ranges and confidence, physical characteristics like real-life screen and pixel size or an agent's context and perceived intention to name some. Some of the context information that is dynamic may be topic-based itself (e.g., positions), some may be produced as a side effect (e.g., latency) and/or are not supposed to be public. Other cases seem impractical to be encoded in the topic string itself (bandwidth use) and introducing it through additional topics (topics talking about topics) imposes a rather artificial set of topic naming conventions that are typically not enforced by the broker and must be documented and applied separately—going against well-proven QA methods like Poka-Yoke (Shimbun, 1989) by opening the door for accidental mistakes by developers or breaking the system through updates. Consequently, Ubi-Interact is also an investigation into how tightly its broker needs to be integrated with the meta-information about clients, devices and components to handle dynamic subscriptions in the system—possibly advancing *Convenience* and *Data Security and Privacy* beyond basic authentication and authorization mechanisms too by, e.g., declining published data or not providing subscription updates based on a client's position data. Ubi-Interact then bridges to other message delivering technologies to build a “system of systems,” e.g., establish communication between MQTT, ROS and other systems without messaging. Ubi-Interact does not presume that all communication be exclusively run through it (e.g., peer-to-peer streaming) nor that all parts use the same protocols (although it can provide one), instead focusing on forming a conjoint system.

If in the future it turns out tight integration of the broker is unnecessary or the benefits of established systems are too big, then the current pub-sub connections could be replaced with, e.g., an MQTT client with relative little effort.

Ubi-Interact's concept of **Topic Multiplexers** leans into the dynamic and adaptable subscription model. Muxers as topic aggregators/disseminators are envisioned to be meta-devices extendable to include any meta-information and characteristics about clients and their components in addition to filters based solely on their topics. To pick up on the example for ubiquity by Weiser (1991): one could envision a MR room where people use AR HMDs in combination with handheld devices (e.g., smartphones) for combined interactions. Devices may belong to the room or be brought in by users, both potentially being

shared between users. Furthermore, some devices in use may not exactly fit the profile of other handhelds but cover some aspects of its functionality (e.g., worn devices with IMUs). A Topic Muxer is designed to catch these (partial) digital profiles and channel them together for combined processing.

To cover aspects from *Adaptivity, Connectivity, Interoperability & Extensibility* and *Quality Assurance*, Ubi-Interact provides **Processing Modules** as black-box functionality to the rest of the application with dynamically mapped I/O and a lifecycle interface. Similar to arguments from Figueroa et al. (2002) and Casarin et al. (2018) for interaction design and flexibility as well as event processing concepts seen in Node-RED (OpenJS Foundation and Contributors, 2013) and ROS (Quigley et al., 2009), Processing Modules allow free implementation of functionality. Using Topic Multiplexers it can combine variable numbers of components, but can also be instantiated by and for single clients to map their specific devices, needs and configurations. As such, the Processing Module interface does not assume any particular language or environment like collaborative interaction design—any node with proper execution of lifecycle callbacks and I/O mapping is thereafter free to execute any code inside the callbacks and rely on any additional dependencies. One use-case could be to encapsulate an image processing algorithm and make it usable between clients and applications. If written by another author, we hopefully follow improvements and updates with minimal effort, as long as outwards behavior is not drastically changed. In the same vein, swapping and comparing two comparable solutions is possible as long as they follow similar I/O signatures. For testing purposes, input can easily be emulated or played back by publishing pseudo or recorded data. Processing Modules also constitute communication and update endpoints for external applications hidden behind it—this supports *Integrability*.

On *Data Security and Privacy*, Ubi-Interact tries to decentralize and keep control in the hand of the individuals where possible—again relating to the idea of a “digital skin” embodying you in the virtual world. An open source library of Processing Modules that is shared, instantiated and adjusted to individual needs on self-owned setups would reduce the need to provide raw data to third parties for processing and interaction. Instead, third-party applications can define minimal data interfaces and Ubi-Interact can instantiate a module that checks against and communicates with said interface.

For *Convenience*, Ubi-Interact comes with a web-frontend providing administration and introspection tools plus simple code examples and live integration/performance tests.

3.3 UbiVis

Perception of digital content is an important role in all Mixed Reality scenarios, independent of the degree of perceived reality. Our UbiVis framework tackles challenges emerging from using different hardware used for the current Mixed Reality experience. This includes on one hand hardware related challenges and on the other solutions for displaying virtual content in collaborative MR-applications. But also, concepts to solve problems for different suitable HCI solutions that dynamically changes depending on real environment and available user-interfaces.

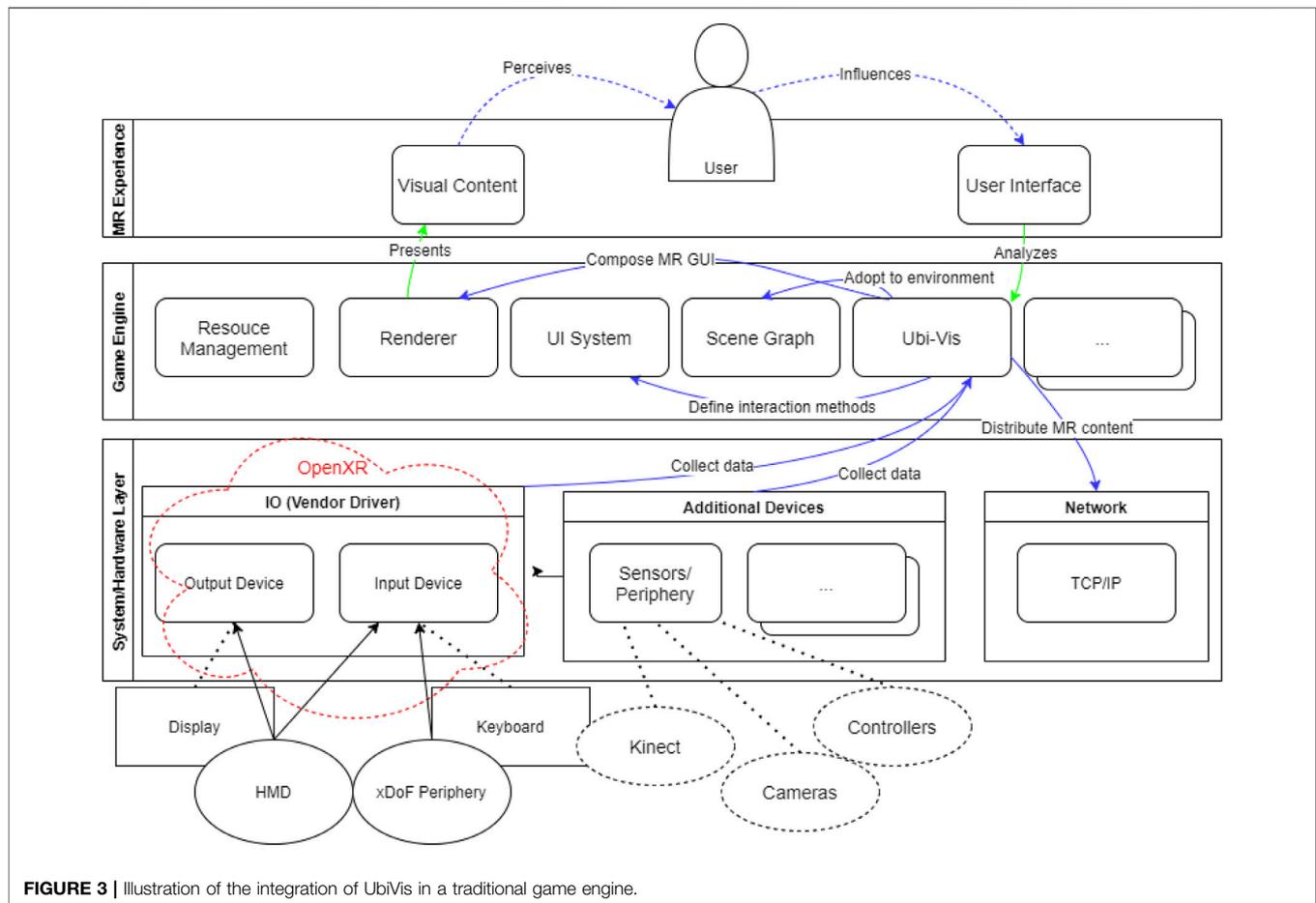


FIGURE 3 | Illustration of the integration of UbiVis in a traditional game engine.

During our work with applications for AR or VR devices, we discovered lacks of usability in creating applications for different hardware environments. In general, there are plugins and tools to handle different platforms within one IDE. There are plugins for Unity3D, Unreal Engine and others to include Oculus VR goggles to your app. But in case you also want to deploy it on a HoloLens, it becomes more difficult to handle all kinds of minor conflicts between these two devices. From the developer's point of view, it is difficult to keep track of all possibilities different devices have that have to be implemented in a content creation tool, e.g., Unity3D. The consumer market becomes increasingly diverse with different vendors for AR and VR HMDs, handheld video-see-through with tablets or smartphone, or smartphone based VR/AR solutions. To reduce the workload for handling each device, Khronos Group started with OpenXR, an initiative to create a common interface for multiple vendors and devices. This might help to decrease the maintenance and engineering effort for content creation tools, but it does not solve issues with different hardware characteristics regarding display capabilities or input options.

As Figure 3 shows, where OpenXR improves communication with hardware from multiple vendors. A lack of missing adaptability of content is recognized in traditional game engine content pipelines as fixed applications for each

hardware setup. This leads to a more challenging task for content creators, as they can not predict on what device an application is running and which user interfaces are available. To reduce this workload, the presented framework handles adjustments to different content of an MR-app during runtime. Concerning virtual UI elements that might be designed as diegetic or non-diegetic content intractable for a user, they have to become context aware and automatically adjust depending on the user's setup and environment. The same goes for virtual objects that are placed in MR experiences having to be relocated if necessary. This enhances usability and user experience of customers by dynamic adjustments of presented content, depending on their current environment and including device characteristics. This fits to the requirement of *Adaptivity* and *Plasticity* as described by Browne (2016) and Lacoche et al. (2015). As a result, content developers do not need to concern themselves whether a user wears an optical see-through, a video see-through or even full VR HMD nor what FoV a system has. Depending on the user's settings, the framework has to evaluate optimal positions for virtual content that is perceived in single or shared environments. Another focus is to reduce mental workload on users that are using the same application with different technologies. This means content like user interfaces should be as identical as possible, but with optimal knowledge

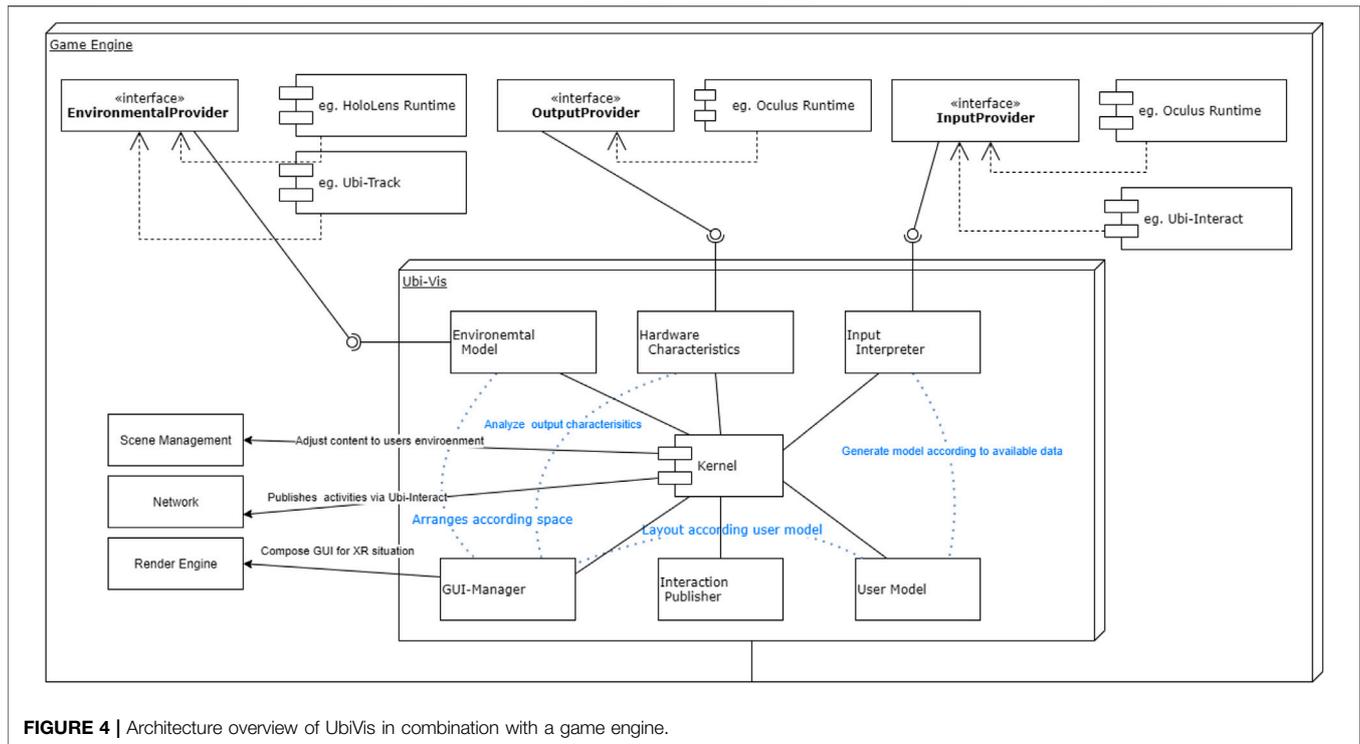


FIGURE 4 | Architecture overview of UbiVis in combination with a game engine.

transfer and interaction possibilities—regardless of whether a user has full hand tracking or just a tracked game-pad.

To solve basic interaction problems in an immersive 3D environment using a HMD, vendors already try to establish frameworks for creating 3D UI. As an example, Microsoft's MRTK⁴ allows a developer to build up 3D UIs for the MS Holographic platform in a very smooth and fast way. As long as only one ecosystem is used, the results are acceptable and in general all you need. To overcome the limitation of vendor specific application, we introduce the UbiVis framework. For a real ubiquitous Mixed Reality experience, it is essential to overcome vendor specific solution towards a dynamic and use oriented way of designing applications.

For integration purpose and easy adaptability to changing software environments, UbiVis is designed as a module or plugin for game engines. **Figure 3** gives an overview about the access to modern MR-technology, like HMDs with hand tracking in contrast to traditional periphery. Also, the information handling and involved system is illustrated to allow easy integration in traditional game engines. This allows content creation by experts with their well-known toolchain with only a minimum additional work to provide metadata that describes their intended functionality for the framework. During runtime, the **UbiVis kernel** receives information about different parts of the actual system. One part of the main kernel are modules listening and analyzing interfaces to the real world and to the user. In **Figure 4** interfaces for environmental providers, like depth maps of the current surrounding or reconstructed

geometries are streaming into the kernel. Depending on the system, information may come directly from vendor specific solutions, like the HoloLens2 API or *via* other frameworks or even different devices in the room. This is achieved by connecting to a Ubi-Interact master node or using UbiTrack data providers, depending on the application environment and specifications. It can benefit from working on data coming from the current device itself (fast closed-loop performance) but also use network services from other headless participants.

The decoupling between low latency information on client side and maybe more delayed information from other participants is important to avoid mismatches in visual cues of users. This is also stated out by Pereira et al. (2021) who tries to overcome the monolithic app limitations by a WebXR based all-in-one platform solution. In contrast to ARENA concept, we designed to be able to run in single device applications by including UbiVis in a compatible game engine as middleware or connecting it with other devices or participants *via* Ubi-Interact. This concept enhances a device's capabilities by combining multiple devices for a single user experience or for shared environments using Ubi-Interact.

By knowing the user's environment and user interface possibilities, we can now think to introduce a GUI system for MR environments. Considering window management systems and concepts for WIMP based desktop application are well known to users. For smartphones, new touch control concepts were introduced in past years and accepted by users. For MR applications we need to design interfaces, integrating as smoothly as possible with known structures and concepts but with full respect to new possibilities.

⁴<https://github.com/microsoft/MixedRealityToolkit-Unity>

This is solved by introducing a GUI manager connected to the UbiVis kernel, seen in **Figure 4**. Connected to the virtual application, a developer is able to define functionality and structure of a desired virtual user interface. During runtime UbiVis analyses the current environment and possibilities per user and decides upon a given rule set which UI elements should be directly accessible, grouped or hidden. Depending on the current spatial relationship between intractable content and a user it defines different interaction possibilities, considers direct manipulation of content in reach of a user or indirect techniques for other objects. Depending on the user's input device, different methods are more appropriate than others as described by LaViola et al. (2017), narrowing the selection of possible solutions. That allows the framework to locate content in a suitable way for highly diverse user scenarios or unpredictable environments.

Content designers do not have to think about every scenario that might occur on a user's setup, but just to identify requirements how virtual content should behave or how it can be manipulated. During runtime, the framework decides on actual methods and visual representation on a chosen location in the user's MR experience. UbiVis relocates virtual content to allow desired interaction concepts and preserve knowledge transfer. Here, information and content needs to be categorized and ordered by importance to identify the best location in the user's primary interaction space. Depending on the current interaction range, some content may be relocated to a position with lower priority—similar approaches are already presented for smartphones notifications by Quigley et al. (2014) or Pielot et al. (2014). Depending on the current optical system, virtual content may be replaced by a suitable representation for a stereoscopic view or a video see through system. For this purpose, UbiVis has to identify users' actual hardware and information about secondary input or output devices in their environment to create a full MR-experience. Therefore, it can be linked to the already presented Ubi-Interact framework in **Section 3.2** to identify other input possibilities or even other users in the current environment.

We have started a prototypical implementation of this framework as a light-weight game engine, including a full rendering and game logic to allow UbiVis low level access to all vendor specific information. The OpenXR framework is evaluated to reduce the programming workload and core capabilities of UbiVis. To allow a broader usage, UbiVis core needs to be extracted in a standalone plugin for commonly used game engines to create a seamless Mixed Reality experience by adopting virtual content to the real environment and the user's capabilities for a broader audience as shown in **Figure 3**. The long term goal is to create a framework that helps people to enjoy and create MR-applications with only a small amount of overhead for MR-related UI concepts and hardware constraints to create a full multi-user MR experience as presented in **Figure 1**.

To conclude, with UbiVis we try to close the gap for a highly adaptive user-interface with respect to a user's input space and its real world environment, independently of whether an AR, VR, or Desktop system is used. The goal is to identify common interaction patterns based on clustering to select suitable direct

or indirect manipulation techniques and combine them with the virtual content around a user. As a long term goal, we have to think about a more flexible content creation pipeline with respect to the different scenarios where it might be used. Here we have to evaluate different methods how a content creator can describe the functionality of virtual objects and their interaction possibilities such that our system can decide during runtime which interaction concept is the best for the current situation. This will bring a more authentic, easier, and better understandable user experience to developers, independent of the used hardware combination. This is especially important if MR applications address a broader audience and use more diverse hardware, rather than staying within a vendor product.

4 USE CASES AND EXAMPLES OF MIXED REALITY APPLICATIONS

The proposed frameworks were not conceived in a vacuum. Most of the requirements and concepts implemented derive from the requirements of actual application projects and reoccurring necessary solutions. They cover the range of serious games, sports and entertainment, simulations, industry and emergency services.

4.1 Industry and Commerce

When applications spread across wide areas, e.g., item picking in logistic centers (Schwerdtfeger and Klinker, 2008), or inspection and maintenance of machines (Klinker et al., 2004) in large plants, it is essential to provide flexible tracking facilities to keep costs under control. Some areas may require high precision tracking, whereas other places get by with lower requirements. It may also be an option to wheel in high-quality equipment on an as-needed basis.

UbiTrack is capable of providing very flexible, quickly reconfigurable tracking solutions for mobile MR apps. It has been developed and used in a number of industry-oriented research projects, such as AVILUS (Alt et al., 2012), ARVIDA (Behr et al., 2017), ASyntra (Pustka et al., 2012), TrackFrame (Keitler et al., 2008; Keitler et al., 2010b) and Presencia (Pustka et al., 2011; Normand et al., 2012).

It has also been essential for the scientific investigation of multi-sensor-based tracking scenarios such as the fusion of mechanical and optical tracking (Eck et al., 2015), and the use of eye tracking for HMD calibration (Itoh and Klinker, 2014) for the EU EDUSAFE (Mantzios et al., 2014) and VOSTARS (Cutolo et al., 2017) projects.

4.1.1 VR Supermarket

To acquire a platform for standardized testing of mobile Health (mHealth) applications, a VR Supermarket Eichhorn et al. (2021) simulation involving the usage of the participant's own smartphone (heterogeneous and specific devices as part of *Interoperability* and *Extensibility*) has been developed. The integration of the known and hence comfortable smartphone in combination with involvement of real apps, hence not derived and reduced simulated versions, helps to eliminate barriers and

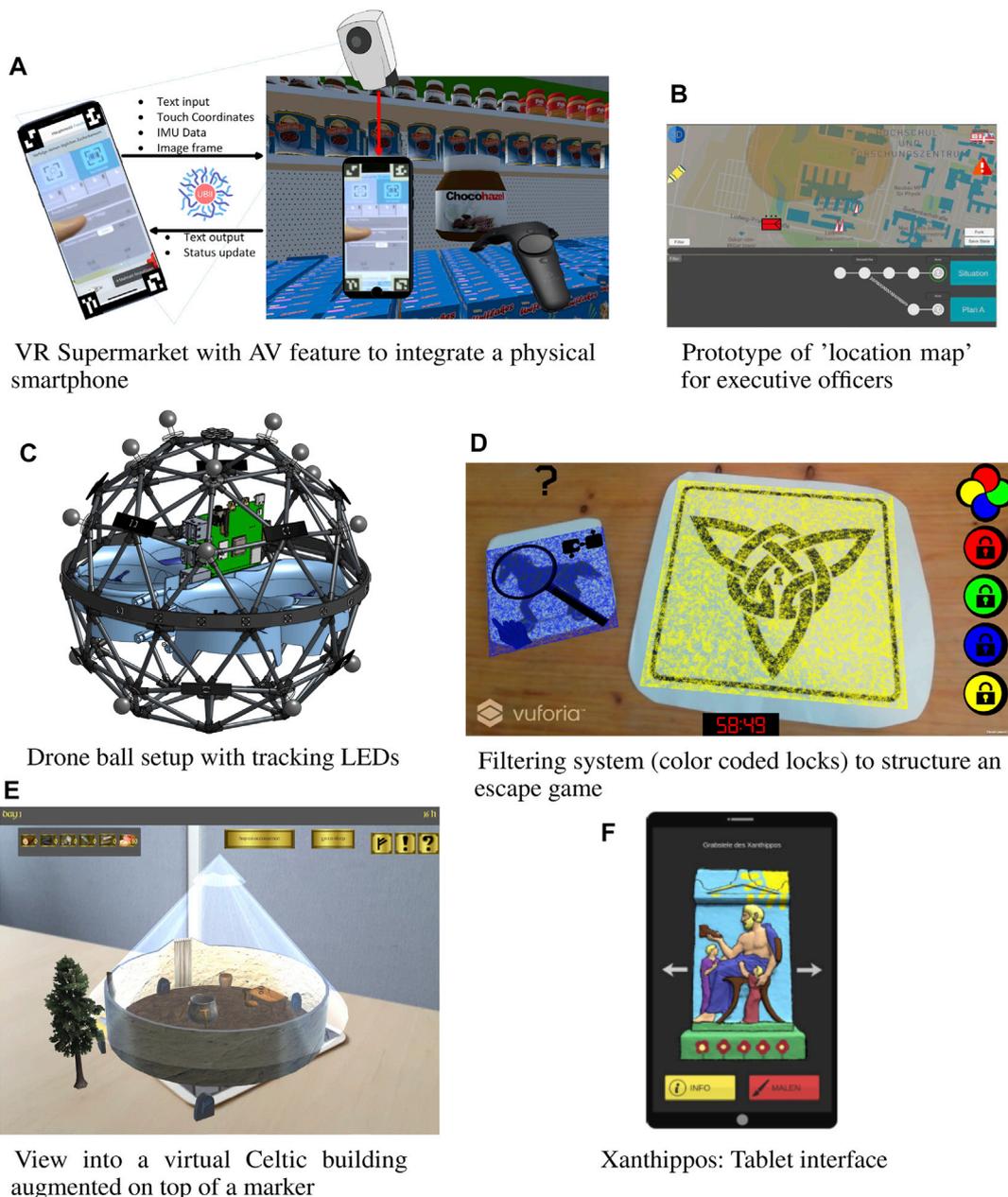


FIGURE 5 | Example applications for different MR scenarios to explore requirements for the presented frameworks.

enhances realism. In the industry, there is a lag of standardized, realistic testing platforms to optimize the grocery shopping experience and the influence of mobile apps on buying decisions.

For our goal, we implemented a realistic replica of a German discounter (digital twin in VR) filled with products, which were modeled to match their real counterparts. To “virtualize” the smartphone, the real-world screen is being tracked with fiducial markers and augmented onto a virtual smartphone canvas (Augmented Virtuality). The markers can be hidden in the frame by extending the pixel colors of the smartphone screen to the respective corners (Diminished Reality). The smartphone

of the experimentee will be connected to the VR PC through Ubi-Interact in the case of a self-made app. This is done to directly improve the tracking of the screen content by transmitting the IMU data in real-time (see **Figure 5A**). This information is used to ensure a realistic rotation of the virtual canvas in the VR environment, hence the perception of using a real smartphone is provided (*Connectivity*). If there are no detected markers for multiple frames, hence the smartphone screen is not in the field of view, the internal tracking will switch to the slower Wi-Fi screen-sharing method. This allows to still offer an updated screen, hence keeping the immersion. On top of that, e.g., a selected food

product can be marked in the VR environment, allowing the evaluation of futuristic AR concepts. A virtual camera can be used as well to scan virtual bar codes of the simulated products (augmentation of the real smartphone screen). Hereby requirements of variable and spontaneous use are included (*Plasticity, Adaptivity*). The implementation does offer an open context, where the market model and smartphone apps can be dynamically swapped, if wanted. With such a combination of MR strategies utilized for real use cases, other platforms can profit from derived guidelines and frameworks.

4.1.2 Construction Industries

Ubiquitous mixed reality is of immense interest for several industries. As an example, we will look at the construction industry, as it has made great progress in the development of the so-called Building Information Model (BIM) in recent years and is thus a pioneer in the development of an industry-specific standard for linking spatial and functional data.

The construction industry has some specifics that should be considered to understand the requirements of this industry for a ubiquitous mixed reality system. First, the characteristics change greatly depending on the phase (Design, Construction, Operation and Maintenance) of the construction project. Since we have the most experience with this phase, we will limit our consideration to the construction phase. During Construction, the most defining characteristic of the use case is of course the complete change of the physical environment. Outdoor scenarios become indoor scenarios, and during the process the planning model and the physical environment slowly converge. This means that when using the model for localization, the current phase of realization must be considered (*N-dimensional Content Reasoning*). Due to incomplete network coverage, connectivity is not reliably achievable. For this reason, AR applications for the construction industry tend to be developed as stand-alone—however, if a ubiquitous system knows that the user will enter such an area, it would be conceivable to provide them with a copy of the required information without having to store the entire BIM model on a mobile device. Additionally, the data needs to be filtered based on roles—however, this level of reasoning is not mixed reality specific and therefore will not be elaborated here. Our definition of *N-dimensional Content Reasoning* is in-fact partially inspired by the 5D (3D space + time + cost) data modeling in building information models (Mitchell, 2012). *Interoperability* and *Scalability* play a rather minor role during the construction phase, as the leading engineering office can influence the hardware used and the number of users—which although large, is known and limited to the stakeholders of the construction project.

In construction, the *Integrability* of Mixed Reality applications into an existing standard is particularly critical. Not only can Mixed Reality be used to visualize BIM data or detect discrepancies, the BIM data can also enhance Mixed Reality applications. For example, they contain information about the materials that are installed in the environment. It is possible that future tracking systems will be able to draw conclusions about their own usability based on these material properties. In addition, exclusion zones can be defined. The visual overlay of

a user's field of vision could be automatically deactivated as soon as they are in a dangerous area (stairs, for example) that requires their attention. Another requirement (especially for AR-based annotations) is a high visual overlay accuracy because annotations pointing on a wrong pipe or place on a wall might do more harm than a missing annotation. Therefore, *Quality Assurance* and *Integrability* as technical requirements play a big role for construction industries.

4.2 Civil Services

4.2.1 Mixed Reality Concept for Emergency Forces Assisting in Catastrophic Situations

In cooperation with “Werkfeuerwehr TUM” and “Feuerwehr München” as experts in rescue and emergency situations, we discussed how technology can assist the current commanding structure and communication process in a large scale emergency situation as well as daily business to accelerate the learning curve with new technology brought in. Among potential scenarios are fires in large living blocks, mass casualty incidents, or natural disasters. The proposed system needs to be highly robust in short and long term usage, and just as flexible as the classical approach with pen and paper. Here we identified a central need to enhance the existing communication process in different critical situations, e.g., when different parties communicate in different locations to coordinate their efforts while personal or coordination changes happen. Another major problem is the well-established command and operation structure in emergency units, so the major challenge is to reduce the initial barrier of new technology for highly specialized and experienced staff.

This discussion leads into a concept for a Mixed Reality setup for coordination center during long term or catastrophic situations. The central element for all planning and coordination operations is a “location map” relocated on a multitouch table as shared working space for dedicated officers in the fire department. Traditional 2D planning is already prototyped with a first mockup to handle typical coordination tasks of available units, showed in **Figure 5B**. This interactive map as shared resource can be easily linked to other dedicated location, e.g., on the accidental location or other authorities. This map is designed to act as close to the traditional planning tasks but offers possibilities of automatic updates, like GPS positions of included forces.

As Mixed Reality part, other participants can use tablets (with video-see-through) or HMDs (with optical-see-through) to get additional input options and visual information layers according to their specific task. Holograms can be used to transport of three-dimensional information, like building complex, in combination with a two-dimensional view on the shared display. As a next step, depth sensors can be integrated which observe the area around the table to allow non HMD-wearing people to point at, or interact with data without direct input on the “location map.” This can also be enhanced with haptics, as Ultraleap⁵ shows in their demonstration. As a final extension, it should be possible to communicate in such a local MR-Application with other

⁵<https://www.ultraleap.com/haptics/>

participants in a full VR or similar MR-Environment to communicate on the same data in a three-dimensional way, without the need of physical presence. In addition, there are several tasks similar to operation and maintenance tasks presented in **Section 4.1.2** that can be applied during operation planning or debriefing.

Our solution includes different hardware and sensors used by multiple users, problems discussed in *Interoperability* and *N-Dimensional Reasoning* are faced in this application. Also, to enhance the input space of users, external sensors need to cooperate to observe the entire work space. Which allows virtual content to be visible and intractable from different points of views and with different devices. Each user might use a different type of input controller for manipulating elements on our “location map” or with secondary devices introduces challenges of *Plasticity* and *Adaptivity* presented in **Section 2.4.1**. In this scenario, we see UbiVis and Ubi-Interact as possible solutions, to overcome challenges in visualization of data in a full MR-application with different kind of spatialized UIs and adaptive to the user’s needs. Also, the experience needs to be as similar as possible independent of the user’s interaction and input space, but with an easy transition from wearing an HMD or using the shared “location map” on a central table to allow a suited learning curves for users.

4.3 Sports and Entertainment

4.3.1 AR Tennis

In this tennis simulation, either a person is competing against a virtual, augmented opponent or another human player. The players are wearing varying and affordable AR HMDs, e.g., smartphones. The interaction with virtual objects will occur *via* an additional handheld smartphone, which will be used as a tennis bat. SLAM-based indoor tracking realized through ARCore or ARKit is used. The playing field can vary in size and layout, e.g., 5 m long and 3 m wide or 10 m long and 5 m wide, through the help of scene understanding (*N-Dimensional Content Reasoning*). Players are hitting the virtual ball *via* movement of the smartphone measured through the integrated IMU in real-time. Both smartphones are connected in real-time through Ubi-Interact and the bat smartphone will transmit IMU data to the HMD smartphone. Hereby, hardware is different for each device, which results in varying sensor quality, which should be accounted for through the *Interoperability* and *Extensibility* requirement. Because of the limitation of inertial tracking with one IMU, an attached 3D marker on the smartphone will be occasionally tracked (sensor fusion) with the smartphone camera to correct its 6DoF pose. In the multiplayer scenario, a player can join spontaneously and both augmentation smartphones need to share a common world representation based on point clouds (*Connectivity*). The IMU with camera data fusion and the alignment of both coordinate systems represent complex spatial requirements.

4.3.2 Superhuman Sports

Dynamic use cases in the newly defined genre of superhuman sports demand for an asynchronous Mixed Reality approach to realize competitive AR multiplayer sports games. These games

have the goal to overcome the limitations of the human body by utilizing technology (Eichhorn et al., 2019). A foundation of this genre is competition, hence games involve either a player competing against a virtual opponent or another human player. We now describe a game concept with existing demonstrator and how we can utilize the newly developed frameworks in the future.

The game **Catching the Drone** is part of superhuman sports, where traditionally the focus lies on purely virtual game elements in AR games, because of the ability to easily manipulate their behavior and involvement of “superhuman moves.” But this results in unnatural interactions, where controllers need to be involved to influence virtual game objects, and hence this limits the option for a natural game design. To achieve the need for “superhuman moves,” an engine propelled, augmented ball in the form of a drone (Eichhorn et al., 2020) with a cage acting as a playing ball has been envisioned. The drone achieves a limited range of agencies because it is equipped with a camera and sensors, which are used to detect players with the goal to avoid them (see **Figure 5C**). The playing field has a size of half a football field, hence there are concrete requirements also for team size (*Plasticity*). To score points, it is necessary to throw the augmented drone ball through ring targets, which are positioned at the end of each side of the playing field. The augmentation is achieved through varying optical see-through HMDs (heterogeneous devices as part of *Interoperability*, *Extensibility* as well as *Plasticity*, *Adaptivity*) and a 6DoF tracking algorithm. This will help to visualize the “superhuman moves” and the state of the drone for the other players. If a player catches it, rules of positioning will take effect and dictate the possibilities of the actions of this player (*N-Dimensional Content Reasoning*). To achieve a persistent state of the game for all the HMDs of the players and flying drone, complex spatial requirements are present and real-time networking requirements with low latency will be necessary (*Connectivity*). In terms of *Plasticity* and *Adaptivity*, all players and spectators are connected to one central game logic (server) and can join at any given time (*Scalability*). Based on pricing constraints, number of players and the need for optical see-through HMDs, it should be possible to swap platforms, tracking algorithms and networking solutions. Ubi-Interact is involved as key technology and integrated as variable networking solution for the Superhuman Sports Platform (Eichhorn et al., 2020). The platform targets Mixed Reality games and separates game logic, devices and specific solutions implemented in, e.g., OpenCV or with neural networks.

4.3.3 ARescape–AR Escape Room

In recent years, Escape Rooms have become very popular. It is a great social experience to solve riddles and puzzles in teams. In our project ARescape (Plecher et al., 2020a) we used AR to transfer this concept to a gaming application that can be played in a competitive (two teams) and cooperative (one team) mode independent of the users’ location. The application became an alternative for real escape rooms, especially in times of the Covid-19 pandemic.

The game uses marker-based AR quizzes the players should solve within a certain time limit. Markers are distributed over the room or the table. Each marker is representing a quiz to be solved. The quizzes are restricted with color-coded locks to guarantee a certain order (see **Figure 5D**).

In cooperative mode, one team within one room (local network) or at different places (internet connection) is using multiple tablets running the same application. All information is shared. In competitive mode, each team has a separate room (plus remote players) and instance of the game. Within the teams, the information is shared on multiple tablets like in cooperative mode (*Connectivity*).

In competitive mode, it is also possible to earn points by solving “side-quests.” These can be spent on additional hints (bonus) or to make the other team’s life harder (malus). Necessary information depending on the selected game mode is shared, enabling the interaction between both teams regarding the malus system. In the future, the state (un-/solved) of physical gadgets will also be shared (*N-Dimensional Content Reasoning*).

While engineering escape rooms, ideally all gadgets and elements involved are reusable and rearrangeable to keep things fun and interesting. A magic wand able to levitate and set objects on fire, for example, could have many applications in different fantasy settings. Thinking about gadgets like this presents them as an extreme case for *Plasticity*. The more general the interfaces and affordances of individual gadgets can be described and the bigger and more varied the set of them is, the better we can freely combine them and open design space for very creative puzzles for developers and/or solutions by players (*N-Dimensional Content Reasoning*) making the whole system feel less rigid and artificial.

4.4 Cultural Heritage

4.4.1 *Oppidum*—A Serious AR Game About the Celts

Oppidum (Plecher et al., 2019) is a serious AR game that transfers knowledge about Celtic history and culture to the player. It is designed for two players of all ages who are playing the game while either sitting in the same room at one table (like a board game) or being in different locations (remote scenario). The game uses two tablets to present Celtic buildings modeled on the basis of archaeological findings in AR view. The player has to manage a Celtic village and to acquire resources for producing goods or setting up buildings. Tracked markers define the position of these buildings. When reducing the distance between camera and marker, the player can inspect the interior of the virtual buildings (see **Figure 5E**) and gain historical knowledge. The goal is to collect victory points, which can be obtained in different ways. This can be done by acting cleverly in the round-based game, by successfully completing quests (drawn quest cards) or by answering questions in the interactive quiz (quiz war) against the opponent.

The information about the win state (reaching seven victory points) or a list of unlocked technology/buildings must be monitored and provided. Moreover, the latter is important because the questions in the quiz are based on the shared possible knowledge between both players.

In the current version, the game logic runs on the clients (tablets), the necessary statuses are provided by the server. So a crash of a client results in losing the game context. A solution could be to export the game logic to an external process which is independent of clients. The game states could be externalized to a separate process as well to avoid, e.g., cheating. This way, it would be possible to save game states and to resume the running game later on (*N-Dimensional Content Reasoning*).

In the future, we will expand the game to be played with different devices and displays (*Plasticity*). This will result in the inclusion of different tracking methods. One player uses the already explained procedure with tablet and marker-based tracking in a local environment. The opponent uses an HMD with markerless tracking and places his buildings in the open landscape. Therefore, it would be possible to actually enter the buildings virtually, requiring on the one hand a switch from environment mounting to world mounting (Tönnis et al., 2013) and moreover on the other hand from AR to VR.

4.4.2 *Xanthippos*—Projective AR in a Museum

In cooperation with the “Museum für Abgüsse Klassischer Bildwerke” (“Museum of Casts of Classical Statues”) in Munich, we developed a projective AR application (Plecher et al., 2020b) to offer visitors new possibilities to interact with the exhibits. Many statues and stelae of antiquity were colored at that time. However, there is rarely a reliable reconstruction of the color. In this project, we give the visitors the possibility to paint the tomb stele of *Xanthippos* with the help of a tablet (see **Figure 5F**) and projective AR as they like it. The painting process is done on a tablet that is installed in public. A projector connected *via* Wi-Fi overlays the images on the stele which is placed inside a booth. Furthermore, the application has a guiding mode. Here, information about the art historical background or the life of *Xanthippos* is displayed on the tablet, while at the same time the corresponding parts of the exhibit are highlighted by the projector. The coloring selected by the visitors is projected (image data) in near real time onto the stele.

In the future, the application could be opened for simultaneous use by multiple users. They could color the stele using different devices at the same time (*Plasticity, Scalability*). Virtual instances would also be possible, in which the exhibit is augmented into the room for coloring (*N-Dimensional Content Reasoning*). The collaboration of multiple devices in a public space requires some restrictions to maintain privacy and security. Likewise, the creation process of the “artwork” must be monitored for unwanted comments or signs. Until now, this has been done by a supervisor (*Data Security and Privacy*).

4.5 Common Factors for all Applications and Mapping to Frameworks

Looking at the various use cases and applications and trying to extrapolate a path into the future what their vision looks like, the challenges to frameworks for the tasks of tracking, visualization and interactivity become clearer (see **Table 1**).

TABLE 1 | Relation between applications, requirements and frameworks. Only special considerations are marked, otherwise a basic need for, e.g., convenience is assumed. T = UbiTrack, I=Ubi-Interact, V=UbiVIs.

	Oppidum	Museum	AR-escape	AR-tennis	VR-shop	SHS	BIM	FeuER
Plasticity, Adaptivity	T,I,V	T,I,V	I	T,I,V	I	T,I	T,I	I,V
Connectivity			I	T	I	T,I		I
Scalability		I			I	V		I
Data Security and Privacy		V			T,V		T,I,V	
N-Dimensional Content reasoning	I	I,V	T,I,V			I	I,V	I,V
Interoperability, Extensibility				T		I,V	I	
Convenience						I,V		I
Quality Assurance							T,V	I
Integrability							I,V	

5 DISCUSSION AND FUTURE WORK

The analysis of the applications reveals research gaps with respect to the Ubi-frameworks. This chapter discusses areas that need further investigation and outlines possible solutions to link the frameworks together.

5.1 Tracking

Regarding tracking, we first need to investigate the different scales. Some applications may only use a limited range as in Oppidum (Section 4.4.1) of this spectrum, but it should certainly be assumed that the full scale up to GPS data might be involved and used in combination as seen in the industrial and civil service examples where we need pinpoint accuracy at a maintenance/building site or collaborative operating room as well as large scale localization and navigation. Although a certain tracking method like marker-based (fiducial marker, objects) or marker-less (SLAM), inertial or GPS might be best suitable and sufficient for one isolated scenario—covering as many methods as possible and allowing their combination opens possibilities in terms of robustness and flexibility in the design of new applications.

Furthermore handheld devices become capable of building their own environment mapping, place themselves within it and even utilize machine learning to identify objects or surfaces Marchesi et al. (2021), it becomes crucial to provide ways of finding common anchors and sharing the same coordinate system/information between devices.

In the superhuman sports genre, an up to this day neglected focus point is the integration of the audience into the game. To achieve that, a game concept such as Catching the Drone (Section 4.3.2) should enable spectators to see augmentations and game information through mobile devices. At the same time, with the focus moving towards multiple smaller-scale devices each gathering information about their shared environment on equal terms also comes the challenge of having to consolidate partial and incomplete or unreliable information. For AR Tennis, the need for fusion of inertial and visual-based tracking has been identified as a challenge. It should not be necessary to reinvent a fitting algorithmic solution for such use cases each time. But rather this should be provided through the framework in combination with networking as part of the *Connectivity* requirement. For the Catching the Drone use case, the need to have varying custom tracking algorithms arose (Eichhorn et al.,

2020). This is part of the Plasticity and Adaptivity requirement, where the need to fluently change the algorithm based on circumstances, such as a different HMD, is important.

5.2 Interaction

Looking at application examples, the requirements and how a system like Ubi-Interact can help in these regards, we identified the following core points:

The origin for the terms *Plasticity and Adaptivity* stem from HCI and user interface considerations. We think for MR their meaning should extend to encompass flexibility in all interfaces between logically separate entities. The biggest challenges relate to 1) describing digital profiles or affordances of devices once and reusing them in other contexts (Sections 4.3.3, 4.2.1), 2) finding suitable pairings for sets of devices relating to their use by agents (Section 4.3.1), 3) treating real and virtual representations of application elements interchangeably (Section 4.1.1) and 4) adjusting application logic and behavior to changes of the environment over time (Sections 4.3.2, 4.1.2). Crucially, any framework should keep base dependencies to a minimum and have minimal presupposed classification hierarchies since they essentially restrict flexibility. Base dependencies are communication channels and shared data format protocols. It should then give maximum flexibility choosing dependencies necessary for desired interactions, without introducing them to the rest of the application. Ubi-Interact makes an effort to keep the common I/O language open and extendable as a minimal base dependency, while offering nonrestrictive Processing Modules to encapsulate functionality.

For *Connectivity*, one interesting challenge apart from hard latency requirements (Section 4.3.2) is conditional communication filters adjusting dynamically to the situation (Section 4.2.1). This goes beyond, e.g., basic publish-subscribe with up-front authorization checks.

Whenever crowds of people are involved (Sections 4.4.2, 4.1.1, 4.3.2, 4.2.1) *Scalability* becomes an issue. Ubi-Interact has yet to evaluate massive numbers of clients—common strategies like broker clustering will be adopted if the dynamic subscription mechanisms can not easily be integrated into existing brokers with *Scalability* features outright.

To give transparent *Data Security and Privacy* measures as required in Sections 4.1.1, 4.1.2, Ubi-Interact will use the dynamic pub-sub structures to establish checks whether the

broker should accept/distribute data. Clear interfaces and I/O signatures of modules are given and open to inspection by participants. Trust and parity checks for running open-source modules are also considered for future development.

To *N-Dimensional Content Reasoning*, lots of contextual information (Sections 4.3.3, 4.1.2) has to be made available in a timely fashion (Section 4.3.2) for MR to expand its range of possibilities. Ubi-Interact here tries to integrate as much information as necessary and provide methods of inquiry in all parts of the system.

In scenarios with a plurality of standards and data schemes as in Section 4.1.2, Ubi-Interact can help with *Integrability* to achieve a standardized data scheme interacting seamlessly with different agents in the building and delivering BIM data, scans, annotations or planning data necessary for them.

5.3 Visualization

Concerning visualization of virtual objects in those diverse and interactive MR scenarios, it is important to solve the following issues. Design and structure of interactive UI elements for users has to be designed and behave in an understandable and intuitive way. Otherwise, users will not tend to accept new ways of interaction possibilities. Adaptive adjustments of UI elements should help to avoid information overload for new users or in situation with high mental load on real world aspects. The UI should also follow user's needs and react to changes in the environment and input capabilities.

In shared environments, content may be visible to users with different hardware capabilities, like in scenario Section 4.2.1. Depending on the user's setup, UI for virtual content needs to be adopted to the user's capabilities. This might also require changes in visual appearance for virtual objects, but gives applications the opportunity to use spatial knowledge of environments to increase usability and information density.

In current hardware configuration for stand-alone AR or VR goggles, the available computation resources are rather limited. With respect to the hardware development, this bottleneck might be ignorable. But nevertheless, it should be important to limit resources for adaptive content. Otherwise, user experience would be reduced for quality and acceptance.

To gain the optimal user experience, all kind of virtual content needs to fit to the actual user. This includes hardware specifications like FoV, resolution, optical characteristics but also with same priority possible input capabilities and user's environment.

5.4 Open Questions

Since we opted for a separation of scope with the approach of having multiple frameworks, one of the main questions is where to locate and manage state and information that takes influence from multiple sources spread over different frameworks and potentially consolidates and merges it. This issue occurs at runtime and during development.

For development, a practical solution would be to combine or integrate all frameworks into an engine that drives application development as an integrated system. Compared to, for example,

traditional game engines, this would still be a distributed inter-process system. This of course loses some freedom of combining solutions originally intended, but would give developers a singular and convenient package when trying to build applications from the ground up. For such an engine, the description and handling of the abstract concepts of combined virtual and physical agents, environments and devices needs to be investigated further.

To establish runtime environments, a practical step would be to deliver a platform on which content in the form of multiple applications can be managed and integrated with other aspects like user management and information priority. It might be possible to integrate with existing projects like Arena (Pereira et al., 2021). An open question is whether there could be a combined mixed reality platform or whether we need specific platforms for different use-cases. The second is conceivable, for example, if we have industrial scenarios with severe IT security requirements or sports and entertainment scenarios with extreme scalability and connectivity requirements.

6 CONCLUSION

The approaches towards ubiquitous mixed reality are naturally as manifold as the concept of reality itself. The people working on this vision, from industry, consortia and academic research alike, share some ideas - but add different perspectives and requirements. By presenting a broad spectrum of applications, summarizing technical solutions represented by the Ubi-frameworks and collecting crucial requirements from the related work, we hope to deliver a structured mind-set for the discussion about this vision and identify future developments necessary to realize our shared dream.

AUTHOR CONTRIBUTIONS

SW: Ubi-Interact main developer. LR: UbiTrack content, use case of industrial building planning. SL: UbiVis main developer, use case of emergency/catastrophy. CE: use cases of VR supermarket/catching the drone/display fusion/AR tennis. DP: use cases of Oppidum/Xanthippos/ARescape. DD: Ubi-Interact contribution, figures. GK: use case of industrial applications of UbiTrack.

FUNDING

This project/research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3), from the project "Zentrales Innovationsprogramm Mittelstand" of the German Federation of Industrial Research Associations (ZF4086002SS7) as well as the project "Enable 2.0" of the Federal Ministry of Education of Research (FKZ 01EA 1807A).

REFERENCES

- Alt, T., Schreiber, W., Wohlgenuth, W., and Zimmermann, P. (2012). "Das Verbundprojekt AVILUS," in *Virtuelle Techniken im industriellen Umfeld*. Editors W. Schreiber and P. Zimmermann (Berlin, Heidelberg: Springer-Verlag), 4–18. doi:10.1007/978-3-642-20636-8_2
- A. Quigley, S. Diamond, P. Irani, and S. Subramanian (Editors) (2014). *Proceedings of the 16th International Conference on Human-Computer Interaction with mobile Devices & Services - MobileHCI '14* (New York, New York, USA: ACM Press). doi:10.1145/2628363
- Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., et al. (2001). "Design of a Component-Based Augmented Reality Framework," in *4th International Symposium on Augmented Reality (ISAR 2001)* (IEEE Computer Society), 45–54. doi:10.1109/ISAR.2001.970514
- Bauer, M., Schlegel, M., Pustka, D., Navab, N., and Klinker, G. (2006). "Predicting and Estimating the Accuracy of Vision-Based Optical Tracking Systems," in *Proc. IEEE International Symposium on Mixed and Augmented Reality* (New York: ISMAR'06), 43–51.
- Behr, J., Blach, R., Bockholt, U., Harth, A., Hoffmann, H., Huber, M., et al. (2017). "ARVIDA-referenzarchitektur," in *Web-basierte Anwendungen Virtueller Techniken*. Editors W. Schreiber, K. Zühl, and P. Zimmermann (Berlin, Heidelberg: Springer-Verlag), 117–191. doi:10.1007/978-3-662-52956-0_3
- Benzina, A., Dey, A., Toennis, M., and Klinker, G. (2012). "Empirical Evaluation of Mapping Functions for Navigation in Virtual Reality Using Phones with Integrated Sensors," in *Asia Pacific Conference on Computer Human Interaction, APCHI '12* (New York: ACM), 149–158. doi:10.1145/2350046.2350078
- Blanco-Novoa, Ó., Fraga-Lamas, P., Vilar-Montesinos, M. A., and Fernández-Caramés, T. M. (2020). Creating the Internet of Augmented Things: An Open-Source Framework to Make IoT Devices and Augmented and Mixed Reality Systems Talk to Each Other. *Sensors* 20, 3328. doi:10.3390/s20113328
- Bondi, A. B. (2000). "Characteristics of Scalability and Their Impact on Performance," in *Second International Workshop on Software and Performance, WOSP 2000, Ottawa, Canada, September 17-20, 2000* (New York: ACM), 195–203. doi:10.1145/350391.350432
- Bowman, S. L., Atanasov, N., Daniilidis, K., and Pappas, G. J. (2017). "Probabilistic Data Association for Semantic Slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (New York: IEEE), 1722–1729. doi:10.1109/icra.2017.7989203
- Browne, D. (2016). *Adaptive User Interfaces*. Amsterdam: Elsevier.
- Casarin, J., Pacquerraud, N., and Bechmann, D. (2018). Umi3D. *Proc. ACM Hum.-Comput. Interact.* 2, 1–20. doi:10.1145/3274298
- Chen, L., Tang, W., John, N. W., Wan, T. R., and Zhang, J. J. (2018). *Context-Aware Mixed Reality: A Framework for Ubiquitous Interaction*. CoRR abs/1803.05541.
- Cutolo, F., Fontana, U., Carbone, M., D'Amato, R., and Ferrari, V. (2017). "[POSTER] Hybrid Video/Optical See-Through HMD," in *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2017 Adjunct, Nantes, France, October 9-13, 2017* (New York: IEEE Computer Society), 52–57. doi:10.1109/ISMAR-Adjunct.2017.31
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. (2017). "Scannet: Richly-Annotated 3d Reconstructions of Indoor Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. New York: IEEE, 5828–5839. doi:10.1109/cvpr.2017.261
- Daniilidis, K. (1999). Hand-Eye Calibration Using Dual Quaternions. *Int. J. Robotics Res.* 18, 286–298. doi:10.1177/02783649922066213
- Davison (2003). "Real-time Simultaneous Localisation and Mapping with a Single Camera," in *Proceedings Ninth IEEE International Conference on Computer Vision*. New York: IEEE, 2, 1403–1410. doi:10.1109/ICCV.2003.1238654
- De Guzman, J. A., Thilakarathna, K., and Seneviratne, A. (2019). Security and Privacy Approaches in Mixed Reality: A Literature Survey. *ACM Comput. Surv. (Csur)* 52, 1–37. doi:10.1145/3359626
- Durrant-Whyte, H., and Henderson, T. C. (2016). "Multisensor Data Fusion," in *Springer Handbook of Robotics*. Editors B. Siciliano and O. Khatib (Heidelberg: Springer), 867–896. Springer Handbooks. doi:10.1007/978-3-319-32552-1_35
- Eck, U., Pankratz, F., Sandor, C., Klinker, G., and Laga, H. (2015). Precise Haptic Device Co-location for Visuo-Haptic Augmented Reality. *IEEE Trans. Vis. Comput. Graphics* 21, 1427–1441. doi:10.1109/TVCG.2015.2480087
- Eichhorn, C., Jadid, A., Plecher, D. A., Weber, S., Klinker, G., and Itoh, Y. (2020). "Catching the Drone – A Tangible Augmented Reality Game in Superhuman Sports," in *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (New York: IEEE), 24–29.
- Eichhorn, C., Lurz, M., Plecher, D. A., Weber, S., Wintergerst, M., Kaiser, B., et al. (2021). "Inspiring Healthy Food Choices in a Virtual Reality Supermarket by Adding a Tangible Dimension in the Form of an Augmented Virtuality Smartphone," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (New York: IEEE), 548–549. doi:10.1109/vrw52623.2021.00156
- Eichhorn, C., Plecher, D. A., Inami, M., and Klinker, G. (2019). "Physical Objects in AR Games – Offering a Tangible Experience," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (New York: IEEE), 1801–1806.
- Fehr, M., Furrer, F., Dryanovski, I., Sturm, J., Gilitschenski, I., Siegwart, R., et al. (2017). "Tsdf-based Change Detection for Consistent Long-Term Dense Reconstruction and Dynamic Object Discovery," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (New York: IEEE), 5237–5244. doi:10.1109/icra.2017.7989614
- Figueroa, P., Green, M., and Hoover, H. J. (2002). "Intml: a Description Language for Vr Applications," in *Proceedings of the Seventh International Conference on 3D Web Technology*. Salinas, CA: Web3D Consortium, 53–58.
- Fleck, P., Calepso, A. S., Sedlmair, M., and Schmalstieg, D. (2022). Ragrug: A Toolkit for Situated Analytics. *IEEE Trans. Vis. Comput. Graph.* (Submitted for publication). doi:10.1109/TVCG.2022.3157058
- Horn, B. K. P. (1987). Closed-form Solution of Absolute Orientation Using Unit Quaternions. *J. Opt. Soc. Am. A* 4, 629–642. doi:10.1364/JOSAA.4.000629
- Huber, M., Michael, S., and Klinker, G. (2009). "Temporal Calibration in Multisensor Tracking Setups," in *8th IEEE and ACM International Symposium on Mixed and Augmented Reality* (New York: ISMAR), 201–212. doi:10.1109/ismar.2009.5336465
- Huber, M., Pustka, D., Keitler, P., Florian, E., and Klinker, G. (2007). "A System Architecture for Ubiquitous Tracking Environments," in *Proceedings of the 6th International Symposium on Mixed and Augmented Reality* (New York: ISMAR), 211–214. doi:10.1109/ismar.2007.4538849
- Huber, M., Schlegel, M., and Klinker, G. (2014). "Application of Time-Delay Estimation to Mixed Reality Multisensor Tracking," in *Journal of Virtual Reality and Broadcasting Düsseldorf*, 11.
- Itoh, Y., and Klinker, G. (2014). "Interaction-free Calibration for Optical See-Through Head-Mounted Displays Based on 3D Eye Localization," in *2014 IEEE Symposium on 3D User Interfaces* (New York: 3DUI), 75–82. doi:10.1109/3DUI.2014.6798846
- Keitler, P., Becker, B., and Klinker, G. (2010a). "Management of Tracking for Industrial AR Setups," in *9th IEEE International Symposium on Mixed and Augmented Reality, ISMAR* (New York: IEEE Computer Society), 73–82. doi:10.1109/ISMAR.2010.5643553
- Keitler, P., Pustka, D., Huber, M., Ehtler, F., and Klinker, G. (2010b). "Management of Tracking for Mixed and Augmented Reality Systems," in *The Engineering of Mixed Reality Systems*. Editors E. Dubois, P. Gray, and L. Nigay (Heidelberg: Springer-Verlag), 73–82. doi:10.1007/978-1-84882-733-2_13
- Keitler, P., Schlegel, M., and Klinker, G. (2008). "Indirect Tracking to Reduce Occlusion Problems," in *Advances in Visual Computing, Fourth International Symposium, ISVC* (Berlin: Springer), 224–235. 5359(2) of Lecture Notes in Computer Science. doi:10.1007/978-3-540-89646-3_22
- Klein, G., and Murray, D. (2007). "Parallel Tracking and Mapping for Small AR Workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. New York: ACM, 225–234. doi:10.1109/ISMAR.2007.4538852
- Klinker, G., Najafi, H., Sielhorst, T., Sturm, F., Ehtler, F., Isik, M., et al. (2004). "FixIt: An Approach towards Assisting Workers in Diagnosing Machine Malfunctions," in *Proc. Of the International Workshop Exploring the Design and Engineering of Mixed Reality Systems - MIXER*.
- Krekhov, A., Grüniger, J., Baum, K., McCann, D., and Krüger, J. (2016). "Morphableui: a Hypergraph-Based Approach to Distributed Multimodal Interaction for Rapid Prototyping and Changing Environments," in *The 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2016A: Pilsen, Czech Republic. Plzen-Radnice: WSCG/Vaclav Skala*.

- Lacoste, J., Duval, T., Arnaldi, B., Maisel, É., and Royan, J. (2016). "D3part: A New Model for Redistribution and Plasticity of 3d User Interfaces," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)* (New York: IEEE), 23–26. doi:10.1109/3dUI.2016.7460026
- Lacoste, J., Duval, T., Arnaldi, B., Maisel, E., and Royan, J. (2015). "Plasticity for 3d User Interfaces: New Models for Devices and Interaction Techniques," in *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. New York: ACM, 28–33.
- LaViola, J. J., Kruijff, E., McMahan, R. P., Bowman, D. A., and Poupyrev, I. (2017). *3D User Interfaces: Theory and Practice*. second edition edn. Boston, Massachusetts, United States: Addison-Wesley. Pearson always learning (Boston and Columbus and Indianapolis and New York and San Francisco and Amsterdam and Cape Town and Dubai and London and Madrid and Milan and Munich and Paris and Montreal and Toronto and Delhi and Mexico City and São Paulo and Sidney and Hong Kong and Seoul and Singapore and Taipei and Tokyo).
- Mantzios, V.-M., Apostolopoulos, T., and Beltramello, O. (2014). "Augmented Reality Applications Assisting Maintenance Activities in Extreme Environments: Hci Issues," in *HCI International 2014 - Posters' Extended Abstracts*. Editor C. Stephanidis (Heidelberg: Springer International Publishing), 613–618. doi:10.1007/978-3-319-07857-1_108
- Marchesi, G., Eichhorn, C., Plecher, D. A., Itoh, Y., and Klinker, G. (2021). Envsam: Combining Slam Systems and Neural Networks to Improve the Environment Fusion in Ar Applications. *Ijgi* 10, 772. doi:10.3390/ijgi10110772
- Milgram, P., and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. *IEICE TRANSACTIONS Inf. Syst.* 77, 1321–1329.
- Mitchell, D. (2012). "5d BIM: Creating Cost Certainty and Better Buildings," in *2012 RICS Cobra Conference*. Athens: RICS.
- Myers, B., Hudson, S. E., and Pausch, R. (2000). Past, Present, and Future of User Interface Software Tools. *ACM Trans. Comput.-Hum. Interact.* 7, 3–28. doi:10.1145/344949.344959
- Newcombe, R. A., Fitzgibbon, A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., et al. (2011). "Kinectfusion: Real-Time Dense Surface Mapping and Tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. New York: IEEE, 127–136. doi:10.1109/ISMAR.2011.6092378
- Newman, J., Bornik, A., Pustka, D., Ehtler, F., Huber, M., Schmalstieg, D., et al. (2007). "Tracking for Distributed Mixed Reality Environments," in *Workshop on Trends and Issues in Tracking for Virtual Environments at the IEEE Virtual Reality Conference (VR'07)*.
- Normand, J.-M., Sanchez-Vives, M. V., Waechter, C., Giannopoulos, E., Grosswindhager, B., Spanlang, B., et al. (2012). Beaming into the Rat World: Enabling Real-Time Interaction between Rat and Human Each at Their Own Scale. *PLoS one* 7, e48331. doi:10.1371/journal.pone.0048331
- Ohlenburg, J., Broll, W., and Lindt, I. (2007). "DEVAL - A Device Abstraction Layer for VR/AR," in *International Conference on Universal Access in Human-Computer Interaction* (Heidelberg: Springer), 497–506. doi:10.1007/978-3-540-73279-2_56
- OpenJS Foundation & Contributors (2013). *Node-red*. v: OpenJS Foundation.
- Pankratz, F., and Klinker, G. (2015). "AR4AR: Using Augmented Reality for Guidance in Augmented Reality Systems Setup," in *Proceedings of the 14th IEEE International Symposium on Mixed and Augmented Reality ISMAR*. New York: IEEE, 140–143. Extended poster.
- Pereira, N., Rowe, A., Farb, M. W., Liang, I., Lu, E., and Riebling, E. (2021). "Arena: The Augmented Reality Edge Networking Architecture," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (New York: IEEE), 479–488. doi:10.1109/ismar52148.2021.00065
- Pielot, M., Church, K., and de Oliveira, R. (2014). "An In-Situ Study of mobile Phone Notifications," in *Proceedings of the 16th International Conference on Human-Computer Interaction with mobile Devices & Services - MobileHCI '14*. Editors A. Quigley, S. Diamond, P. Irani, and S. Subramanian (New York, New York, USA: ACM Press), 233–242. doi:10.1145/2628363.2628364
- Piumsombon, T., Day, A., Ens, B., Lee, Y., Lee, G., and Billingham, M. (2017). "Exploring Enhancements for Remote Mixed Reality Collaboration," in *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. SA '17. doi:10.1145/3132787.3139200
- Plecher, D. A., Eichhorn, C., Köhler, A., and Klinker, G. (2019). "Oppidum - A Serious-AR-Game about Celtic Life and History," in *International Conference on Games and Learning Alliance* (Heidelberg: Springer), 550–559. doi:10.1007/978-3-030-34350-7_53
- Plecher, D. A., Ludl, M., and Klinker, G. (2020a). "Designing an AR-Escape-Room with Competitive and Cooperative Mode," in *GI VR/AR Workshop*. Editors B. Weyers, C. Lürig, and D. Zielasko (Bonn: Gesellschaft für Informatik e.V.).
- Plecher, D. A., Ulschmid, A., Kaiser, T., and Klinker, G. (2020b). "Projective Augmented Reality in a Museum: Development and Evaluation of an Interactive Application," in *ICAT-EGVE 2020 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*. Editors F. Argelaguet, R. McMahan, and M. Sugimoto (Geneve: The Eurographics Association). doi:10.2312/egve.20201258
- Pustka, D. (2006). "Construction of Data Flow Networks for Tracking in Augmented Reality Applications," in *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR* (New York: Gesellschaft für Informatik), 13–24.
- Pustka, D., Huber, M., Bauer, M., and Klinker, G. (2006). "Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems," in *Proc. IEEE International Symposium on Mixed and Augmented Reality* (New York: ISMAR'06), 88–97. doi:10.1109/ismar.2006.297799
- Pustka, D., Huber, M., Ehtler, F., and Keitler, P. (2007). *UTQL: The Ubiquitous Tracking Query Language v1.0*. München, Germany: Institut für Informatik, Technische Universität München. Tech. Rep. TUM-10718.
- Pustka, D., Huber, M., Waechter, C., Ehtler, F., Keitler, P., Newman, J., et al. (2011). Automatic Configuration of Pervasive Sensor Networks for Augmented Reality. *IEEE Pervasive Comput.* 10, 68–79. doi:10.1109/MPRV.2010.50
- Pustka, D., Hulss, J.-P., Willneff, J., Pankratz, F., Huber, M., and Klinker, G. (2012). "Optical Outside-In Tracking Using Unmodified mobile Phones," in *11th IEEE International Symposium on Mixed and Augmented Reality, ISMAR* (New York: IEEE Computer Society), 81–89. doi:10.1109/ISMAR.2012.6402542
- Pustka, D., and Klinker, G. (2008). "Dynamic Gyroscope Fusion in Ubiquitous Tracking Environments," in *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR* (New York: IEEE Computer Society), 13–20. doi:10.1109/ISMAR.2008.4637317
- Pustka, D., Willneff, J., Wenisch, O., Lükewille, P., Achatz, K., Keitler, P., et al. (2010). "Determining the point of Minimum Error for 6DOF Pose Uncertainty Representation," in *Proceedings of the 9th International Symposium on Mixed and Augmented Reality* (New York: ISMAR), 37–46. doi:10.1109/ismar.2010.5643548
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). "Ros: an Open-Source Robot Operating System," in *ICRA Workshop on Open Source Software* (Kobe, Japan: IEEE), 3, 5.
- Russell, S., and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. 4th edition. Hoboken, NJ: Pearson.
- Sandor, C., and Klinker, G. (2005). A Rapid Prototyping Software Infrastructure for User Interfaces in Ubiquitous Augmented Reality. *Pers Ubiquit Comput.* 9, 169–185. doi:10.1007/s00779-004-0328-1
- Schwerdtfeger, B., and Klinker, G. (2008). "Supporting Order Picking with Augmented Reality," in *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008*, Cambridge, UK, 15–18th September 2008 (New York: IEEE Computer Society), 91–94. doi:10.1109/ISMAR.2008.4637331
- Shimbun, N. K. (1989). *Poka-yoke: Improving Product Quality by Preventing Defects*. Boca Raton: CRC Press.
- Sielhorst, T., Bauer, M., Wenisch, O., Klinker, G., and Navab, N. (2007). "Online Estimation of the Target Registration Error for N-Ocular Optical Tracking Systems," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 652–659.
- Sosin, A., and Itoh, Y. (2019). "WARP: Contributional Tracking Architecture towards a Worldwide Augmented Reality Platform," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct* (New York: ISMAR-Adjunct), 71–72. doi:10.1109/ismar-adjunct.2019.00032
- Swan, J. E. (2018). "The Replication Crisis in Empirical Science: Implications for Human Subject Research in Mixed Reality," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct* (New York: ISMAR-Adjunct), xxxvi. doi:10.1109/ISMAR-Adjunct.2018.00019
- Tahara, T., Seno, T., Narita, G., and Ishikawa, T. (2020). "Retargetable Ar: Context-Aware Augmented Reality in Indoor Scenes Based on 3d Scene Graph," in *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (New York: IEEE), 249–255. doi:10.1109/ismar-adjunct51615.2020.00072
- Thevenin, D., and Coutaz, J. (1999). Plasticity of User Interfaces: Framework and Research Agenda. *Interact* 99, 110–117.

- Tönnis, M., Plecher, D. A., and Klinker, G. (2013). Representing Information - Classifying the Augmented Reality Presentation Space. *Comput. Graphics* 37, 997–1011. doi:10.1016/j.cag.2013.09.002
- Waechter, C. A. L., Pustka, D., and Klinker, G. J. (2013). “Real-time Monocular People Tracking by Sequential Monte-Carlo Filtering,” in *Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications* (New York: ACM), 4. doi:10.1145/2466715.2466728
- Waechter, C., Huber, M., Keitler, P., Schlegel, M., Klinker, G., and Pustka, D. (2010). “A Multi-Sensor Platform for Wide-Area Tracking,” in *2010 IEEE International Symposium on Mixed and Augmented Reality*, 275–276. doi:10.1109/ISMAR.2010.5643604
- Waechter, C., Pustka, D., and Klinker, G. (2009). “Vision Based People Tracking for Ubiquitous Augmented Reality Applications,” in *8th IEEE and ACM International Symposium on Mixed and Augmented Reality* (Orlando, USA: ISMAR), 221–222. doi:10.1109/ismar.2009.5336452
- Waldow, K., and Fuhrmann, A. (2019). “Using Mqtt for Platform Independent Remote Mixed Reality Collaboration,” in *Mensch und Computer 2019-Workshopband*. Bonn: Gesellschaft für Informatik.
- Weber, S., Ludwig, M., and Klinker, G. (2021). Ubi-Interact: A Modular Approach to Connecting Systems. *ICST Trans. Mobile Commun. Appl.* 6, 170291. doi:10.4108/eai.14-7-2021.170291
- Weiser, M. (1991). The Computer for the 21st Century. *Sci. Am.* 265, 94–104. doi:10.1038/scientificamerican0991-94
- Welch, G., and Foxlin, E. (2002). Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *IEEE Comput. Grap. Appl.* 22, 24–38. doi:10.1109/MCG.2002.1046626
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Weber, Rudolph, Liedtke, Eichhorn, Dyrda, Plecher and Klinker. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.