Fighting Pandemics with Augmented Reality and Smart Sensing-based Social Distancing

Cao, Jacky

2023-01

Cao , J , Mehmood , H , Liu , X , Tarkoma , S , Gilman , E & Su , X 2023 , ' Fighting Pandemics with Augmented Reality and Smart Sensing-based Social Distancing ' , IEEE Computer Graphics and Applications , vol. 43 , no. 1 , pp. 65-75 . https://doi.org/10.1109/MCG.2022.3229107

http://hdl.handle.net/10138/354772
https://doi.org/10.1109/MCG.2022.3229107

Fighting Pandemics with Augmented Reality and Smart Sensing-based Social Distancing

Jacky Cao
University of Oulu

Hassan Mehmood
University of Oulu

Xiaoli Liu
University of Helsinki

Sasu Tarkoma
University of Helsinki

Ekaterina Gilman
University of Oulu

Xiang Su
Norwegian University of Science and Technology and University of Oulu

***Abstract*—In a post-pandemic 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.**

Index Terms: augmented reality, Internet of Things, social distancing, smart campus

INTRODUCTION

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 specialised 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 semi-transparent displays, where augmentations are directly displayed in front of users' eyes [1]. 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 recognise objects and events within the vicinity of users to assist in their daily lives, for example, in city-wide 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 practising 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 [2] and ARAroundME [3], are developed to overlay social distancing guidelines in real-world environments.

We present `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 visualisations, which helps them reduce their potential exposure to airborne virus particles. In addition to depth estimations on user devices, the application renders analysed sensing data (passive infrared 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 as users may need to understand the number of people in a wider environment and not just within their local line of sight. We deploy 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 [4]. 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 [5], [6], and there has been considerable interest in using AI algorithms to calculate distances between people from optical images. Himeur *et al.* [7] classify the tools in existing social distancing monitoring frameworks into "visual" and "non-visual". Visual-based social distancing includes analysis of media streams using hand-crafted feature-based, CNN-based, transfer learning-based, and 3D-based algorithms. Comparatively, non-visual methods include analysis of Bluetooth signals and passive WiFi sensing. Rezaei *et al.* [8] demonstrate 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.* [9] test 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.* [10] demonstrate 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 [3] and “Sodar” by Google [2]. 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 [11] 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 [3] and Sodar [2], 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 analysed to detect people and distances. Then, the devices can render this information to the user, e.g., as circles around each person. Secondly, these areas can be colour-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 be stored and analysed on a central server. Then, for visualising this information in AR, buildings or rooms could be overlaid with a similar colour 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 (CO2) concentrations highly correlate with aerosols containing COVID-19. Users could use data gathered by CO2 and passive infrared motion sensors to gain insights into potential virus transmission risks and the crowdedness of environments [12]. In this research, we have utilised passive infrared 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.

Figure 1. A vision of supporting social distancing during and after the COVID-19 global pandemic using AR and smart sensing technologies.

DistAR: AN AUGMENTED REALITY 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, 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 analysing 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. 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 envi-

ronments 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 visualisations. 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-analysed 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 [13]. 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 utilise a button to swap between the transparent and coloured opaque visualisations.

Further iterations of the application should focus on refining and improving the application to fulfil 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 pre-defined system conditions. Additionally, more memory-efficient code constructs could help minimise 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 centres and public transport hubs.

System architecture

Considering the potential of utilising AR and smart sensing for social distancing, we develop a prototype, `DistAR`, which is an AR system that utilises 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 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 localiser, 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

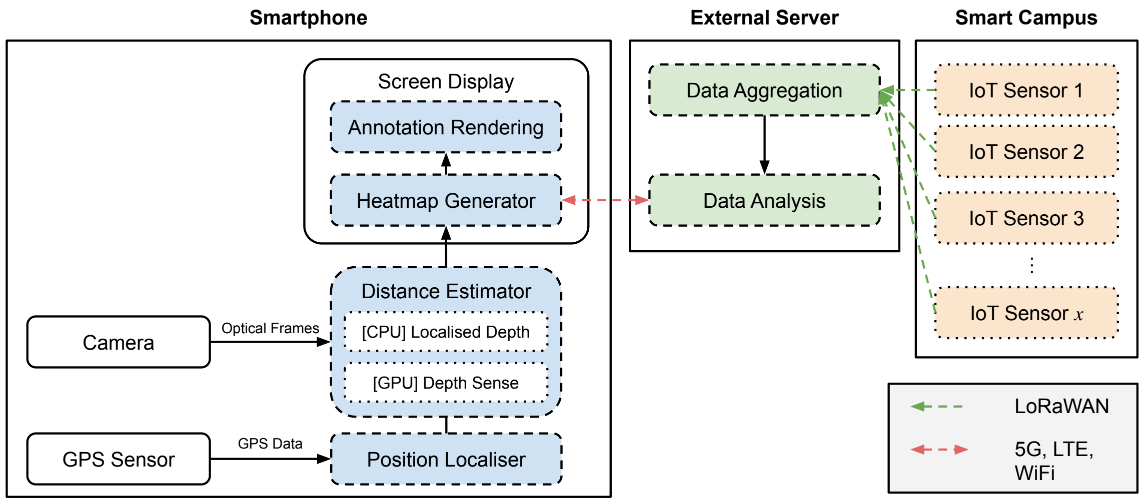

Figure 2. The architectural design of the system containing the components for AR environment and smart sensing analysis.

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 visualises 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 firstly because smartphones' ubiquity and prevalence enable quick and easy access to the application. Secondly, 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`[1]. 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 (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 [14]. 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. Additionally, 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 unaware or forget the required

[1]A video demonstration can be found here: https://www.youtube.com/watch?v=-VkhjL5WMDY

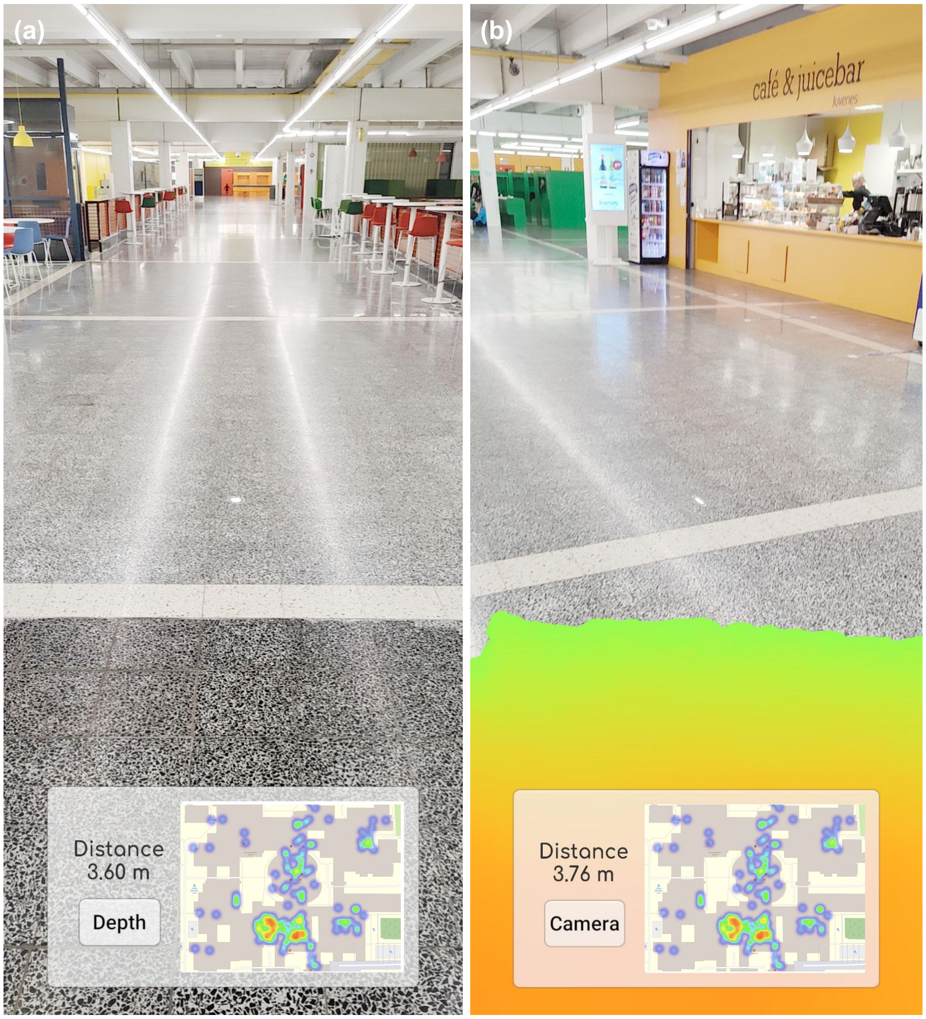

Figure 3. Example views of the application while on the main thoroughfare of a university campus: (a) the application's camera passthrough mode with the distance returned from the central reticle, and (b) the view when regions within 2 m are highlighted to the user as a heatmap-style visualisation. Both views contain access to the crowded geographical heatmap map from smart sensing data.

distances. Our application would therefore remind those users and support their social distancing.

To help users identify and recognise 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 coloured heatmap-type visualisation (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 realise breaches to their social distancing limit. By using colour 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 visualise distances from a user's smartphone device [15]. A method called *localised 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 [16]. The current deployment contains 410 active fixed-in-location sensors that collect various measurements, including CO2, light levels, noise, etc. The real-time sensor data is sent to a storage server over LoRaWAN using the university 5G Test Network. The centralised data is accessed in real-time using a RESTful API. We use this API to retrieve the sensor data, pre-process to remove missing values, and select the subset of measurements that indicate crowdedness within an environment, i.e., passive infrared (PIR). Based on this 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 towards 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 i7-9700K 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. This data is 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 centralised server stores the sensor data, a custom drift detection algorithm is applied to detect erroneous values. If the data is 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 analyses the environment in real-time and on-device to generate the heatmap-style visualisation 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 visualisation rendering completes. Using the *depth sense* method to generate the heatmap-style visualisation takes an average time of 0.37 ms while using *localised depth* takes a time of 0.05 ms. Both latency values are small, highlighting the real-time analysis and rendering of the depth visualisations 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 [17]. However, client-server communication utilises 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 invite 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

Table 1. Several example questions and statistics from our usability survey after 5 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

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 non-technical 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 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 recognise 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 utilising 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 standardisation efforts to bring the capabilities of traditional installation-based AR and virtual reality applications to web browsers [18]. 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 standardised 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 [19]. Further iterations of `DistAR`could utilise 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 localisation 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, localisation and positioning techniques within these areas still require improvement to be truly precise and accurate. For this purpose, we will utilise 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 visualisations to be more exact, e.g., overlaying visualisations on people and the entire environment. In addition to better positioning, the application and system would require an object detection component to recognise 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. Firstly, we solely used the PIR metric for representing crowdedness. Instead of just this data, the results could be further reinforced with other calculated metrics, such as CO2 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 of 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.

Additionally, AR is not limited to smartphones; head-mounted displays (HMD) 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 analyse 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 utilised 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 non-personal AR devices could potentially be social distancing tools, such as interactive public displays [20]. 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 visualisation may be cumbersome to understand and challenging to plot when relating to their current surrounding environment. AR would be more suitable as personalised 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 quality of experience (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 optimise between software features and hardware utilisation to provide the best trade-off 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 analyse 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. Firstly, 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 is closer to end users. Secondly, 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 modelling is also needed where 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.

Synchronisation 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. Synchronising these two technologies to provide updated information promptly presents a complex scheduling challenge. Namely, ensuring that servers can succinctly analyse 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 above-described methods for improving QoE, 5G and edge computing technologies are two potential solutions for AR sensor synchronisation 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 synchronise 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 that data. One potential solution would be to introduce a standardised 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 to, 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.

CONCLUSIONS

The COVID-19 pandemic has impacted our daily lives and societies, and adapting to the post-pandemic 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 utilised on-device depth estimation and externally analysed sensor data for crowdedness to allow users to make informed decisions on social distancing.

Acknowledgement

This research work was supported by the Academy of Finland projects 323630, 319670, 325774, 345008, and 326305.

REFERENCES

1. D. Chatzopoulos, C. Bermejo, Z. Huang, A. Butabayeva, R. Zheng, M. Golkarifard, and P. Hui, "Hyperion: A wearable augmented reality system for text extraction and manipulation in the air," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 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, L. Gao, C. Cheng, Q. Zhou, J. P. Uy, K. Heiner, and C. Sun, "Efficacy of face mask in preventing respiratory virus transmission: a systematic review and meta-analysis," *Travel Medicine and Infectious Disease*, p. 101751, 2020.
5. C. T. Nguyen, Y. M. Saputra, N. V. Huynh, N.-T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz, S. Chatzinotas, and B. Ottersten, "A comprehensive survey of enabling and emerging technologies for social distancing—part i: Fundamentals and enabling technologies," *IEEE Access*, vol. 8, pp. 153 479–153 507, 2020.
6. C. T. Nguyen, Y. M. Saputra, N. Van Huynh, N.-T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz, S. Chatzinotas, and B. Ottersten, "A comprehensive survey of enabling and emerging technologies for social distancing—part ii: Emerging technologies and open issues," *IEEE Access*, vol. 8, pp. 154 209–154 236, 2020.
7. Y. Himeur, S. Al-Maadeed, N. Almaadeed, K. Abualsaud, A. Mohamed, T. Khattab, and O. Elharrouss, "Deep visual social distancing monitoring to combat covid-19: A comprehensive survey," *Sustainable Cities and Society*, vol. 85, p. 104064, 2022. [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," *Applied Sciences*, vol. 10, no. 21, 2020. [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 *2021*

5th International Conference on Computing Methodologies and Communication (ICCMC), 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. 129 830–129 840, 2021.
11. G. Campos, "Sensors and projection enforce social distancing." [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, P. Toivonen, M. A. Zaidan, E. Lagerspetz, E. Peltonen, E. Gilman, P. Nurmi, and S. Tarkoma, "Monitoring social distancing in smart spaces using infrastructure-based sensors," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 124–129.
13. F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.
14. W. Y. B. F. Vaccinated, "How to protect yourself and others," *Centers for Disease Control and Prevention*, 2021.
15. R. Du, E. Turner, M. Dzitsiuk, L. Prasso, I. Duarte, J. Dourgarian, J. Afonso, J. Pascoal, J. Gladstone, N. Cruces, S. Izadi, A. Kowdle, K. Tsotsos, and D. Kim, "DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality," in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '20. ACM, 2020, pp. 829–843.
16. E. Gilman, S. Tamminen, R. Yasmin, E. Ristimella, E. Peltonen, M. Harju, L. Loven, J. Riekki, and S. Pirttikangas, "Internet of things for smart spaces: A university campus case study," *Sensors*, vol. 20, no. 13, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/13/3716
17. J. Nielson, "Response times: The 3 important limits," *Nielson Norman Group: Jan*, vol. 1, 1993.
18. J. Brandon, M. Goregaokar, and R. Cabanier, "WebXR Device API," 2021. [Online]. Available: https://www.w3.org/TR/webxr/
19. "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 *Proceedings of the 2012 International Symposium on Pervasive Displays*, ser. PerDis '12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2307798.2307815

Biographies

Jacky Cao is a doctoral researcher at the University of Oulu. He received his MPhys. Physics degree from the University of Durham in 2019. His research interests include mobile augmented reality, edge computing, and 5G networks. Contact him at jacky.cao@oulu.fi.

Hassan Mehmood is a doctoral researcher at the University of Oulu. His main research interests are in 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.

Xiaoli Liu is a postdoctoral researcher in computer science at the University of Helsinki. She received her PhD in mathematics and statistics at the University of Helsinki in 2017. Her research interests include big data analysis, AI, the Internet of Things, and wearable computing. Contact her at xiaoli.liu@helsinki.fi.

Sasu Tarkkoma is a Full Professor at the Department of Computer Science, University of Helsinki. He completed his PhD in Computer Science at the University of Helsinki in 2006. His research interests include mobile computing, Internet technologies, and AI. Contact him at 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, Finland. Her current research interests include data analytics, context modelling and reasoning, and machine learning in ubiquitous computing, IoT, and data-intensive systems. Contact her at ekaterina.gilman@oulu.fi.

Xiang Su is an Associate Professor at the Norwegian University of Science and Technology. He received his PhD in technology from the University of Oulu in 2016. Prof. Su has extensive expertise in the Internet of Things, edge computing, and augmented reality. Contact him at xiang.su@ntnu.no.