

# Fighting Pandemics With Augmented Reality and Smart Sensing-Based Social Distancing

Jacky Cao  and Hassan Mehmood, University of Oulu, 90570, Oulu, Finland

Xiaoli Liu  and Sasu Tarkoma , University of Helsinki, 00100, Helsinki, Finland

Ekaterina Gilman, University of Oulu, 90570, Oulu, Finland

Xiang Su , Norwegian University of Science and Technology, 7034, Trondheim, Norway

*In a postpandemic world, remaining vigilant and maintaining social distancing are still crucial so societies can contain the virus and the public can avoid disproportionate health impacts. Augmented reality (AR) can visually assist users in understanding the distances in social distancing. However, integrating external sensing and analysis is required for social distancing beyond the users' local environment. We present DistAR, an android-based application for social distancing leveraging AR and smart sensing using on-device analysis of optical images and environment crowdedness from smart campus data. Our prototype is one of the first efforts to combine AR and smart sensing technologies to create a real-time social distancing application.*

Augmented reality (AR) applications create immersive user experiences by overlaying the physical world with device-rendered virtual annotations. These virtual objects are aligned and occluded as if they are a part of the real world. Users may use two typical types of hardware to access AR augmentations, i.e., ubiquitous smartphones and specialized head-mounted displays (HMDs), such as the Microsoft HoloLens. For smartphone-based AR applications, the device renders virtual augmentations on top of the video feed from the world-facing camera. Comparatively, HMD-based AR uses semitransparent displays, where augmentations are directly displayed in front of users' eyes.<sup>1</sup> Optical video streams are vital data sources to support AR in both types of hardware. Object detection and object recognition help the AR system to understand the environments surrounding users. AR applications can detect and recognize objects and events within the vicinity of users to assist in their daily lives, for example, in citywide user navigation and displaying virtual nutritional information for food items in a supermarket.

Over the past few years, the global COVID-19 pandemic has changed how we interact with and experience the world. For example, societies have to live with the "new normal," which includes short-term solutions, such as practicing social distancing and long-term preventative measures, such as large-scale vaccination programs. AR could support several of these aspects. Social distancing limits the spread of the virus. But, the measure is typically considered the most difficult to enforce as people may underestimate distances and thus be within an unsafe distance of other people. AR applications, such as Sodar<sup>2</sup> and ARaroundME,<sup>3</sup> are developed to overlay social distancing guidelines in real-world environments.

We present the DistAR application, which supports social distancing by leveraging AR technology and smart sensing. DistAR helps users to make safe choices when navigating environments by displaying distance visualizations, which helps them reduce their potential exposure to airborne virus particles. In addition to depth estimations on user devices, the application renders analyzed sensing data [passive infrared (PIR) measurements] as a geographical heatmap to provide additional context to support social distancing (i.e., the crowdedness of locations). Unlike existing AR social distancing applications, we highlight AR along with smart sensing for social distancing

---

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>  
Digital Object Identifier 10.1109/MCG.2022.3229107  
Date of publication 14 December 2022; date of current version 31 January 2023.

as users may need to understand the number of people in a wider environment and not just within their local line of sight. We deployed the application on a smart campus at the University of Oulu, Finland. To the best of our knowledge, this is one of the first contributions of leveraging AR and smart sensing for social distancing.

## RELATED WORK

Undoubtedly, COVID-19 has significantly affected individuals' lifestyles. Practical measurements and technologies are needed to detect, prevent, and mitigate adverse effects caused by COVID-19. Conventional preventative measures, such as face masks, good personal hygiene, and social distancing, provide superior protection compared to other responsive measures, such as mobility restrictions and lockdowns.<sup>4</sup> Technology can play a crucial role in supplementing and enhancing these measures, for example, helping people to social distance. Computing technologies, such as AI, wireless communication, the blockchain, and computer vision, enable social distancing<sup>5,6</sup> and there has been considerable interest in using AI algorithms to calculate distances between people from optical images. Himeur et al.<sup>7</sup> classified the tools in existing social distancing monitoring frameworks into "visual" and "nonvisual." Visual-based social distancing includes analysis of media streams using hand-crafted feature-based, CNN-based, transfer learning-based, and 3D-based algorithms. Comparatively, nonvisual methods include analysis of Bluetooth signals and passive WiFi sensing. Rezaei et al.<sup>8</sup> demonstrated how server-deployed deep neural networks can use typical CCTV camera footage to accurately detect people and the distances between them, and at the same time deal with challenges of occlusion, light variations, etc. Similarly, Shah et al.<sup>9</sup> tested CCTV camera feed analysis using various object detection models to calculate if users follow social distancing norms. These works highlight the need for environmental analysis to obtain data for social distancing. However, their work assumes environments have both normal and bird's eye cameras, which may not always be available, and these algorithms are typically computationally intensive, which may not be feasible for deployment on user devices, such as smartphones. Tanwar et al.<sup>10</sup> demonstrated social distancing using the blockchain and AI, namely, trusted information exchange between entities, such as optical cameras and AI processing layers. While their proposed scheme enables secure real-time social distancing, real-life implementations would be invasive due to the all-opt-in nature. Instead, individuals should have the choice to perform social distancing using their devices

and applications, as seen with Bluetooth-based contact tracing applications.

As an emerging technology, AR has significant potential for supporting social distancing and preventing the spread of COVID-19. Some AR social distancing smartphone applications already exist, such as AR AroundMe Social Distancing by CodingVR<sup>3</sup> and Sodar by Google.<sup>2</sup> These applications overlay social distancing guidelines on the environment, identify where the ground is, and create rings to mark the appropriate safe distance from others (e.g., 1-meter, 1.5-meter, and 2-meter boundaries). While AroundMe is a standalone application for Android and Apple devices, Sodar is a browser and web-based application. However, although both applications support social distancing, they only consider the view of the environment from the optical cameras. We consider external sources and integrate them with AR to support social distancing. QueueSight<sup>11</sup> is another AR social distancing tool, which uses physical projections on the environment to help people correctly socially distance. While an innovative idea, this comes at a cost and requires indoor sensors and projectors to be installed and deployed en masse. We propose using personal devices, such as smartphones, as a more efficient and ubiquitous way to ensure people can quickly access an AR social distancing tool.

## FIGHTING COVID-19 WITH AR AND SMART SENSING

Existing AR applications, such as AR AroundMe Social Distancing<sup>3</sup> and Sodar,<sup>2</sup> typically function offline without requiring access to the Internet or other sensors. Figure 1 presents a social distancing vision using AR and smart sensing. Users accessing the application can view several layers of information in varying combinations to support and ensure safe distances. In this vision, a user's device camera captures images, and the environment is analyzed to detect people and distances. Then, the devices can render this information to the user, e.g., as circles around each person. Second, these areas can be color coded according to risk. For example, red indicates a high possibility of walking very close to another person, and orange and yellow represent lower risks. In addition to estimating risk, the application could also perform pathfinding to allow users to navigate environments safely and avoid contact with as many people as possible. Finally, an AR social distancing application could provide location overviews to aid in deciding if they are safe to enter according to the number of people. The system could achieve this by accessing large-scale city-wide deployments of indoor and outdoor IoT sensors and portable wearable sensors. The sensor data would



**FIGURE 1.** Vision of supporting social distancing during and after the COVID-19 global pandemic using AR and smart sensing technologies.

be stored and analyzed on a central server. Then, for visualizing this information in AR, buildings or rooms could be overlaid with a similar color coding scheme as the individual people risk assessment. Additional augmentations could contain general information, such as location names, estimated number of people, and other relevant contextual information.

These different information layers would ensure users can safely navigate environments and make informed decisions about whether to enter locations. However, this could become confusing and mentally taxing with large amounts of information. Therefore, an AR social distancing application should use selective layers to allow users to view which type of distancing they prefer. Alternatively, the application would intelligently swap between layers according to location and whether social distancing features are required.

Using data streams from sensors in smart spaces is vital for supporting social distancing because carbon dioxide (CO<sub>2</sub>) concentrations highly correlate with aerosols containing COVID-19. Users could use data gathered by CO<sub>2</sub> and PIR motion sensors to gain insights into potential virus transmission risks and the crowdedness of environments.<sup>12</sup> In this research, we have utilized PIR sensors installed in the smart environment of the University of Oulu to estimate area crowdedness. On average, thousands of individuals visit the smart campus, and during previous pandemic lockdowns, the mobility patterns varied from typical norms. Therefore, such environments could act as

potential testbeds for developing social distancing applications and insights for enacting mobility policies in future pandemic scenarios.

## DistAR: AN AR SYSTEM FOR SOCIAL DISTANCING

### Design Considerations

In light of needing a social distancing application, as well as understanding the capabilities of AR and smart sensing, in the following we propose several key considerations that an AR application supporting social distancing should follow.

- 1) Leveraging AR and smart sensing with high accuracy allows social distancing applications to understand the number of people within an environment (i.e., the crowdedness). In this way, a person using the application would feel confident that they could safely and efficiently navigate an environment while performing social distancing. Estimating location crowdedness can be done by analyzing data collected from various environment-based physical sensors.
- 2) Near real-time analysis and rendering of relevant information is essential when considering the shifting nature of people within environments. If the total time between data capture, data analysis, and subsequent information rendering is too large, a user could unknowingly walk into areas

containing many airborne virus particles. Solutions are, therefore, needed to reduce the overall system latency.

- 3) Support for heterogeneous devices ensures an application is usable by the broadest group of people on a large variety of devices. In addition to user devices, device operating systems (OSs) considerations are needed as specific AR framework APIs may not be available across the different OS versions.
- 4) Resource usage of the application should be well monitored and managed. For example, the device battery could quickly deplete if the application processes are unchecked, the system would then reduce the energy for device components, which could impact other running applications and, subsequently, the user's quality of experience (QoE). Screen dimming could make augmentations more challenging to see, and environmental analysis could take more time. The application should, therefore, strike a balance between functionality and impact on device resources.
- 5) Ease of use is essential if social distancing applications are to be used by the broadest possible group of users who may span various age ranges and technical competencies. The application's user interface (UI) should be as simple as possible to reduce mental load when traversing environments while simultaneously looking at smartphones. Users could become inundated and mentally stressed if the application contains too much information.

From the stated requirements for a social distancing application, we develop a prototype system that addresses three of the main requirements, namely:

- 1) leveraging smart sensing and AR with high accuracy;
- 2) near real-time analysis and rendering; and
- 3) ease of use.

We focus on these requirements in particular as they form the core backbone of the application (i.e., social distancing).

To achieve smart sensing and AR with high accuracy, we estimate crowdedness using a large number of sensors constantly monitoring a campus. In this way, we represent the real-time campus status with a high degree of accuracy and provide AR data visualizations. We must consider several aspects for producing near real-time analysis and rendering. One aspect is related to processing smart sensor data. While user devices can retrieve sensor data from a local cloud server, continual real-time

data processing and analysis is not feasible as clients lack computation power. A local cloud or edge server with more powerful resources should execute these rather than on users' AR devices. In this way, the client can retrieve the cloud-analyzed results and present them in a near real-time fashion. Finally, for meeting the requirement of ease of use, we consider the definition of perceived ease of use where a person considers to what degree using a system is free of effort.<sup>13</sup> To achieve this, we keep the number of UI features and screen clutter to a minimum to not mentally overload users and to reduce their overall input effort. For example, our application has a geographical map in a small screen area. In this case, we keep the immediate user attention on the AR experience, and we utilize a button to swap between the transparent and colored opaque visualizations.

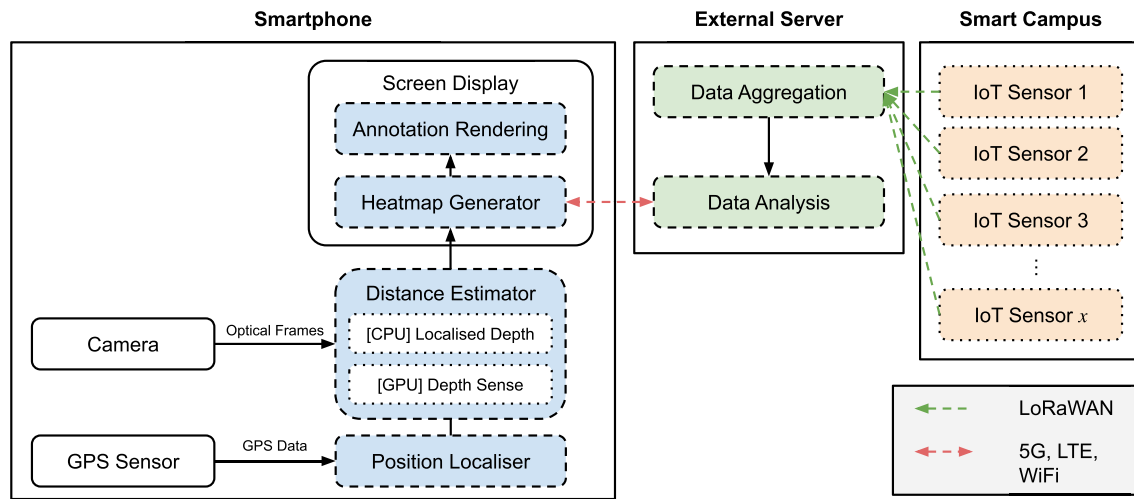
Further iterations of the application should focus on refining and improving the application to fulfill the other requirements, including supporting heterogeneous devices and improving application resource usage efficiency. For example, the Android OS provides several classes for checking, managing, and releasing memory when the device meets predefined system conditions. In addition, more memory-efficient code constructs could help minimize application memory usage.

How users use the application in specific environments and scenarios should also be considered. When using an AR application, there are two potential approaches that a user can take. The first is full immersion with the real-world passthrough view to navigate environments. However, this is cumbersome and could lead to accidents, as users may not see obstacles which appear in their way. A more realistic scenario is a user using the AR application and paying attention to the real world. This way, they are more aware of their surroundings, can find a safe and socially distanced path, and do not need to fixate their gaze on their devices. Instead, users can use the AR application to inform themselves which areas are safe and which to avoid. The application would be especially invaluable for places with masses of people, and navigating it quickly and safely would be difficult, for example, busy shopping centers and public transport hubs.

## System Architecture

Considering the potential of utilizing AR and smart sensing for social distancing, we develop a prototype, DistAR, which is an AR system that utilizes data of users' surroundings from on-device and external sensors to infer and calculate distances for users to decide on how to social distance in real time. If achieving real-time analyses is not feasible, users may break





**FIGURE 2.** Architectural design of the system containing the components for AR environment and smart sensing analysis.

their social distancing bubble, inadvertently breathing in virus particles.

Figure 2 provides an architecture overview, as well as the key components of the system. The DistAR system consists of three main components, i.e., the (user) smartphone, an external server, and sensors deployed on the smart campus. The application obtains camera and GPS sensor data from the smartphone device. An on-device distance estimator uses two methods of estimating distances from the user. Then, with a position localizer, distances and position are input into a heatmap generator, which renders both the distances calculated on-device and a crowdedness geographical heatmap. This latter geographical map is created on an external server that acts as a central aggregation point for IoT sensor data collected from the smart campus. The sensors communicate and transmit their data to the local cloud server using LoRaWAN. The server performs analysis, and the user application can retrieve the analysis from that server. In summary, the distance estimation of the user's surroundings is entirely contained on-device, and data from external sensors and its subsequent analysis provide supplemental environment crowdedness information.

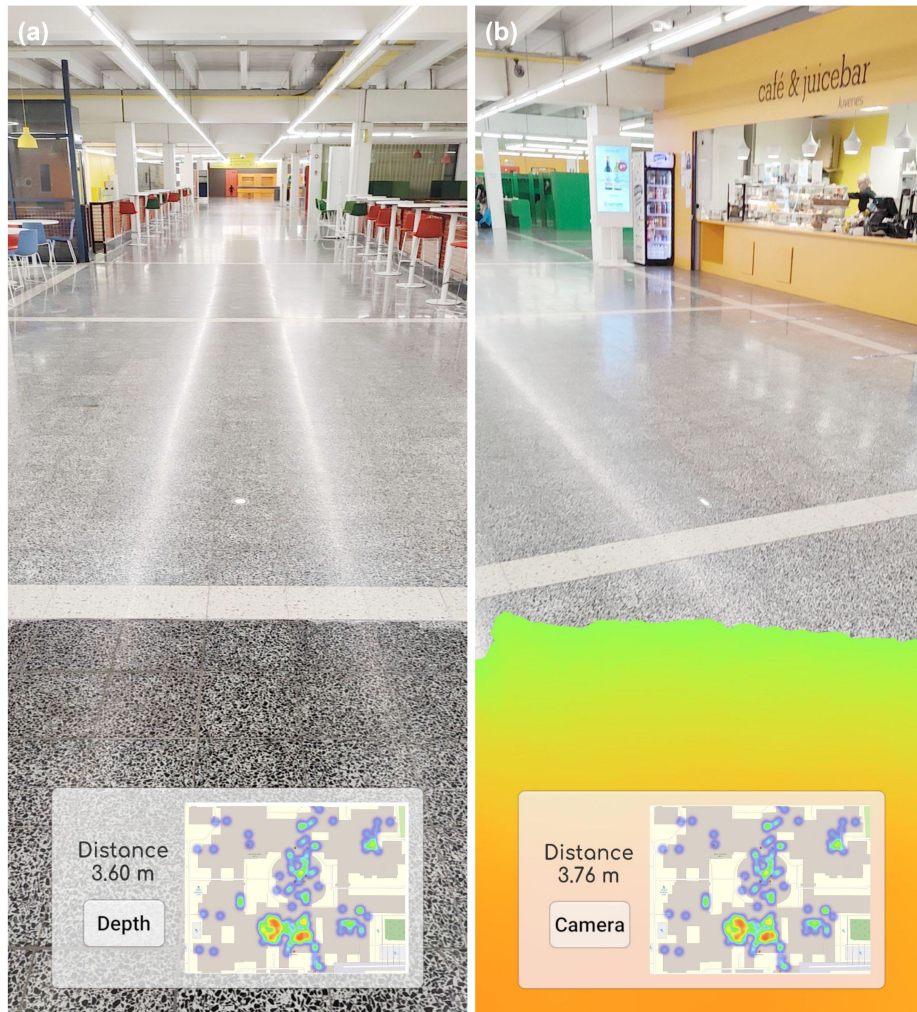
## Implementation

DistAR is a native android-based application that measures and visualizes distances between a user and their surroundings, and contains an interactive geographical map that allows users to view the general occupancy or busyness of a university campus. We develop an android-based application first because smartphones' ubiquity and prevalence enable quick and easy access to the application. Second, modern smartphone devices

have powerful resources enabling real-time on-device computer vision analyses. Finally, enabling the on-device estimation of the distances around the user requires using the depth API from ARCore, which is currently only supported on android.

Figure 3 presents the main views of DistAR<sup>a</sup>. We intend for the application interface to be as simple as possible to be understood by broad groups of people, irrespective of language or technological capability. The initial view [see Figure 3(a)] when a user loads the application is of the camera passthrough from the primary world-facing device camera. We choose a safety distance of 2 m as recommended by several governments during the initial pandemic onset.<sup>14</sup> Regions less than 2 m from the user are "clear" in view, and regions larger than 2 m have a slight opacity applied to the view. We use transparency to differentiate between distances, allowing users to view obstacles that may appear in front of them instead of completely obscuring their view. In addition, there is a central reticle where depth is estimated, and the application displays this distance in an information box at the bottom of the screen. Whenever the reticle moves to a region larger than 2 m, the application activates the smartphone's haptic motors to alert the user that those areas would break social distancing. We chose this reticle to move along flat surfaces as this gives users a better understanding of distances rather than a static and fixed circle. We also implement this feature because while people can supposedly keep a safe distance from others, there are times when people are

<sup>a</sup>A video demonstration can be found here: <https://www.youtube.com/watch?v=-VkhjL5WMDY>



**FIGURE 3.** Example views of the application while on the main thoroughfare of a university campus. (a) The application’s camera pass-through mode with the distance returned from the central reticle. (b) The view when regions within 2 m are highlighted to the user as a heatmap-style visualization. Both views contain access to the crowded geographical heatmap map from smart sensing data.

unaware or forget the required distances. Our application would, therefore, remind those users and support their social distancing.

To help users identify and recognize correct distances, a “depth” mode is available once the button is pressed in the application view. This mode replaces the opaque regions that are less than 2 m with a colored heatmap-type visualization [see Figure 3(b)]. Regions closer to the user are red, which then changes into a spectrum of green and blue when distances are further away from the user. We provide this function as if users are using the application and performing other tasks; they may not realize breaches to their social distancing limit. By using color and the previous haptic feedback, we can better alert users visually, clearly, and directly.

These features allow users to make informed decisions on where to walk to avoid areas that potentially contain dangerous particles. The described depth estimation uses ARCore’s depth APIs to measure and visualize distances from a user’s smartphone device.<sup>15</sup> A method called *localized depth* samples depth values stored on the CPU, and another method, *depth sense*, processes the entire GPU-stored screen data to estimate depth and distances. Both methods perform their processing entirely on-device, negating the need to offload images or video streams to an external server for analysis. Users can view the results of this analysis in near-real time, allowing them to make prompt and informed decisions on where to move.

While users can understand their surroundings through AR-based analysis, smart sensing and analytics are crucial for detecting areas beyond smartphone viewing range. Within our system, we employ a sensing system and smart campus data from the University of Oulu.<sup>16</sup> The current deployment contains 410 active fixed-in-location sensors that collect various measurements, including CO<sub>2</sub>, light levels, noise, etc. The real-time sensor data are sent to a storage server over LoRaWAN using the university 5G test network. The centralized data are accessed in real time using a RESTful API. We use this API to retrieve the sensor data, preprocess to remove missing values, and select the subset of measurements that indicate crowdedness within an environment (i.e., PIR). Based on these data, we generate a geographical crowdedness heatmap of the entire university campus. The map is produced on the same cloud server and loaded into the AR application's information box.

## Evaluation

We look toward the design considerations and the three main requirements we defined to evaluate our system. We evaluate using a OnePlus 9 Pro smartphone for the user client and a PC (Intel Core i7-9700 K CPU, 32-GB RAM, and an NVIDIA GeForce RTX 2080 with Max-Q Design GPU) as a local private cloud.

The first evaluation pertains to *smart sensing and AR with high accuracy*. We estimate crowdedness using the PIR sensor measurements. These data are part of the continuous data streams from each campus-deployed sensor. The implementation of the smart environment contains several methods to ensure that the PIR values and the other data retrieved are accurate. As the broker collects the sensor measurements and the centralized server stores the sensor data, a custom drift detection algorithm is applied to detect erroneous values. If the data are drifting, the system attempts to correct the values. However, if the deviancy is too large, this is then flagged to a system operator, who proceeds to debug the sensor on-location manually. In this way, we can ensure that the subsequent analysis for estimating crowdedness is accurate and the values reflect the actual nature of the environment.

The second evaluation relates to *near real-time analysis and rendering*. We look at the end-to-end latency and separate the latency for different system components. We group these individual components into client and server. The client analyzes the environment in real time and on-device to generate the heatmap-style visualization and to calculate the central reticle distance. We obtain the application's latency in performing these calculations by timing the call of the

depth API method after receiving a new frame until the on-screen visualization rendering completes. Using the *depth sense* method to generate the heatmap-style visualization takes an average time of 0.37 ms while using *localized depth* takes a time of 0.05 ms. Both latency values are small, highlighting the real-time analysis and rendering of the depth visualizations of the environment. Comparatively, the server takes an average time of 3.90 s to generate the geographical heatmap of the smart university campus, which is a potentially significant value as the limit of a user's flow of thought is approximately 1 s.<sup>17</sup> However, client-server communication utilizes the university's 5G network, so the average latency between the smartphone and the local cloud averages 27.1 ms when using 5G. The large latency attributed to the generation of the geographical heatmap is because our current cloud scripts generate the map for the entire university campus. Reducing this total latency would be possible by shrinking the map region of concern and providing the current user coordinates to move and generate more valuable maps on-the-fly adaptively. In general, we find that the AR system component performs in near real time and with further improvements, so can the sensor-based analysis.

The last evaluation pertains to *ease of use*, a highly subjective metric to evaluate. We invited five users (age:  $\bar{x} = 25.4$  years,  $\mu = 4.83$  years) to use the application to navigate a busy thoroughfare at the University of Oulu. The participants included four male master's students in computer science and one female master's student in learning, education, and technology. After using our application, we asked the participants to complete a short usability survey and gathered their general opinions and feedback. Table 1 lists several questions from our survey and their corresponding average score. Users selected an integer value for each question from 1 to 5, where 1 relates to "strongly disagree" and 5 to "strongly agree." We see from questions 1) through 3) that users found the system easy to use, and when given to other nontechnical users, the application would be easy to pick up and use. One participant commented how the application is "straightforward." However, from questions 4) and 5), the participants indicate that the application's UI could be further refined and developed. For example, after being shown the vision in Figure 1, several participants wished there could be a more similar interface with numerical values of crowdedness. Finally, from question 6), the participants did not see the significance of social distancing, attributed to their young-minded mentality of the COVID-19 pandemic being over. However, they did note that DistAR would be helpful for those who may still want and medically need to

**TABLE 1.** Several example questions and statistics from our usability survey after five users tried our AR social distancing application.

Question	Average	Std. Dev.
1) I thought the system was easy to use.	3.80	1.10
2) I think that I would need the support of a technical person to be able to use this system.	2.00	1.22
3) I imagine most people would learn to use this system very quickly.	4.00	0.71
4) I liked the graphics and images of DistAR.	3.20	1.30
5) The layout of DistAR was visually pleasing.	2.60	1.14
6) I would like to use this system frequently.	2.80	1.48

socially distance. Furthermore, they suggested that the application vision would help them to evaluate whether to enter areas which may not seem busy from the outside. This small study has shown us which potential parts of our system need further improvement, and we recognize the need to further explore with a larger number of participants with diverse backgrounds.

## DISCUSSION

This work enables social distancing in various environments by leveraging AR and smart sensing. Using both is important when considering current or future pandemic scenarios where maintaining proper distances between people is necessary to ensure the general public's health and the proper containment of viruses. By utilizing ubiquitously available devices, such as smartphones, and considering the increasing prevalence of public and private smart environments and infrastructures, applications to support similar scenarios can be quickly deployed and accessed by all stakeholders in society.

However, user adoption of social distancing applications could be further improved by considering web-based AR applications. These browser experiences use the WebXR device API, an API that is undergoing standardization efforts to bring the capabilities of traditional installation-based AR and virtual reality applications to web browsers.<sup>18</sup> Several benefits arise from using WebXR, for example, accessing applications without needing to install from a traditional "app" store. Other benefits include pushing updates on the fly to all users and creating compatible applications across different devices (albeit with the caveat of only supporting

WebXR API-supported devices). Replicating our application using WebXR is only possible on android as the WebXR depth sense API is yet to be formally standardized and is still undergoing active development. This API would allow access to the same depth information as in the native-deployed ARCore depth API. Nonetheless, the WebXR working group has published one initial version of the depth sense API and is currently being tested for a future release for Chrome on android.<sup>19</sup> Further iterations of DistAR could utilize this API to provide better compatibility with a more significant number of devices. In addition, further considerations should also be made of the user hardware itself. For example, our implementation uses an android OnePlus 9 Pro smartphone, which features a powerful device chip for processing capabilities, allowing our application to perform analyses on both the CPU and GPU. Another consideration for broader adoption is whether the same performance is achievable on other, less powerful devices.

We have developed DistAR, a proof-of-concept social distancing application. To understand the application's usability, we will conduct additional extensive user studies to obtain more subjective opinions of how the application performs and is perceived by a broader group, including more varied ages and technological capabilities. In addition, one essential requirement of the application is to provide insights in near real time. Therefore, the application should be low latency, which requires an extensive evaluation of the computational throughput on the cloud and measuring the latency of insights arriving at the user application. Also, DistAR is capable of performing localization of the end user. However, one key consideration is that the application should be location agnostic (i.e., usable in indoor and outdoor environments). Considering indoors, localization, and positioning techniques within these areas still require improvement to be truly precise and accurate. For this purpose, we will utilize Bluetooth access points installed on the smart campus to enhance indoor positing when using the social distancing application. With improved positioning, we could refine the AR visualizations to be more exact, e.g., overlaying visualizations on people and the entire environment. In addition to better positioning, the application and system would require an object detection component to recognize people and where they are in the environment.

We enrich our AR application with the real-time crowdedness context retrieved from sensors. However, other considerations could be possible. First, we solely used the PIR metric for representing crowdedness. Instead of just these data, the results could be further reinforced with other calculated metrics, such as CO<sub>2</sub> concentrations and noise levels, to estimate crowdedness. We are also exploring the best ways to



render this information, e.g., showing a full heatmap or concentrating on the region where the user is. Data rendering is essential as the information should be as easy to consume as possible. Another aspect to consider is how we alert users to breaches of their social distancing limits. The current iteration of our application uses the deployment device's haptic motor to alert users, although, if a user is wearing gloves or distracted in other matters, there is a possibility that they do not perceive the vibrations. One way to overcome this would be to provide haptic notifications on a separate device, such as a wearable armband. Users would then be more aware if there are incursions on their social distancing bubble and adjust accordingly. While this would require an additional cost to users, this inclusion would greatly benefit those who are visually impaired or have other disabilities.

## Future Directions

### *Cross-Platform AR Applications*

Our original native-device application is developed and deployed for android smartphones. Large-scale adoption of social distancing applications by the general public requires the application to be available for as many devices as possible, such as Apple's iPhone. This cross-platform access would be significantly easier with development using the aforementioned WebXR API. However, there is still a lack of general support for the API on iOS.

In addition, AR is not limited to smartphones; HMDs are important devices for immersive AR experiences, for example, Microsoft's HoloLens 2. Using HMD-based applications would allow users to free up their hands instead of holding up a smartphone to analyze their surroundings constantly. In addition, users could also be constantly aware of the distances to other people around them. One example could be a person in a supermarket purchasing groceries. While with a smartphone-based AR social distancing application, navigating a store around people would be a cumbersome experience if the user is pushing a trolley, for example. The user would constantly need to put down their phone to either push their trolley or pick up items. An AR social distancing application for HMDs would certainly alleviate this issue. However, one limitation to achieving this is the support and performance of distance estimation algorithms. For example, the depth API utilized in our application to estimate depth and distances is not usable on HoloLens 2 as ARCore is not natively supported. Instead, other algorithms or computation offloading of device data is required to achieve the same experience but potentially to the detriment of user

experience. There are other limitations to HMD-based applications, such as the current costs of HMDs remaining relatively high and their functionality outdoors is not a stable experience yet. However, as there are more generations of HMDs, we expect their cost to decrease and their outdoor support to improve to allow users to use HMDs in all environments.

In a related fashion, other nonpersonal AR devices could potentially be social distancing tools, such as interactive public displays.<sup>20</sup> However, in practical deployments, these AR displays would be more suitable for near-stationary scenarios, such as queuing in public places. For example, suppose a person is walking to a destination at pace. In that case, there is a possibility that they do not view the social distancing information at all as the visualization may be cumbersome to understand and challenging to plot when relating to their current surrounding environment. AR would be more suitable as personalized information would be available.

### *QoE and User Acceptance*

Ensuring a good user experience is essential for any AR application, regardless of the use case. Users expect augmentations or holograms to be rendered directly on relevant objects promptly, ensuring that the information matches the environment before becoming too stale and irrelevant. Several approaches can potentially improve the user QoE.

First, the device resource usage must be monitored and adaptive when needs change. AR applications rely heavily on the user screen, the device CPU and GPU, and onboard sensors, such as the camera, GPS, and Bluetooth. These are expenses on the already finite energy resource of device battery life. Left unchecked, users might find their devices very quickly depleted, very hot from extensive energy usage, and their phone's overall performance throttled on an OS level to conserve the remaining battery, all of which would lead to a worsened user experience. Therefore, the application should optimize between software features and hardware utilization to provide the best tradeoff for users.

Other external factors could also be considered to improve the overall user-perceived QoE. As part of our system, DistAR requires accessing an external server to aggregate and analyze sensor data for the beyond-user awareness of the crowdedness of locations. Naturally, this component could be improved to reduce the latency required to communicate and retrieve the sensor analytics results. First, an edge server is more beneficial for sensor data analysis than placing the calculations on a remote cloud server. Using an edge server would reduce the communication latency from user devices to the server and provide benefits, such as

improved security and privacy as the data are closer to end users. Second, leveraging large-scale sensor deployments for AR applications, like in a campus area or at an urban scale, poses challenges for real-time analysis and good user experiences, as overlaying real-time contextual information could suffer due to network delays and resource demands for large data rendering. For such scenarios, leveraging 5G networks with edge computing infrastructure could enable smoother experiences and provide much faster data transfer rates than 4G/LTE and WiFi communication links.

### ***Analysis of Smart Sensing Environments***

The current implementation of the smart campus sensing system performs drift detection to correct erroneous measurements, which may appear in the collected data. In this way, we ensure that the estimations of crowdedness are accurate to produce a viable social distancing system. However, predictive modeling is also needed where the estimations of crowdedness would allow users to plan whether to go through an area. By implementing this forecasting, the vision of Figure 1 would be better achievable as pathfinding algorithms could function based on historical data and predictive estimations.

### ***Synchronization of AR and Sensing Systems***

Context-aware augmented interactions require sensing capacities on large scales. AR and sensing systems are distinct network applications with specific constraints and deadlines. Synchronizing these two technologies to provide updated information promptly presents a complex scheduling challenge, namely, ensuring that servers can succinctly analyze the data from sensors to provide the most relevant information to be rendered in AR before the information becomes stale and irrelevant. Similar to the abovementioned methods for improving QoE, 5G and edge computing technologies are two potential solutions for AR sensor synchronization because they will reduce the communication time between clients and servers. Therefore, the IoT data streams and AR rendering engines would be given additional time to ensure that both can synchronize succinctly.

Another consideration is an AR application's ability to traverse different environments and maintain the same functionality and experience. Massively deployed sensing is still an ongoing process, and not all environments contain the sensors required to supplement the AR-based distance calculations. However, once those sensors are available, the data would most likely be stored on a different server (i.e., on-premises). The question then arises of how the application can access those data. One potential solution would be to introduce a standardized system and API that the application can easily interface to, for

example, initially contacting the cloud to discover which local edge servers are available to communicate with, or using Bluetooth signals to determine the local edge servers. In this way, the application could be useful not just in public spaces, such as universities, but also within private companies and other similar spaces.

## **CONCLUSION**

The COVID-19 pandemic has impacted our daily lives and societies, and adapting to the postpandemic new normal is of the utmost importance. As part of this, specific measures should be taken by those more vulnerable and susceptible to the virus, for example, conventional prevention measures, such as wearing face masks or social distancing. However, constantly adhering to these measures may not always be easy. Instead, users can use technology to support their maintenance of these measures. Maintaining social distancing is one measure that can be supported using augmented reality and smart sensing technologies. In this work, we have demonstrated one of the first efforts at combining both technologies for social distancing in our android application, DistAR. We utilized on-device depth estimation and externally analyzed sensor data for crowdedness to allow users to make informed decisions on social distancing.

## **ACKNOWLEDGMENTS**

This research work was supported by the Academy of Finland under Projects 323630, 319670, 325774, 345008, and 326305. This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

## **REFERENCES**

1. D. Chatzopoulos et al., "Hyperion: A wearable augmented reality system for text extraction and manipulation in the air," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 284–295.
2. Google, "Sodar," 2020. [Online]. Available: <https://sodar.withgoogle.com/>
3. L. Paule, "Augmented reality for social distancing," 2020. [Online]. Available: <https://blog.laval-virtual.com/en/augmented-reality-for-social-distancing/>
4. M. Liang et al., "Efficacy of face mask in preventing respiratory virus transmission: A systematic review and meta-analysis," *Travel Med. Infect. Dis.*, vol. 36, 2020, Art. no. 101751.
5. C. T. Nguyen et al., "A comprehensive survey of enabling and emerging technologies for social distancing—Part I: Fundamentals and enabling technologies," *IEEE Access*, vol. 8, pp. 153479–153507, 2020.

6. C. T. Nguyen et al., "A comprehensive survey of enabling and emerging technologies for social distancing—Part II: Emerging technologies and open issues," *IEEE Access*, vol. 8, pp. 154209–154236, 2020.
7. Y. Himeur et al., "Deep visual social distancing monitoring to combat COVID-19: A comprehensive survey," *Sustain. Cities Soc.*, vol. 85, 2022, Art. no. 104064. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210670722003821>
8. M. Rezaei and M. Azarmi, "DeepSOCIAL: Social distancing monitoring and infection risk assessment in COVID-19 pandemic," *Appl. Sci.*, vol. 10, no. 21, 2020, Art. no. 7514. [Online]. Available: <https://www.mdpi.com/2076-3417/10/21/7514>
9. J. Shah, M. Chandaliya, H. Bhuta, and P. Kanani, "Social distancing detection using computer vision," in *Proc. 5th Int. Conf. Comput. Methodol. Commun.*, 2021, pp. 1359–1365.
10. S. Tanwar, R. Gupta, M. M. Patel, A. Shukla, G. Sharma, and I. E. Davidson, "Blockchain and ai-empowered social distancing scheme to combat covid-19 situations," *IEEE Access*, vol. 9, pp. 129830–129840, 2021.
11. G. Campos, "Sensors and projection enforce social distancing," 2020. [Online]. Available: <https://www.avinteractive.com/news/covid-19/queuesight-ar-social-distancing-tool-maps-space-around-queue-04-08-2020/>
12. N. H. Motlagh et al., "Monitoring social distancing in smart spaces using infrastructure-based sensors," in *Proc. IEEE 7th World Forum Internet Things*, 2021, pp. 124–129.
13. F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quart.*, vol. 13, no. 3, pp. 319–340, 1989.
14. "How to protect yourself and others," Centers Dis. Control Prevention, Tech. Rep., Oct. 2022. [Online]. Available: <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html>
15. R. Du et al., "DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality," in *Proc. 33rd Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 829–843.
16. E. Gilman et al., "Internet of things for smart spaces: A university campus case study," *Sensors*, vol. 20, no. 13, 2020, Art. no. 3716. [Online]. Available: <https://www.mdpi.com/1424-8220/20/13/3716>
17. J. Nielson, "Response times: The 3 important limits," Nielson Norman Group, Jan. 1993. [Online]. Available: <https://www.nngroup.com/articles/response-times-3-important-limits/>
18. B. Jones, M. Goregaokar, and R. Cabanier, "WebXR device API," *World Wide Web Consortium*, 2021. [Online]. Available: <https://www.w3.org/TR/webxr/>
19. Google, "WebXR depth API - chrome platform status," Apr. 2022. [Online]. Available: <https://chromestatus.com/feature/5742647199137792>

20. F. Alt, S. Schneegaß, A. Schmidt, J. Müller, and N. Memarovic, "How to evaluate public displays," in *Proc. Int. Symp. Pervasive Displays*, 2012, pp. 1–6, doi: [10.1145/2307798.2307815](https://doi.org/10.1145/2307798.2307815).

**JACKY CAO** is a doctoral researcher at the University of Oulu, 90570, Oulu, Finland. His research interests include mobile augmented reality, edge computing, and 5G networks. Cao received his M.Phys. degree from the University of Durham, Durham, U.K. He is a graduate student member of the IEEE. He is the corresponding author of this article. Contact him at [jacky.cao@oulu.fi](mailto:jacky.cao@oulu.fi).

**HASSAN MEHMOOD** is a doctoral researcher at the University of Oulu, 90570, Oulu, Finland. His research interests include distributed computing, data management and integration, machine learning, big data analytics, edge computing, and concept drift detection and adaptation in real-time data streams. Contact him at [hassan.mehmood@oulu.fi](mailto:hassan.mehmood@oulu.fi).

**XIAOLI LIU** is a postdoctoral researcher in computer science at the University of Helsinki, 00100, Helsinki, Finland. Her research interests include big data analysis, AI, the Internet of Things, and wearable computing. Liu received her Ph.D. degree in mathematics and statistics from the University of Helsinki. Contact her at [xiaoli.liu@helsinki.fi](mailto:xiaoli.liu@helsinki.fi).

**SASU TARKOMA** is a full professor at the Department of Computer Science, University of Helsinki, 00100, Helsinki, Finland. His research interests include mobile computing, Internet technologies, and AI. Tarkkoma received his Ph.D. degree in computer science from the University of Helsinki. He is a senior member of the IEEE. Contact him at [sasu.tarkoma@helsinki.fi](mailto:sasu.tarkoma@helsinki.fi).

**EKATERINA GILMAN** is a postdoctoral researcher supported by the Academy of Finland at the Center for Ubiquitous Computing, University of Oulu, 90570, Oulu, Finland. Her research interests include data analytics, context modeling and reasoning, and machine learning in ubiquitous computing, IoT, and data-intensive systems. She is a member of the IEEE. Contact her at [ekaterina.gilman@oulu.fi](mailto:ekaterina.gilman@oulu.fi).

**XIANG SU** is an associate professor at the Norwegian University of Science and Technology, 7034, Trondheim, Norway, and also with the University of Oulu, 90570 Oulu, Finland. His research interests include the Internet of Things, edge computing, and augmented reality. Su received his Ph.D. degree in technology from the University of Oulu, 90570, Oulu, Finland. Contact him at [xiang.su@ntnu.no](mailto:xiang.su@ntnu.no).