BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems

Matthew Corbett
Virginia Tech
Blacksburg, Virginia, USA
matthewc84@vt.edu

Brendan David-John
Virginia Tech
Blacksburg, Virginia, USA
bmdj@vt.edu

Jiacheng Shang
Montclair State University
Montclair, New Jersey, USA
shangj@montclair.edu

Y. Charlie Hu
Purdue University
West Lafayette, Indiana, USA
ychu@purdue.edu

Bo Ji
Virginia Tech
Blacksburg, Virginia, USA
boji@vt.edu

Abstract

Augmented Reality (AR) devices are set apart from other mobile devices by the immersive experience they offer. While the powerful suite of sensors on modern AR devices is necessary for enabling such an immersive experience, they can create unease in bystanders (i.e., those surrounding the device during its use) due to potential bystander data leaks, which is called the bystander privacy problem. In this paper, we propose BystandAR, the first practical system that can effectively protect bystander visual (camera and depth) data in real-time with only on-device processing. BystandAR builds on a key insight that the device user's eye gaze and voice are highly effective indicators for subject/bystander detection in interpersonal interaction, and leverages novel AR capabilities such as eye gaze tracking, wearer-focused microphone, and spatial awareness to achieve a usable frame rate without offloading sensitive information. Through a 16-participant user study, we show that BystandAR correctly identifies and protects 98.14% of bystanders while allowing access to 96.27% of subjects. We accomplish this with average frame rates of 52.6 frames per second without the need to offload unprotected bystander data to another device.

CCS Concepts

• **Security and privacy** → **Domain-specific security and privacy architectures**; **Privacy protections**; • **Human-centered computing** → **Mobile devices**.

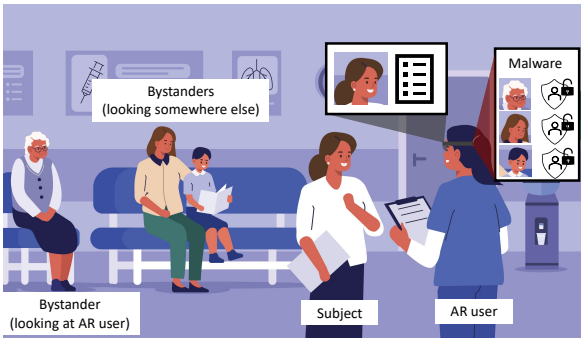


Figure 1: An illustration of the medical use case of AR, where a nurse wearing an AR device is interacting with a patient while there are bystanders present (watching or not watching the nurse). In this situation, while the patient's medical record information needs to be presented to the nurse via the AR device, bystander information must be protected.

1 Introduction

Augmented Reality (AR) devices are expected to reach an estimated 1.7 billion users by 2024, expanding from 1 billion in 2022 [4]. This is driven in part by industrial, healthcare, automotive, and military applications, with AR devices creating advances in mental health research, military decision-making, and assisting students with disabilities [59, 66]. These applications rely on the unique capabilities of AR devices, namely the ability to understand the physical world, and seamlessly blend the physical world and the holographic, digital world. This ability to create a virtual mapping of a physical space through Simultaneous Localization and Mapping (SLAM), establish synthetic holographic contact, and sense user eye gaze and hand gestures, is made possible by the integrated and powerful suite of sensors on modern AR devices. These sensors include Visible Light

ACM Reference Format:
Matthew Corbett, Brendan David-John, Jiacheng Shang, Y. Charlie Hu, and Bo Ji. 2023. BystandAR: Protecting Bystander Visual Data in Augmented Reality Systems . In *The 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys '23), June 18–22, 2023, Helsinki, Finland.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3581791.3596830

MobiSys '23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0110-8/23/06.
https://doi.org/10.1145/3581791.3596830

Cameras (VLCs), depth sensors, eye-tracking sensors, embedded microphones, accelerometers, and more [63].

Such sensors, while essential to the immersive experience that makes AR devices unique and powerful, do not discriminate in the data they collect. AR devices capture data required for well-intentioned tasks (e.g., SLAM, pose estimation, and gesture recognition), but also capture visual (e.g., camera and depth) data about bystanders (i.e., persons surrounding the device during its use), which can potentially be used to identify sensitive information (age, gender, emotion, gait, etc.) of bystanders for malicious purposes [7–9, 22, 28, 33]. This threat of bystander data leak is called the *bystander privacy problem* [14, 15, 52] (see definition in §2).

Example scenarios where the bystander privacy problem can arise include so-called *life-logging* or recording of visual evidence of everyday activities [20], assisting an Alzheimer's patient with memory care [25], and the AR functionality required to understand and interact with the physical world [56, 63]. Additionally, consider an AR-assisted medical service scenario, where an AR application is employed to assist nursing staff with triage at a hospital. The AR application may use facial recognition to present the nurse with a medical record chart for a patient, displayed in the AR device worn by the nurse. In a typical usage scenario (as shown in Fig. 1), a patient may be accompanied by a family member who sits or stands near her during a visit, and there could be other nearby patients as well. In this situation, the AR application may identify the faces of the bystanders (i.e., her family member and other patients) and unintentionally retrieve and present their medical records instead of, along with, that of the patient. This violates bystander privacy.

While there are very few studies addressing the bystander privacy problem specifically for AR devices [58, 65], there exists a large body of bystander privacy protection (BPP) systems for mobile and wearable devices in general. These systems can be divided into two categories based on whether the required input is *explicit* or *implicit*. *Explicit BPP systems* require the bystander or the user to possess an item, perform a gesture, or enroll in a system to expect privacy protection [1, 31, 35, 35, 53, 58, 64, 65, 67], which is not desired. In contrast, by using *implicit* information about the bystander (e.g., distance from the camera, the direction of eye gaze, emotion, and position in the frame) to determine if she is meant to be a part of the captured image. Also, *implicit BPP systems* seek to protect bystander privacy even when the bystander is unaware of the presence of the device [12, 26]. However, such systems can perform poorly when bystanders do not present themselves as expected in the captured image. In addition, existing implicit systems are implemented exclusively *off-device*, meaning that bystander data must be transferred to another device for processing. This opens an additional attack surface during the movement of this unprotected bystander data to a remote location [55].

Contributions. Due to the limitations of these existing BPP systems (which are designed for mobile and wearable devices in general, not specifically for AR devices), there is currently no practical BPP solution for head-worn AR devices. To that end, *we propose BystandAR, the first practical BPP system that can effectively differentiate a subject from a bystander and protect bystander visual (camera and depth) data in real-time with only on-device processing that does not negatively impact the immersive applications AR devices offer.* The main contributions of our work are as follows.

(1) From interesting studies in the field of psychology, we draw a useful insight that the device user's eye gaze and voice are highly effective indicators for subject/bystander determination in interpersonal interaction. Inspired by this key insight, we propose BystandAR, a novel bystander privacy protection system that uses the AR user's eye gaze and voice data to accurately determine the subject of interpersonal interaction, by leveraging novel AR capabilities such as eye gaze tracking and wearer-focused microphone array.

(2) Building such a system that exploits our key insight is highly nontrivial and entails two practical challenges: (C1) The device user's eye gaze may wander off the subject during the interaction; (C2) Performing the four basic tasks of bystander privacy protection (face detection, eye gaze tracking, subject identification via face-eye-gaze matching, and obscuration of bystanders) for every captured frame on-device is too costly and infeasible to keep up the frame rate. To address these challenges, we leverage the inherent functionality of AR devices, namely spatial awareness and eye gaze tracking, to locate faces in 3D and monitor the user's interactions with the detected face, hence removing the need to infer the location of faces from every captured frame and maintaining a usable frame rate with only *on-device* processing.

(3) Through an evaluation involving 16 participants, BystandAR was successful in protecting 98.14% of bystander faces through obscuration and in identifying the subject of an AR interaction in 96.27% of output frames. This ensures that the visual data of identified subjects remain available for legitimate uses. Our evaluation also shows an improvement in bystander protection by 12% over the most accurate existing solution and shows a marked increase in bystander perceptions of privacy. These improvements are gained while keeping bystander data *on-device*, removing the need to offload unprotected bystander data to another device, and maintaining frame rates as high as 52.6 frames per second (FPS).

2 Background and Motivation

In this section, we provide a brief background of the bystander privacy problem and explore the impact of modern AR devices.

2.1 The Bystander Privacy Problem

We first give some key definitions. An *AR user* (or simply *user*) is a person who wears an AR device; a *subject* is a person with whom the user intends to interact; a *bystander* is any non-user, non-subject third-party surrounding the device during its use. The *Bystander Privacy Problem* refers to when data (images, video, audio, etc.) that can be used to identify sensitive information (age, ethnicity, physical disability, emotion, etc.) is collected from bystanders who have not given consent to be part of the data collection [14, 15, 52]. Such bystander data leaks could also happen when well-meaning users, using AR applications for well-intentioned purposes, unintentionally violate bystander privacy.

2.2 Modern AR Devices Escalate the Bystander Privacy Problem

Surveys of bystander attitudes toward the presence of digital communication devices [51] were conducted as early as the year 2000, which shows negative perceptions of cellphone use (then novel) in public places. The survey participants reported that strangers

felt "intruded upon" when these devices breach the physical/digital barrier in their presence.

The rise of modern mobile devices, such as AR devices, opens the door to the development and growing adoption of novel immersive applications such as AR. These applications perform continuous sampling and recording of the physical world as part of their inherent operations, and in doing so, have propelled the bystander privacy problem into prominence. For example, a recent study focused on bystander perception of AR devices shows that bystanders are concerned with data collected by AR devices being used to uniquely identify them, and would prefer a way to require permission for the data to be recorded [17, 32]. Even medically assistive devices, including those designed to help persons with visual impairments, have been shown to elicit negative bystander feedback [3]. On the other hand, more recent work shows that bystanders are more willing to allow visual data to be captured if a blurring filter is added during capture; an additional 17.5% stated that they were willing over the un-blurred baseline [18]. Hence, there is a pressing need to develop a solution that can effectively protect bystander privacy without sacrificing the immersive experience AR devices offer.

3 Related Work

To the best of our knowledge, *there is no practical deployment of a bystander privacy protection (BPP) system for head-worn AR devices.* Hence, we discuss existing works on tackling the bystander privacy problem in mobile and wearable devices in general, including works that consider social media photo-sharing and compare their advantages and shortcomings.

There is a large body of existing BPP systems for mobile and wearable devices in general. These systems can be divided into two categories based on whether the input the system requires is *explicit* or *implicit*. *Explicit BPP systems* require the device user or the bystander to interact with the system, through a published privacy policy, hand gestures, etc. In contrast, *implicit BPP systems* use natural actions that occur with human interaction with the system (e.g., eye gaze, voice communication, and physical distance) to differentiate between the bystander and the subject.

3.1 Explicit BPP Systems

Explicit BPP systems require either the bystander, the device user, or both to perform some explicit actions to ensure bystander privacy. Works such as [1, 35] require potential bystanders to upload photos of their faces to train a facial classifier or respond to a prompt on their mobile device after a nearby photo capture to achieve some measure of privacy. Systems such as [29, 64] require bystanders to have a pre-defined privacy policy and facial signature on a linked server and potentially require a device user to audit and validate the privacy filters applied to scenarios that the system deems sensitive. Solutions such as [27, 35, 53, 58, 65, 67] are software frameworks and GUIs, requiring special equipment to be worn by bystanders, or by the device user in the presence of bystanders. Methods such as [31, 57, 64] require explicit hand gestures or markings from the device user or bystanders to express their privacy preferences or to exclude certain objects from an image, again requiring a user/bystander to make a conscious decision to be included or not.

In general, by giving the bystander or the user control of the situation, these explicit BPP systems potentially provide the kind of assurances that allow AR devices to be more acceptable in public places [18]. However, requiring the user/bystander to perform explicit actions such as hand gestures or to wear special physical devices imposes a significant burden on the user/bystander, resulting in such required actions being overridden, ignored, or unused if the user/bystander is not paying enough attention. Furthermore, such systems often require the device to be connected to a server to transport raw data, which increases the exposure of the data to potential misuse that the system is designed to prevent [55].

3.2 Implicit BPP Systems

Implicit BPP systems seek to protect bystander privacy without requiring any explicit actions to be taken by the user or the bystander. Such systems generally use a machine learning model (e.g., a neural network) to detect bystanders in the images captured by the device's camera [11–13, 26]. Such inference models extract various types of information from the images, such as distance from the camera, the direction of eye gaze, emotion, and position in the frame, and use them to improve detection accuracy.

By omitting the need for explicit actions taken by the user or bystander, implicit BPP systems can be easily deployed even when the bystander is unaware of the potential for their information to be recorded and hence promise much wider adoption. However, such systems can potentially suffer poor accuracy in bystander detection when bystanders present themselves in unexpected ways in the captured images. For example, two key features extracted and used in the bystander detection model in such systems are the eye gaze direction of a person in the captured image and being closer to the center of the frame than other persons; if a person's eye gaze is toward the device user and/or near the center of the frame, it is likely to be interacting with the device user and hence unlikely to be a bystander. However, as shown in the doctor visit scenario in Fig. 1, a bystander in the background could be looking at or moving toward the nurse (i.e., the device user). In this case, existing BPP solutions utilizing the two aforementioned features would erroneously label the bystander as a subject.

4 Key Insights and Challenges

In this section, we introduce our key insight that the AR user's eye gaze along with her voice can be much better indicators in differentiating the subject and the bystander and discuss the main challenges in designing a bystander privacy-preserving system that exploits this key insight.

4.1 Key Insights

A seminal work from the field of psychology shows that in interpersonal communication, a participant is expected to look at her partner more than 60% of the time, with a higher rate of 73% while the participant is listening to her partner speak [5]. Other works report this rate to be as high as 88% during a conversation [70]. Interestingly, while speaking, the speaker's eye contact can drop to as low as 41% of the total conversation time. Conversely, for eye contact with strangers or persons with whom one does not intend to speak (who are typically bystanders in an interpersonal

interaction), other works report an upper bound of 3.3 seconds before the eye contact becomes undesirable to the recipient [6].

From these psychological studies, we draw the following key insights. First, we can infer that *in an interpersonal interaction using immersive devices such as head-worn AR devices, eye contact of the device user with others can be a highly effective indicator or feature in distinguishing the subject from a bystander,* because psychologically, the user is likely to make frequent eye contact with the subject, but unlikely to stare at strangers (typically bystanders) for extended periods of time. Second, while the device user's eye gaze may vary across different cultures (e.g., it may occasionally wander off the listener in certain cultures) [2], it is more likely to be directed at the listener while speaking. This suggests that *combining the information of the device user's eye gaze and voice can be a highly effective indicator of whether the person is the intended subject of the interaction or just a bystander.*

4.2 Design Challenges

A straightforward design of a bystander privacy protection (BPP) system that exploits the above key insights would simply perform the following four basic tasks for *every camera-captured frame*, to identify the subject and bystanders in the frame: (1) *Face detection*: identify all of the faces in the frame, e.g., using a state-of-the-art face detection model; (2) *Eye gaze tracking*: track the eye gaze of the device user during the frame interval; (3) *Subject identification via face-eye-gaze matching:* identify the face in the frame that the user's eye gaze intersects with, and label the face as the subject of the current interaction, and the remaining faces as bystanders; (4) *Obscuration of bystanders:* obscure the faces of the bystanders in the frame using masking or other techniques (blurring, avatar, silhouette, pixelating, etc.) and export the frame (e.g., recording it or sending it to a third-party application). However, such a straightforward design would not work well due to the following two practical challenges.

Challenge 1: The device user's eye gaze may wander off the subject during the interaction. As shown in numerous psychological studies discussed in §4.1, while the device user's eye gaze tends to remain on the subject's face during a personal interaction, it is not 100% of the time. This happens for two possible reasons. *First*, the user's eye gaze occasionally wandering away from the subject's face is a normal part of human conversations, expected as a way to signal the natural transitions in the conversation [16]. *Second*, the device user may need to refer to outside aids, such as maps or charts, as part of this interaction [70]. The consequence of this eye gaze wandering behavior is that simply relying on face-eye-gaze matching for each individual frame can misidentify a subject as a bystander (i.e., false negative) and a bystander as a subject (i.e., false positive).

Challenge 2: Performing the aforementioned four tasks for every frame on the device is too costly and infeasible to keep up the frame rate. Face-eye-gaze matching (Task 3) and obscuration of bystander faces (Task 4) tend to be light-weight, and eye gaze tracking (Task 2) comes at almost no cost with hardware support (such as the Microsoft HoloLens 2 [44]). However, identifying faces in a frame (Task 1) generally requires performing face detection inferences, and high-accuracy face detection using deep neural network (DNN) models (e.g., DeepFace [62]) tends to be compute-intensive and incurs a long inference time when running on resource-constrained mobile devices. For example, works such as [23, 24, 30, 37, 69] show that, while possible, on-device inference in resource-constrained mobile devices is generally difficult, and in this AR context, negatively impacts the frame rates that are directly correlated with user experience.[1] Even models specifically designed for use on mobile devices can limit frame rates to an unacceptable level [24, 36]. Therefore, we seek a practical solution that protects bystander privacy without compromising user experience by maintaining usable frame rates (e.g., near 60 FPS).

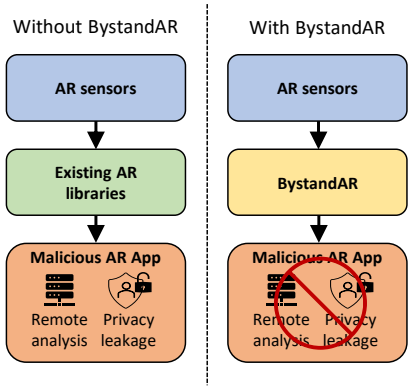


Figure 2: A malicious AR application is designed to request bystander visual data from a device, infer sensitive information from this data, and offload the inference results to another location for exploitation. BystandAR, shown between the device sensors and the malicious applications, is designed to prevent this.

5 BystandAR Design

In this section, we discuss the threat model that our proposed BystandAR system is designed against, and how the key system components of BystandAR overcome the two main challenges discussed in §4.2.

5.1 Threat Model and Our Approach

BystandAR is designed to prevent malicious AR applications running on AR devices from collecting sensitive information from visual data of bystanders of interpersonal interactions during the execution of the AR application. Such malicious applications may perform *hidden operations* that extract sensitive information in bystander visual data captured during AR application execution [34].

As AR applications, these malicious applications can have full access to AR device cameras, microphones, and network stack after a cursory set of permissions requests, which will be granted by the device user [19, 21]. We note that, while BystandAR is designed to prevent malicious activities, the design prevents accidental bystander privacy leaks as well. Fig. 2 gives an illustration of this threat.

[1] Microsoft recommends that applications using their HoloLens 2 maintain a frame rate of 60 FPS in order to provide a positive user experience [40].

The high-level approach of BystandAR to overcome the above threat is to intercept the frame input operations of the application by (1) modifying the operating system's method of allowing access to raw visual data frames, (2) identifying the bystanders/subjects and obscuring bystander faces accordingly, and then (3) passing on the obscured frames to the application. Alternatively, one can implement BystandAR by requesting applications to use special APIs for reading visual data frames. These APIs, which are provided as libraries or a modified framework, implement the key tasks performed by BystandAR, noted above. During application installation, permissions will be given to these special APIs but not the regular APIs for reading visual data frames.

5.2 High-Accuracy Bystander Detection using History Information

Recall from Challenge 1 in §4.2 that we cannot simply match the location of detected faces in every frame with the user's eye gaze. Doing so would disregard the assumption that the user's eye gaze will wander as a natural part of human interaction. When this wandering gaze intersects with a bystander, this bystander could erroneously be labeled a subject and left unprotected in the output frame, violating bystander privacy.

We overcome this challenge by collecting data on the history of the user's gaze, as opposed to instantaneous information. We then use the historical information for different persons in these recent frames to determine who is the *subject(s)*. Specifically, the identification of the subject/bystander in the current frame is controlled by a threshold. Since we have two input modalities (i.e., eye gaze and voice), the threshold is two-fold. We set a threshold for purely eye gaze contact with a face and one for eye gaze and simultaneous voice contact. Since previous work suggests that eye and voice contact is more indicative of human attention [60, 70], we give the eye gaze threshold with voice a lower value. We do not provide for a voice-only threshold, as the user's voice alone does not identify a face.

These thresholds are the minimum total rate of eye/voice contact over the life of the detection and are informed by the works discussed in §4.1. These works give a range of eye gaze expected in conversations ranging from 41% to 73% and can be as low as 6% when the user is referring to maps, charts, or other visual aids. Additionally, when speaking to a person, eye contact is made less often than when listening to a person speak. Since BystandAR has no way of knowing if the user is listening, we treat the user's eye gaze and voice as a more sure sign of a conversation than purely eye gaze and provide a lower threshold when voice is present.

Algorithm 1 shows the history-based bystander detection algorithm. First, BystandAR uses the eye-tracking sensors and wearer-focused microphones present on nearly any modern AR device [38, 39, 42], to log the device user's eye gaze and voice data and establish context (Lines 5 and 6).

To determine the history of a user's contact with persons in the field of view, we also monitor the total amount of eye/voice contact with every detected face (Lines 7-14). Every face detection begins labeled a bystander. If over the life of the detection, the user has made enough contact with the detection to meet the threshold, the face is labeled a subject. BystandAR also allows for labeling multiple subjects using this method. It is possible that the label of *subject* can revert to *bystander* if the threshold for contact is no longer met.

Algorithm 1 BystandAR Control Loop

```
1:  Parameters: sampling interval N
2:  FrameCounter = 0
3:  while True do
4:      Increment FrameCounter
5:      Compute the current location of the eye gaze
6:      Monitor if voice input is above noise floor
7:      if Eye-gaze/Voice intersects with a face then
8:          Increment eye/voice tracker for face
9:          if Eye-gaze/Voice history > Threshold then
10:             Label face a subject
11:         else
12:             Label becomes/remains bystander
13:         end if
14:     end if
15:     if FrameCounter ≥ N then
16:         FrameCounter = 0
17:         Retrieve raw depth and camera frames
18:         Infer location of all faces in frame
19:         for each Face detected do
20:             Transform 2D detection to 3D world space
21:             if face overlaps with an existing face then
22:                 Replace current detection; reset TTL
23:             else
24:                 Create new detection
25:             end if
26:             if Application requesting sensor data then
27:                 Obscure bystander faces in frame
28:             end if
29:         end for
30:     else
31:         if Application requesting sensor data then
32:             Obscure bystander faces in frame
33:         end if
34:     end if
35:     Release frames to application
36: end while
```

5.3 On-Device Bystander Detection via Periodic Face Detection

The history-based bystander detection method described above still assumes performing face detection on *every* camera frame. As noted in Challenge 2 in §4.2, performing face detection on every frame is too costly on mobile devices and cannot keep up with the high frame rate (e.g., near 60 FPS) needed to support a high quality of user experience (QoE). To overcome this challenge, we explore how to avoid performing face detection on every frame.

If during an interpersonal interaction, the AR device does not move, consecutive frames captured by the camera would have the same spatial frame of reference, and we could simply skip every N frames for face detection in the above history-based algorithm

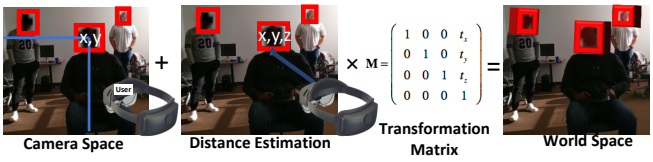


Figure 3: An illustration of the 2D-to-3D camera-to-world transformation process. Here, matrix M is an arbitrary transformation matrix created from the metadata included with each captured frame; it is used to convert 3D points in camera space to world space.

and still be able to successfully match faces with the user's eye gaze and voice. One challenge is that the history of eye-gaze and voice information needs to be accumulated for the same person across frames, but the faces (of bystanders or the subject) may move across frames. We thus need to keep track of their movement so we know faces in different frames correspond to the same person. This could be achieved by lightweight motion tracking techniques such as optical flow [50].

The above simple frame-skipping scheme, however, cannot handle the movement of the device itself. This can happen often in interpersonal interactions as the user moves her head and position and changes the spatial frame of reference of consecutive frames, which makes it much harder to match faces in different frames to the same person.

To tackle this challenge, we observe that one of the built-in capabilities of AR devices, SLAM, which is the foundation of the device's spatial awareness, can be used to track the location of a detected face when the device moves. We can exploit this unique AR device capability to compensate for the movement of the device.

To do this, we estimate the time required for a face to move outside of a face detection's bounding box (described in more detail in §8 (BystandAR Limitations)). From this, we can then estimate the number of subsequent frames N during which the faces will remain in the same bounding box, and skip face detection for these N frames. During the capture of these subsequent N frames, we assume that the face will remain inside its previous bounding box, removing the need for per-frame detection. In other words, we exploit a novel capability of AR devices to track faces over time, even with device movement, which removes the requirement to perform face detection more often than the device can support.

Tracking face movement using 3D-2D transformation. Using Algorithm 1, we explain this process as follows:

If a frame is selected for inference at the interval N, we capture the frame (Line 17) and infer the location of each face in the captured frame (Line 18). We must now locate each face detected in this frame using an absolute spatial reference, called a *world coordinate system* or *world space* in AR [49]. By doing so, we ensure that the face can be accounted for as the user moves, even when this motion causes the face to move completely out of the next camera frame. We can use the method provided by AR cameras to convert a 2D point (e.g., that of a face detected in the 2D frame) to the 3D world space (Line 20). If a face overlaps with a previous detection, indicating that the face has moved, we update the location of the face. In this way, we can track face movement (Lines 21-25). To prevent stale faces, if the face has not been updated in a given Time-to-Live (TTL) window, we remove it (Line 22). Fig. 3 presents an illustration of this process.

5.4 Frame Obscuration

Any usable solution to increase bystander expectations of privacy, must also allow third-party applications access to any obscured frames. In the final step, BystandAR performs obscuration of the faces of detected bystanders in every frame and its corresponding depth frame, before passing them to the requesting AR applications (Lines 31 and 32).

Specifically, every frame is compared against existing, detected faces for potential obscuration to ensure bystander privacy as follows. As each new frame is captured, BystandAR compares existing facial detections and labels to the current frame. Using a system similar to the one we designed to create 3D locations from 2D detections, it then converts the 3D location of any detected face to a 2D position relative to the camera's current frame. If the detection has been labeled a subject and the user is making eye contact, we do nothing. If it has been labeled a bystander or is a subject not currently under the user's attention, we obscure it in both the camera and depth frame.

5.5 Putting It All Together - BystandAR Architecture

Fig. 4 shows the architecture of our proposed BystandAR system. The camera and depth frames are continuously captured by the AR device camera. At a given sampling interval, the face detection module infers the 2D location of any faces present in the frame, and BystandAR locates these faces in 3D after 2D-to-3D transformation. Using this location, we create a 3D bounding box, invisible to the user, that serves as the 3D anchor for each detection. By default, these faces are labeled bystanders. As sampled face detection continues and the position of the face changes, BystandAR updates the location of the face and moves the 3D bounding box accordingly. We provide a discussion about the speed of face movement BystandAR can tolerate in §8.

In parallel with the above face detection and tracking process, BystandAR collects information about the user's eye gaze and voice using the AR device's onboard eye gaze tracking and wearer-focused microphone. For every camera frame, BystandAR tracks on which face the user's attention is currently focused on and maintains a history of this information for all currently detected faces. Once the history of the user's attention (eye gaze or simultaneous eye gaze and voice input) meets a pre-specified threshold, the detection is labeled a subject. With this context, the face obscuration module obscures the faces of each detection as required. After bystander visual data has been removed from each frame, the frame is safe for release to any third-party application.

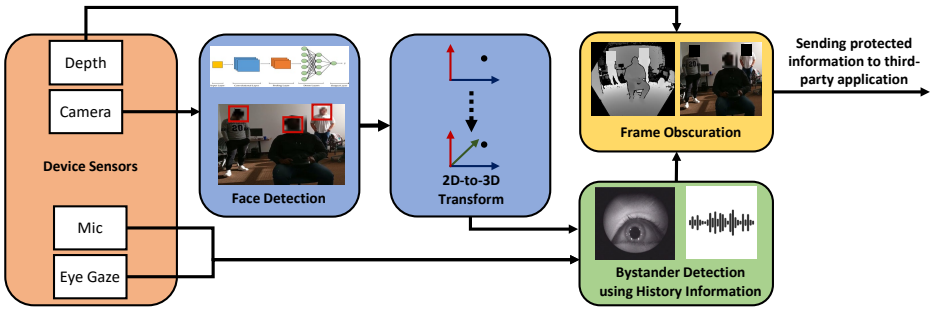


Figure 4: BystandAR Architecture. Raw data is captured from the device's sensors and is used both in face detection and learning eye gaze and voice history information for bystander detection. Afterward, bystander detection is used to obscure human faces not designated subjects in both camera and depth frame data.

6 Prototype Implementation

We built the BystandAR prototype in Unity 2021.3.11f1, using Microsoft's Mixed Reality Toolkit (MRTK) version 2.7.2, and we deployed BystandAR on a Microsoft HoloLens 2 device running Windows Holographic for Business Build 20348.1528. The core functionality of BystandAR was implemented in about 1,200 lines of code. All source code was completed in C#.

We use Microsoft's *MediaCapture* class [46] to capture camera and depth frames, as well as collect the transformation matrices required for the 2D-to-3D detection conversion. The *FaceDetector* class of Microsoft's *FaceAnalysis* namespace is used for face detection [43]. This library provides accurate and tested functionality for determining the location of faces in 2D images, something we require to infer the 3D location of faces using spatial awareness. This face is represented by a Unity GameObject [68]. The onboard SLAM then continuously monitors the face in the physical world as the user moves. This functionality is not designed by us but comes "for free" with AR devices. We use a sampling rate of every *8* frames, informed by pilot testing and determined to be a good balance of accuracy and device resource load. This rate was the highest inference rate where we could expect 50+ FPS of frame rate during pilot testing. On a system with more computing power, this rate could be increased, yielding higher accuracy and similar frame rates. We collect camera frames with an input resolution of 1290×1080 pixels, a parameter determined in pilot testing to be a balance of inference accuracy and latency. We test our implementation at two different threshold levels. One, designed to test the higher limits of expected human eye/voice contact, sets a minimum of 50% pure eye contact or 25% simultaneous eye gaze and voice contact over the life of the detection. A lower threshold, 25% and 15%, respectively, was designed to test the lower bound of expected contact. These thresholds were created after studying literature on the dynamics of human eye contact in interpersonal interactions (see §4.1), and validated during our pilot testing. They are designed to serve as the highest and lowest values of the middle 50% of expected eye and voice contact between human being engaging in a conversation. However, these values can be altered for optimal use across different contexts.

For the obscuration of the frame, we use a complete mask of the camera frame informed by the face detection's bounding box. The complete masking, as opposed to blurring, is informed by works (e.g., [71]) that show *deblurring* of such an image is indeed possible. Unique to the depth obscuration, we do not simply change the bit values of the raw depth data to create a mask, but seek to *plateau* the existing data to keep from creating "depth holes" in the image. Instead of completely masking the depth area, we smooth over the face of bystanders using the depth at the edges of the detection as a reference. This keeps from creating "depth holes" artificially. As noted in works such as [72], depth holes create inconsistencies that can hamper the AR user experience. If using the depth frame, knowledge of the presence of the face is still kept, but the details and potentially uniquely identifying contours of the face are removed. We show an example of the output of BystandAR in Fig. 5.

(a) Camera Frame **(b) Depth Frame**

Figure 5: An illustration of the output of BystandAR if a third-party application is requesting visual data. We blur the face of the subject only to protect the participant's identity. Note: the "boxes" over the face of the bystanders in the depth data in (b) reflect the depth of the bystander themselves in a plateauing manner.

We use the HoloLens 2's onboard eye gaze tracking, wearer-focused microphone, and built-in spatial awareness to accomplish these tasks through the MRTK [48]. These APIs allow us to leverage the novel capabilities of AR devices in order to fully implement the BystandAR design.

7 Evaluation

Our evaluation consists of 16 participants. It is designed to not only test the effectiveness of BystandAR (i.e., the ability of BystandAR to differentiate the subject from a bystander) but also to evaluate

performance when implemented on a Microsoft HoloLens 2. We evaluate the effectiveness of BystandAR by evaluating the success rate, defined as the amount of correctly obscured faces compared to the total number of detected faces corresponding to bystanders in every frame. We evaluate the performance of our prototype by measuring the frame rate, compared with the minimum frame rate for preserving the user experience recommended by the device manufacturer. Additionally, we measure the effect BystandAR has on bystander perceptions of privacy in the presence of AR devices with a post-testing survey.

7.1 Evaluation Procedure

The evaluation for BystandAR has been approved by our organization's Institutional Review Board. Sixteen participants were recruited using a graduate student mailing list and using distribution lists for undergraduate Human-Computer Interaction courses. All participants were 18 or older. Prior to testing, each user was given a 10-minute tutorial on AR gestures, specific to the Microsoft HoloLens 2, and given instructions on fitting and operating the device. Additionally, each user was instructed to complete an eye gaze calibration using HoloLens 2's calibration function. For evaluation, the prototype was designed to offload every 10th obscured frame. This is separate from the inference sampling interval and was used only to capture obfuscated data for evaluation.

The sixteen participants were grouped into eleven tests that contain one user and one to three bystanders. Every test contains a subject, except for the "no subject" test in §7.4. Among the eleven tests, two contain three bystanders, seven contain two bystanders, and two contain a single bystander; five contain the movement of bystanders, and six do not. During this test, the AR user was instructed to ask questions of the partner (i.e., subject) seated approximately 2 meters from them, for a total of 3 minutes. The AR user then swapped roles with their partner by giving them the AR headset and repeating the test.

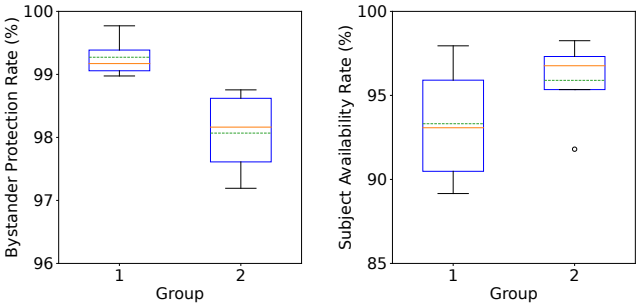
We divided the data collection sessions into two groups in order to test two eye gaze and voice thresholds. Group 1 used a gaze threshold of 50% contact over the life of the face detection. Since we know that eye gaze and voice have a stronger correlation with the user's attention, we gave a lower threshold of 30% to eye gaze contact when the AR user is speaking. Group 2 had these thresholds set to 25% and 15%, respectively, to explore a shorter duration for subject detection.

7.2 Evaluation Metrics

For each obscured frame, we analyze the effectiveness of the bystander protection mechanism by using *DeepFace* [62], an open-source facial analysis tool, to evaluate BystandAR's ability to identify subject/bystanders, and when necessary, protect them. Any bystander face found by DeepFace in the obscured frame indicated a failure of our system. In this experiment, we used a confidence threshold of 90% for DeepFace detections, as recommended by the model author [61]. We compute the total amount of obscured bystander faces compared to the total amount of detected bystander faces as the *bystander protection rate*.

We then analyze each frame to quantify the prototype's effectiveness in determining the subject of the interaction. Knowing the identity of the intended subject, we inspect every frame and record whether the face of the subject was unobscured. If the eye gaze of the AR device user was directed at the face of the subject, we expect the face of the subject to be unobscured. If the eye gaze is directed at the subject and the face remains obscured, we consider this a failure. We compute the total number of subject faces properly unobscured compared to the total number of subject faces as the *subject availability rate*.

Finally, in order to measure performance (in terms of frame rate), we separately calculate the FPS of the prototype as it runs on the Microsoft HoloLens 2. Each FPS calculation is done over a testing period of 3 minutes with the prototype obscuring bystanders in each frame in order to simulate a third-party application requesting visual data.



(a) Bystander protection **(b) Subject availability**

Figure 6: BystandAR effectiveness across the two groups of eye gaze and voice thresholds.

7.3 Effectiveness of BystandAR

Camera Data. We evaluate how well BystandAR, using eye gaze and voice data from the user, protects bystanders from DeepFace face detection on both camera and depth frames captured by the HoloLens 2's onboard camera.

We compare the two groups of eye gaze and simultaneous eye gaze and voice thresholds, Group 1 (50% gaze and 30% gaze/voice) and Group 2 (25% gaze and 15% gaze/voice), to determine which is ideal for BystandAR effectiveness. Fig. 6 illustrates the tradeoff in bystander privacy protection between the two groups. Lower thresholds result in lower rates of bystander privacy protection but a higher subject availability rate. Specifically, the impact of a lower threshold on bystander protection rate is marginal (about 1%), and bystander protection rates of both two threshold options were high (99.32% and 98.14% for Groups 1 and 2, respectively). In terms of subject availability rate, the lower thresholds can improve it by about 2.6%, from 93.63% to 96.27%. In our system, we use the lower thresholds (used in Group 2) as the default option since it provides a more balanced performance for both subject and bystanders.

Depth Data. BystandAR also provides bystander protection on depth frames. For each frame, using the established face detections and their label as a "bystander" or a "subject", we obscure faces using a depth mask that is the same as the depth levels surrounding the detection. This provides the *plateau* effect mentioned in §5. While the HoloLens 2's depth data was insufficiently detailed to

use existing depth-based face detection methods, we verified that every depth frame was obscured in the same way as its camera frame counterpart.

Threshold Recommendation. Between Group 1 and Group 2 thresholds, we achieved comparable bystander protection rates, with a 2.6% improvement in subject availability rates. For this reason, we recommend the lower threshold of Group 2. With this in mind, we conduct all following evaluations of BystandAR using both gaze and simultaneous gaze and voice thresholds of 25% and 15%, respectively.

7.4 Comparison to Offline Solution

We compare the bystander protection rate between BystandAR running in real-time and a highly accurate offline bystander detection model [12]. The offline approach extracts features from each face and classifies it as bystander or subject with up to 94.3% accuracy using Gradient Boosted Decision Tree (GBDT). The extracted features capture the 3D head pose, the angle between the gaze direction and the camera, if the face was out of focus, and the distance from the camera. The offline approach processed every frame and could not feasibly be implemented on an AR device in real-time.

Visual data from three additional scenarios were collected for a robust comparison: a single bystander with no subject, a subject with static bystanders, and a subject with bystanders that include movement. The identified bystanders were positioned on either side of the subject and both in front of or behind them for 60 seconds at a time while camera frames from BystandAR were recorded. Raw frames were also recorded for input to the offline model. Each of the recording scenarios lasted for two minutes in total.

Table 1 presents the bystander protection rates between BystandAR and GBDT. Our BystandAR prototype performed better overall, with an overall rate of 94.1% compared to 82.3%. In the "No Subject" scenario, one individual was present but did not interact with the AR user. In this scenario, BystandAR had a perfect protection rate of 100%, while the GDBT model classified the face as a subject in every frame, producing a protection rate of 0%. BystandAR detects the subject based on eye gaze and voice interaction and does not suffer any false classification as a result. Accurate bystander recognition is important in the absence of a subject, as this is a typical scenario for AR users on a daily basis.

For scenarios that include a subject and multiple bystanders, both BystandAR and GBDT performed worse with a moving bystander. While BystandAR had a 2% higher protection rate with static bystanders, GDBT had a 3.3% higher rate with a moving bystander. Rates from these two scenarios show that a moving bystander is more difficult to identify and obscure, and an offline model applied to every frame is only marginally better than BystandAR running in real-time with an inference sampling interval of 8 frames.

7.5 Impact of Bystander Characteristics

As seen in Table 1, the effectiveness of BystandAR can vary based on the characteristics of bystanders. We further evaluate the impact of different numbers of bystanders in the scene and where motion degrades protection rates.

Table 1: An evaluation of BystandAR against an offline bystander detection model. The protection rate was highest for BystandAR when images contain no subject and static bystanders, while the offline model performed marginally better with a moving bystander.

Scenario	GBDT[12] Protection Rate	BystandAR Protection Rate
No Subject	0%	100%
Static Bystanders	95.3%	97.3%
Moving Bystander	91.6%	88.3%
Overall	**82.3%**	**94.1%**

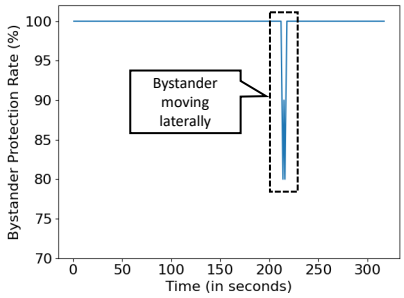


Figure 7: Bystander protection rates of BystandAR over a 10-second average. The inset box highlights a drop in protection rates during large bystander movement, but overall performance is maintained for the majority of the data.

The bystander protection rates across sessions with one, two, and three bystanders were computed from the procedure described in §7.1. We found average protection rates were the highest for three bystanders (99.26%), the median for two bystanders (98.74%), and the lowest for one bystander (98.30%). Overall, we found that increasing numbers of bystanders had a negligible impact on bystander protection.

Next, we evaluate BystandAR results temporally based on the motion level of bystanders. We identified data collection sessions where the bystander(s) made large movements, defined as movements from one side of the user's FOV to the other. In the tests with this motion, BystandAR protected the visual data of bystanders in 98.91% of frames. For tests where the bystanders remained static, we found a protection rate of 98.77%. Additionally, Fig. 7 shows a single test run and the impact that dramatic bystander movement had on the protection rate. Both the figure and the aggregate results show the minimal impact bystander movement had during our testing, but we further discuss situations where this result may differ in §8 (BystandAR Limitations).

7.6 Effect of BystandAR on Bystander Perceptions of Privacy

Following the completion of testing, each participant was asked to complete a brief survey designed to investigate the effect BystandAR has on bystander perception of privacy. Each respondent was first presented with a camera and a depth frame showing a subject,

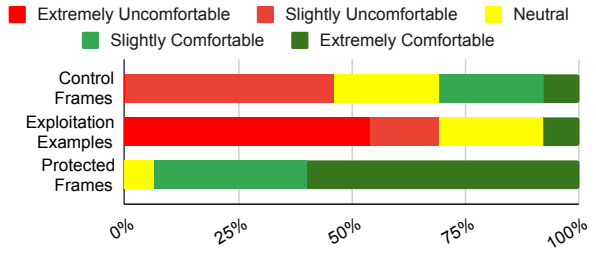


Figure 8: The distribution of the Likert responses for the bystander perception survey across the unmodified frames (Control Frames), the frames with example exploitation from DeepFace (Exploitation Examples), and the frames from BystandAR (Protected Frames).

and two bystanders from a user's point of view. The respondent was asked to assume they were one of the two bystanders and was then asked to rate their comfort level with this information being made available to a third-party application on a Likert scale of one to five with one being "Extremely Uncomfortable" and five being "Extremely Comfortable".

The respondents were then presented with two additional sets of images, one using added DeepFace inference results, and the other after BystandAR obscured the bystander faces in the camera and depth frame. For the protected set, respondents were told that the image was protected from exploitation like DeepFace inference, but also to assume this system was less than perfect. They were then asked to rate their comfort level for each. Fig. 8 shows the aggregated results of this survey. We further discuss these results in §8 (Bystander Perceptions).

7.7 Overhead of BystandAR

Using an inference sampling interval of 8 frames, BystandAR runs at **52.6 FPS** while not being required to obscure frames for release to a third-party application. In fact, this method is likely to be the most widely used if the device is not running an application that requires camera or depth frames. In this mode, BystandAR still collects raw images and obscures faces according to the bystander/subject detection described in §5. It does not, however, apply any masks to any output images. This frame rate is comparable to the HoloLens 2's recommended 60 FPS (when not capturing media frames) as suggested by Microsoft [40]. When the prototype is configured to release obscured frames, BystandAR achieves **33.6 FPS**. Such a drop in frame rate is expected for our prototype, as Microsoft's standard sensor data logging API states that frame rates will drop to around 30 FPS [47]. We note here that BystandAR would switch between offloading frames and not based on whether a third-party application was requesting them.

To stress test our system, we also ran tests using an inference sampling interval of 1 frame, meaning every captured camera frame was used for inference. While not obscuring frames (i.e., simulating running while no third-party application is requesting frames), the prototype runs at 15.2 FPS; while obscuring frames, the prototype runs at 11.5 FPS. This significant drop shows the problems with per-frame inference as stated in §4 as both values are well below any thresholds recommended by Microsoft for usable frame rates as a result of inferring on every frame.

Table 2: A summary of BystandAR's impact on the HoloLens 2's system resources, compared to device idle.

Total System Load	Avg. CPU Usage	Avg. GPU Usage	System Memory	Power Util.
w/BystandAR	72%	0%	2.8 GB	68%
System Idle	45%	0%	2.2 GB	61%

Finally, Table 2 shows a breakdown of BystandAR impacts on the system resources of the HoloLens 2, as compared to the device's idle load. As a third-party application, BystandAR increases the CPU load on the HoloLens 2 by 27%, with minimal increases in memory footprint and power consumption. Also, given the 7% increase in power utilization, we project a roughly 12-15 minute decrease in the 2-3 hour expected battery life of the Microsoft HoloLens 2 [44]. We believe that BystandAR would need to be implemented at the OS level, as discussed in §8, reducing these requirements.

8 Discussion

Scope. BystandAR is designed to protect bystanders (and make subjects available) in interpersonal interactions. Scenarios such as jogging, where the user is not making meaningful contact and moving quickly would be less advantageous for our system. However, the impact of the speed of the user or bystander's movement can also be mitigated, as discussed later in this section.

Bystander Perceptions. As shown in Fig. 8, BystandAR increases bystander confidence in the protection of their visual information in the presence of AR devices. Even when told that such a system may not be completely successful in all use case scenarios, participants were generally more comfortable with a third-party application having access to their visual data, assuming that a system such as BystandAR had protected their privacy first. We believe that systems such as BystandAR are even more effective at increasing perceptions of privacy when bystanders are familiar with the potential threats as well as the protection provided.

OS-level Implementation. In the current demonstration prototype, BystandAR is implemented as a third-party application running on the Microsoft HoloLens 2's Windows Holographic OS. On the HoloLens 2, only one AR application is allowed to run at a time. BystandAR cannot intercept and obscure frames as a sole third-party application. An assumption stated in multiple areas of this work is that, if ever implemented on a production system, BystandAR would need to be implemented at the OS level. This could add challenges, such as how and when to update inference models used by core OS processes, as was required during the COVID-19 pandemic [10]. This also provides some advantages for efficiency, as toolkits and APIs such as MRTK [48] might not be necessary. We do not investigate this further but leave this for future work.

Performance in Multi-user AR scenarios. As part of our testing design, we also sought to evaluate the effectiveness of BystandAR if the subject was also an AR device user. For this, we designed a second test that uses Microsoft's Azure Spatial Anchors [41] to share an absolute understanding of a physical space. The users

were required to collaborate by building a block structure while the application synchronized the location of their blocks on a cloud game server created by Photon Unity Networking [54]. The chosen model, FaceDetector [43], proved to be unreliable when detecting the faces of "subjects" wearing AR devices but just as reliable for device-less bystanders as expected. Models such as these can be retrained to identify faces with AR devices, similar to the case in the recent pandemic [10], but we believe that training models specifically for this purpose is beyond the scope of our work.

BystandAR Limitations. BystandAR is built around the idea that per-frame inference is not necessary to create a reliable system. This is grounded in the fact that face movement, like movement in all media capture, is replicated with a series of frames giving the illusion of actual movement. Faces, people, things, etc., are not "moving" in videos, they are merely shifting in relative position across every frame. We believe that, if the inference rate is fast enough, the bystander/subject face cannot move outside of the obscuration between inferences. However, this does have a limit. For example, at a frame rate of 52 FPS and an inference sampling interval of 8, like BystandAR, an inference occurs about every 154 milliseconds. Given an average face width of 0.15 meters [45], a face at the center of an obscuration box must move 0.15 meters in 154 milliseconds to evade the box and be fully revealed, which leads to the following speed: $\frac{0.15 \text{ meters}}{0.154 \text{ seconds}} = 0.97$ m/s. Given this speed, a face could elude the inference rate and be unprotected. Even in testing, of the small number of bystander faces exposed, about 50% were due to movement. This can be ameliorated through more rapid inference at the cost of lower frame rates. More advanced and powerful AR devices in the future can allow more rapid inference with less (if any) frame rate degradation.

BystandAR is designed to work in a variety of different interpersonal situations. For instance, we asked bystanders to stand, sit, move, or remain still during the various tests conducted. However, we never asked a bystander to move in front of the subject during testing. As the bystander's face would be recognized as a face that had not yet met the thresholds of eye/voice contact to be labeled a "subject", this face would be obscured in any output image. If this face was directly in front of the subject, BystandAR would obscure both faces and temporarily render the subject unavailable in the image for as long as this occurred. Additionally, BystandAR works on the principle that only one subject should be revealed at once. While BystandAR is capable of classifying more than one face as a subject, we only reveal the face of the subject when the user's eye gaze intersects with the subject's face. This could be altered to allow both subjects' faces.

Another limitation comes when we consider a malicious user. BystandAR is built with the natural dynamics of human eye and voice contact in mind. Given the social barriers to staring at persons not part of an interaction, shown in §4, we also believe that the risk of users unintentionally staring at the exposed face of a bystander is possible but limited. However, if the user intends to expose the face of a bystander without them being part of an AR interaction, they can certainly choose to keep eye contact on an exposed face to prevent obscuration by the system. This would override the assumptions made about normal human eye and voice contact. It would also be noticeable to the bystander and other parties that a person not involved in an interaction or conversation was staring at a seemingly random person. We present BystandAR not as the perfect solution, but rather as an example of an on-device, context-enabled method to further bystander protection in AR systems.

9 Conclusion

In this work, we harnessed the dynamics of human interaction to improve bystander visual data protection in AR devices by creating a novel system called BystandAR. This is achieved *on-device* while maintaining usable frame rates on AR devices. We believe that this work expands the understanding of the capability of modern AR devices to protect bystander privacy and to further the trust of bystanders that their privacy is protected, using unique capabilities that only these exciting, advanced AR devices possess.

Acknowledgments

We thank the anonymous shepherd and reviewers for their insightful feedback. We also thank Dr. Qinghua Li and his coauthors of [12] for kindly sharing their source code and dataset. Additionally, we thank the user study participants for volunteering their time. This work is supported in part by the Commonwealth Cyber Initiative (CCI) and the NSF grants under CNS 2112778 and 2153397.

References

[1] Paarijaat Aditya, Rijurekha Sen, Peter Druschel, Seong Joon Oh, Rodrigo Benenson, Mario Fritz, Bernt Schiele, Bobby Bhattacharjee, and Tong Tong Wu. 2016. I-Pic: A Platform for Privacy-Compliant Image Capture. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services* (Singapore, Singapore) *(MobiSys '16)*. Association for Computing Machinery, New York, NY, USA, 235–248. https://doi.org/10.1145/2906388.2906412

[2] Hironori Akechi, Atsushi Senju, Helen Uibo, Yukiko Kikuchi, Toshikazu Hasegawa, and Jari K Hietanen. 2013. Attention to eye contact in the West and East: autonomic responses and evaluative ratings. *PLoS One* 8, 3 (March 2013), e59312.

[3] Taslima Akter, Tousif Ahmed, Apu Kapadia, and Manohar Swaminathan. 2022. Shared Privacy Concerns of the Visually Impaired and Sighted Bystanders with Camera-Based Assistive Technologies. *ACM Trans. Access. Comput.* 15, 2, Article 11 (may 2022), 33 pages. https://doi.org/10.1145/3506857

[4] Thomas Alsop. 2022. Global Mobile Augmented Reality (AR) user devices 2024. https://www.statista.com/statistics/1098630/global-mobile-augmented-reality-ar-users/

[5] M. Argyle. 1994. *The Psychology of Interpersonal Behaviour.* Penguin Books Limited. https://books.google.com/books?id=VQOzdOxJFZAC

[6] Nicola Binetti, Charlotte Harrison, Antoine Coutrot, and Isabelle Johnston, Alan ad Mareschal. 2016. Pupil dilation as an index of preferred mutual gaze duration. *R. Soc. open sci.* 3, 7 (July 2016), 0–0. https://doi.org/10.1098/rsos.160086

[7] Ricard Borràs, Àgata Lapedriza, and Laura Igual. 2012. Depth Information in Human Gait Analysis: An Experimental Study on Gender Recognition. In *Image Analysis and Recognition*, Aurélio Campilho and Mohamed Kamel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 98–105.

[8] Zijun Cheng, Tianwei Shi, Wenhua Cui, Yunqi Dong, and Xuehan Fang. 2017. 3D face recognition based on kinect depth data. In *2017 4th International Conference on Systems and Informatics (ICSAI)*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 555–559. https://doi.org/10.1109/ICSAI.2017.8248353

[9] Edward Chou, Matthew Tan, Cherry Zou, Michelle Guo, Albert Haque, Arnold Milstein, and Li Fei-Fei. 2018. Privacy-Preserving Action Recognition for Smart Hospitals using Low-Resolution Depth Images. *CoRR* abs/1811.09950 (2018). arXiv:1811.09950

[10] James Clayton. 2022. *Facial recognition beats the Covid-mask challenge.* BBC. Retrieved Nov 6, 2022 from https://www.bbc.com/news/technology-56517033

[11] David Darling. 2021. *Automated Privacy Protection for Mobile Device Users and Bystanders in Public Spaces.* Master's thesis. Retrieved from https://scholarworks.uark.edu/etd/4218.

[12] David Darling, Ang Li, and Qinghua Li. 2019. Identification of Subjects and Bystanders in Photos with Feature-Based Machine Learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 1–6.

[13] David Darling, Ang Li, and Qinghua Li. 2020. Automated Bystander Detection and Anonymization in Mobile Photography. In *International Conference on Security and Privacy in Communication Systems*. Springer, 402–424.

[14] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. 2021. Let's SOUP up XR: Collected thoughts from an IEEE VR workshop on privacy in mixed reality. In *VR4Sec: Security for VR and VR for Security, SOUPS 2021 Workshop*.

[15] Jaybie A. De Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. 2019. Security and Privacy Approaches in Mixed Reality: A Literature Survey. *ACM Comput. Surv.* 52, 6, Article 110 (oct 2019), 37 pages. https://doi.org/10.1145/3359626

[16] Ziedune Degutyte and Arlene Astell. 2021. The Role of Eye Gaze in Regulating Turn Taking in Conversations: A Systematized Review of Methods and Findings. *Frontiers in Psychology* 12 (2021). https://doi.org/10.3389/fpsyg.2021.616471

[17] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. 2014. In Situ with Bystanders of Augmented Reality Glasses: Perspectives on Recording and Privacy-Mediating Technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2377–2386. https://doi.org/10.1145/2556288.2557352

[18] Mariella Dimiccoli, Juan Marín, and Edison Thomaz. 2018. Mitigating Bystander Privacy Concerns in Egocentric Activity Recognition with Deep Learning and Intentional Image Degradation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 132 (jan 2018), 18 pages. https://doi.org/10.1145/3161190

[19] Drewbatgit. 2021. *App Capability declarations - UWP applications*. Microsoft. Retrieved Sept 29, 2022 from https://learn.microsoft.com/en-us/windows/uwp/packaging/app-capability-declarations

[20] Passant Elagrousy, Mohammed Khamis, Florian Mathis, Diana Irmscher, Ekta Sood, Andreas Bulling, and Albrecht Schmidt. 2023. Impact of Privacy Protection Methods of Lifelogs on Remembered Memories. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3544548.3581565

[21] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android Permissions: User Attention, Comprehension, and Behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (Washington, D.C.) *(SOUPS '12)*. Association for Computing Machinery, New York, NY, USA, Article 3, 14 pages. https://doi.org/10.1145/2335356.2335360

[22] Sarwesh Giri, Gurchetan Singh, Babul Kumar, Mehakpreet Singh, Deepanker Vashisht, Sonu Sharma, and Prince Jain. 2022. Emotion Detection with Facial Feature Recognition Using CNN and OpenCV. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 230–232. https://doi.org/10.1109/ICACITE53722.2022.9823786

[23] Google. 2023. *Why On-Device Machine Learning?* Google. Retrieved Oct 15, 2022 from https://developers.google.com/learn/topics/on-device-ml/learn-more

[24] Yongjie Guan, Xueyu Hou, Nan Wu, Bo Han, and Tao Han. 2022. Realtime 3D Object Detection for Headsets. *CoRR* abs/2201.08812 (2022). arXiv:2201.08812

[25] Matthew Allan Hamilton, Anthony Paul Beug, Howard John Hamilton, and Wil James Norton. 2021. Augmented Reality Technology for People Living with Dementia and Their Care Partners. In *2021 the 5th International Conference on Virtual and Augmented Reality Simulations* (Melbourne, VIC, Australia) *(ICVARS 2021)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/3463914.3463918

[26] Rakibul Hasan, David Crandall, Mario Fritz, and Apu Kapadia. 2020. Automatically Detecting Bystanders in Photos to Reduce Privacy Risks. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 318–335. https://doi.org/10.1109/SP40000.2020.00097

[27] Jinhan Hu, Andrei Iosifescu, and Robert LiKamWa. 2021. LensCap: Split-Process Framework for Fine-Grained Visual Privacy Control for Augmented Reality Apps. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 14–27. https://doi.org/10.1145/3458864.3467676

[28] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim. 2016. Human Depth Sensors-Based Activity Recognition Using Spatiotemporal Features and Hidden Markov Model for Smart Environments. *Journal of Computer Networks and Communications* 2016 (04 Oct 2016), 8087545. https://doi.org/10.1155/2016/8087545

[29] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. 2013. A Scanner Darkly: Protecting User Privacy from Perceptual Applications. In *2013 IEEE Symposium on Security and Privacy*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 349–363. https://doi.org/10.1109/SP.2013.31

[30] Fucheng Jia, Deyu Zhang, Ting Cao, Shiqi Jiang, Yunxin Liu, Ju Ren, and Yaoxue Zhang. 2022. CoDL: Efficient CPU-GPU Co-execution for Deep Learning Inference on Mobile Devices. In *The 20th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'22)*. Microsoft, ACM. https://www.microsoft.com/en-us/research/publication/codl-efficient-cpu-gpu-co-execution-for-deep-learning-inference-on-mobile-devices/

[31] Marion Koelle, Swamy Ananthanarayan, Simon Czupalla, Wilko Heuten, and Susanne Boll. 2018. Your Smart Glasses' Camera Bothers Me! Exploring Opt-in and Opt-out Gestures for Privacy Mediation. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction* (Oslo, Norway) *(NordiCHI '18)*. Association for Computing Machinery, New York, NY, USA, 473–481. https://doi.org/10.1145/3240167.3240174

[32] Marion Koelle, Matthias Kranz, and Andreas Möller. 2015. Don't Look at Me That Way! Understanding User Attitudes Towards Data Glasses Usage. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 362–372. https://doi.org/10.1145/2785830.2785842

[33] Pavan Kunchala. 2021. Real-time age gender detection using opencv. https://medium.com/analytics-vidhya/real-time-age-gender-detection-using-opencv-fa705fe0e1fa

[34] Sarah M. Lehman, Abrar S. Alrumayh, Kunal Kolhe, Haibin Ling, and Chiu C. Tan. 2022. Hidden in Plain Sight: Exploring Privacy Risks of Mobile Augmented Reality Applications. *ACM Trans. Priv. Secur.* 25, 4, Article 26 (jul 2022), 35 pages. https://doi.org/10.1145/3524020

[35] Ang Li, Qinghua Li, and Wei Gao. 2016. PrivacyCamera: Cooperative Privacy-Aware Photographing with Mobile Phones. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 1–9. https://doi.org/10.1109/SAHCN.2016.7733008

[36] Linzaer. 2022. *Ultra-Light-Fast-Generic-Face-Detector-1MB*. GitHub. Retrieved Oct 15, 2022 from https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB

[37] Luyang Liu, Hongyu Li, and Marco Gruteser. 2019. Edge Assisted Real-Time Object Detection for Mobile Augmented Reality. In *The 25th Annual International Conference on Mobile Computing and Networking* (Los Cabos, Mexico) *(MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 25, 16 pages. https://doi.org/10.1145/3300061.3300116

[38] MagicLeap. 2022. *Eye Tracking - Unity*. MagicLeap. Retrieved Oct 16, 2022 from https://ml1-developer.magicleap.com/en-us/learn/guides/eye-tracking-tutorial-unity

[39] Meta. 2022. *INTRODUCING META QUEST PRO, AN ADVANCED VR DEVICE FOR COLLABORATION AND CREATION*. Meta. Retrieved Oct 16, 2022 from https://www.oculus.com/blog/meta-quest-pro-price-release-date/

[40] Microsoft. 2021. *Hololens 2*. Microsoft. Retrieved Oct 15, 2022 from https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/hologram-stability

[41] Microsoft. 2022. *Azure Spatial Anchors*. Microsoft. Retrieved Nov 6, 2022 from https://azure.microsoft.com/en-us/products/spatial-anchors/

[42] Microsoft. 2022. *Eye tracking on HoloLens 2*. Microsoft. Retrieved Oct 16, 2022 from https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking

[43] Microsoft. 2022. *FaceDetector Class*. Microsoft. Retrieved Oct 16, 2022 from https://learn.microsoft.com/en-us/uwp/api/windows.media.faceanalysis.facedetector?view=winrt-22621

[44] Microsoft. 2022. *Hologram stability*. Microsoft. Retrieved Oct 15, 2022 from https://www.microsoft.com/en-us/hololens/hardware

[45] Microsoft. 2022. *Holographic face tracking sample*. Microsoft. Retrieved Oct 15, 2022 from https://learn.microsoft.com/en-us/samples/microsoft/windows-universal-samples/holographicfacetracking/

[46] Microsoft. 2022. *MediaCapture Class*. Microsoft. Retrieved Oct 16, 2022 from https://learn.microsoft.com/en-us/uwp/api/windows.media.capture.mediacapture?view=winrt-22621

[47] Microsoft. 2022. *Mixed reality capture overview*. Microsoft. Retrieved Nov 6, 2022 from https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/mixed-reality-capture-overview

[48] Microsoft. 2022. *What is Mixed Reality Toolkit 2?* Microsoft. Retrieved Oct 13, 2022 from https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05

[49] Microsoft. 2022. *World locking and spatial anchors in Unity*. Microsoft. Retrieved Nov 6, 2022 from https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/spatial-anchors-in-unity?tabs=wlt

[50] NanoNets. 2022. *Introduction to Motion Estimation with Optical Flow*. NanoNets. Retrieved Dec 8, 2022 from https://nanonets.com/blog/optical-flow/

[51] Leysia Palen, Marilyn Salzman, and Ed Youngs. 2000. Going Wireless: Behavior & Practice of New Mobile Phone Users. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, USA) *(CSCW '00)*. Association for Computing Machinery, New York, NY, USA, 201–210. https://doi.org/10.1145/358916.358991

[52] Alfredo J Perez, Sherali Zeadally, and Scott Griffith. 2017. Bystanders' privacy. *IT Professional* 19, 3 (2017), 61–65.

[53] Alfredo J. Perez, Sherali Zeadally, Scott Griffith, Luis Y. Matos Garcia, and Jaouad A. Mouloud. 2020. A User Study of a Wearable System to Enhance Bystanders' Facial Privacy. *IoT* 1, 2 (2020), 198–217. https://doi.org/10.3390/iot1020013

[54] Photon. 2022. *The Ease-of-use of Unity's Networking with the Performance & Reliability of Photon Realtime.* PUN. Retrieved Nov 6, 2022 from https://www.photonengine.com/pun

[55] Pasika Ranaweera, Anca Delia Jurcut, and Madhusanka Liyanage. 2021. Survey on multi-access edge computing security and privacy. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1078–1124.

[56] Pei-Luen Patrick Rau, Jian Zheng, and Zhi Guo. 2021. Immersive reading in virtual and augmented reality environment. *Information and Learning Sciences* 122, 7/8 (01 Jan 2021), 464–479. https://doi.org/10.1108/ILS-11-2020-0236

[57] Nisarg Raval, Animesh Srivastava, Kiron Lebeck, Landon Cox, and Ashwin Machanavajjhala. 2014. MarkIt: Privacy Markers for Protecting Visual Secrets. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Seattle, Washington) *(UbiComp '14 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1289–1295. https://doi.org/10.1145/2638728.2641707

[58] Franziska Roesner, David Molnar, Alexander Moshchuk, Tadayoshi Kohno, and Helen J. Wang. 2014. World-Driven Access Control for Continuous Sensing. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 1169–1181. https://doi.org/10.1145/2660267.2660319

[59] Joe Saballa. 2022. US Army OKS acquisition of 5,000 IVAS goggles after year-long delay. https://www.thedefensepost.com/2022/09/05/us-army-ivas-goggles-2/

[60] Jodi Schulz. 20212. *Eye contact: An introduction to its role in communication.* Michigan State University Extension.

[61] Sefik Ilkin Serengil. 2022. *Deep Face Detection with MTCNN in Python.* None. Retrieved Dec 8, 2022 from sefiks.com/2020/09/09/deep-face-detection-with-mtcnn-in-python/

[62] Sefik Ilkin Serengil and Alper Ozpinar. 2021. HyperExtended LightFace: A Facial Attribute Analysis Framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE, IEEE, 445 Hoes Lane, Piscataway, NJ 08854., 1–4. https://doi.org/10.1109/ICEET53442.2021.9659697

[63] Jeff Shepard. 2022. What sensors are used in AR/VR systems? https://www.sensortips.com/featured/what-sensors-are-used-in-ar-vr-systems-faq/

[64] Jiayu Shu, Rui Zheng, and Pan Hui. 2016. Cardea: Context-Aware Visual Privacy Protection from Pervasive Cameras. *CoRR* abs/1610.00889 (2016). arXiv:1610.00889

[65] Julian Steil, Marion Koelle, Wilko Heuten, Susanne Boll, and Andreas Bulling. 2019. PrivacEye: Privacy-Preserving Head-Mounted Eye Tracking Using Egocentric Scene Image and Eye Movement Features. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 26, 10 pages. https://doi.org/10.1145/3314111.3319913

[66] Techjury. 2022. *29+ augmented reality stats to keep you sharp in 2022.* Retrieved Sept 29, 2022 from https://techjury.net/blog/augmented-reality-stats/

[67] Khai N. Truong, Shwetak N. Patel, Jay W. Summet, and Gregory D. Abowd. 2005. Preventing Camera Recording by Designing a Capture-Resistant Environment. In *Proceedings of the 7th International Conference on Ubiquitous Computing* (Tokyo, Japan) *(UbiComp'05)*. Springer-Verlag, Berlin, Heidelberg, 73–86. https://doi.org/10.1007/11551201_5

[68] Unity. 2022. *GameObject.* Unity. Retrieved Oct 16, 2022 from https://docs.unity3d.com/ScriptReference/GameObject.html

[69] Stylianos I. Venieris, Ioannis Panopoulos, and Iakovos S. Venieris. 2021. OODIn: An Optimised On-Device Inference Framework for Heterogeneous Mobile Devices. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*. 1–8. https://doi.org/10.1109/SMARTCOMP52413.2021.00021

[70] Roel Vertegaal, Robert Slagter, Gerrit van der Veer, and Anton Nijholt. 2001. Eye Gaze Patterns in Conversations: There is More to Conversational Agents than Meets the Eyes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 301–308. https://doi.org/10.1145/365024.365119

[71] Rajeev Yasarla, Federico Perazzi, and Vishal M. Patel. 2020. Deblurring Face Images Using Uncertainty Guided Multi-Stream Semantic Networks. *IEEE Transactions on Image Processing* 29 (2020), 6251–6263. https://doi.org/10.1109/TIP.2020.2990354

[72] Yunfan Zhang, Tim Scargill, Ashutosh Vaishnav, Gopika Premsankar, Mario Di Francesco, and Maria Gorlatova. 2022. InDepth: Real-Time Depth Inpainting for Mobile Augmented Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 37 (mar 2022), 25 pages. https://doi.org/10.1145/3517260