Combating the Spread of Coronavirus by Modeling Fomites with Depth Cameras

ANDREW D. WILSON, Microsoft Research

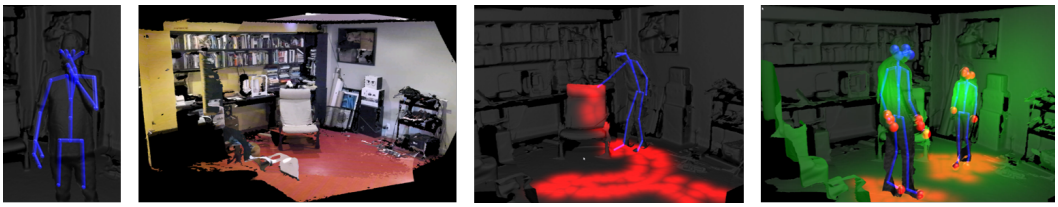

Fig. 1. From left to right: Using Azure Kinect body tracking to detect face touching; 3D reconstruction of room with two calibrated depth cameras; 3D contamination map; rendering of social distance calculations (green) and joints marked as contaminated (red spheres).

Coronavirus is thought to spread through close contact from person to person. While it is believed that the primary means of spread is by inhaling respiratory droplets or aersols, it may also be spread by touching inanimate objects such as doorknobs and handrails that have the virus on it ("fomites"). The Centers for Disease Control and Prevention (CDC) therefore recommends individuals maintain "social distance" of more than six feet between one another. It further notes that an individual may be infected by touching a fomite and then touching their own mouth, nose or possibly their eyes. We propose the use of computer vision techniques to combat the spread of coronavirus by sounding an audible alarm when an individual touches their own face, or when multiple individuals come within six feet of one another or shake hands. We further propose using depth cameras to track where people touch parts of their physical environment throughout the day, and a simple model of disease spread among potential fomites. Projection mapping techniques can be used to display likely fomites in realtime, while headworn augmented reality systems can be used by custodial staff to perform more effective cleaning of surfaces. Such techniques may find application in particularly vulnerable settings such as schools, long-term care facilities and physician offices.

CCS Concepts: • **Human-centered computing → Interactive systems and tools**.

Additional Key Words and Phrases: depth cameras, sensing, augmented reality, coronavirus; COVID-19

ACM Reference Format:
Andrew D. Wilson. 2020. Combating the Spread of Coronavirus by Modeling Fomites with Depth Cameras. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 203 (November 2020), 13 pages. https://doi.org/10.1145/3427331

1 INTRODUCTION

There is still much to learn about how the COVID-19 virus is transmitted. The Centers for Disease Control and Prevention notes that COVID-19 most commonly spreads "Between people who are in close contact with one another (within about 6 feet)," and "through respiratory droplets or small

Author's address: Andrew D. Wilson, Microsoft Research, awilson@microsoft.com.

2573-0142/2020/11-ART203 $15.00
https://doi.org/10.1145/3427331

particles, such as those in aerosols, produced when an infected person coughs, sneezes, sings, talks, or breathes." [4] To prevent the transmission of the virus the CDC recommends maintaining "social distance" of 6ft (1.8m) between individuals. Further, in many public places such as grocery stores, wearing facial masks is strongly encouraged or mandatory.

The CDC further notes that "a person may get COVID-19 by touching the surface or object that has the virus on it and then touching their own mouth, nose, or eyes." It has been observed in the popular press that it can be very difficult to stop touching your face. This is somewhat comically illustrated by numerous recent appearances by public officials recommending to stop touching your face, who in the very same appearance touch their face [28]. By some estimates, people touch their face as many as sixteen times an hour [22]. Additionally, it has been suggested to avoid shaking hands in greeting, and instead "bump elbows."

How long the virus can live on a surface may depend in part on the surface material [11]. One study found that the virus can survive up to three days on hard metal surfaces and plastic and less time on more porous materials [27]. "Fomites" are physical objects in the environment that when contaminated with an infectious agent, can transfer the disease to someone who touches them. Given the lifetime of the virus, it would seem that potential fomites such as doorknobs, light switches, hand rails, shoes, etc., may play an important role in the transmission of the virus, and should be cleaned regularly [6, 10, 35].

Fomites, face touching and difficulty maintaining social distance suggest that once the virus has entered an environment such as an office space, it can quickly spread throughout the environment. YouTube personality Mark Rober performed a compelling and safe demonstration of this with an elementary school class, where he applied an invisible powder to the hands of the teacher, who then shook hands with three students on their way into the classroom [24]. Imaging the powder with a blacklight revealed that the powder spread widely across the classroom over the course of the day. Rober motivates the experiment by noting that people would be much more careful if they could see the germs around them.

Naturally there are those seeking technological solutions to combat the spread of the virus. An interesting early example uses computer vision techniques to raise an alarm when the user of a web camera-equipped computer touches their face [18], while a variety of other sensors have been proposed [1, 5, 19, 26]. In this paper we propose using depth cameras such as the Microsoft Azure Kinect to detect face touching behavior, but to also gain a greater situational awareness of the room, with possibly multiple people, to more generally model the formation of fomites as people move about and touch their environment. Such techniques may find application in particularly vulnerable settings such as schools, long-term care facilities and physician offices.

In September 2020, it is commonly believed that the virus spreads mainly through airborne means rather than by touching surfaces. Attention has shifted from preventing face touching to encouraging the use of face masks and social distancing. Yet in a recent press release the CDC has reiterated that fomite transmission remains a risk for catching COVID-19 [3], while others have noted that surfaces touched by a large number of people, such as door handles and elevator buttons may carry enough copies of the virus to act as fomites [21].

This paper makes the following contributions:

- Detection and alarm of face touching and handshaking behavior among multiple people in a room.
- Detection of probable touching of inanimate objects in the environment, which when accumulated over time, suggest the presence of fomites.

- A simplistic model of virus spreading where the contagion may be "gathered" from a fomite and "scattered" on another body part or part of the environment, leading to the formation of new fomites.
- A computational framework where calculations on fomites are performed on 3D representations of the environment that are easily computed from a network of depth cameras.
- The use of projection mapping techniques to display in realtime the evolution of fomites and social distancing among multiple people.

Given the present difficulty in performing user studies with human subjects, and obtaining IRB approval, the paper focuses instead on describing promising algorithms and strategies to build a realtime system using commonly available hardware. The validation of the system with a live virus is considered well beyond the scope of this work.

2 RELATED WORK

Having considered a number of studies related to coronavirus, we next consider previous works exploring the use of depth cameras. Depth cameras such as the Microsoft Kinect present opportunities for analysis of the physical world, leveraging techniques such as 3D surface reconstruction and realtime person tracking.

Using the Kinect for Xbox 360, Wilson demonstrated using a depth camera as a touch sensor by comparing a live depth image to a stored depth image modeling the static environment, a kind of simple 3D surface reconstruction [30]. This allows detecting touch on surfaces of arbitrary complexity, with the challenge that the accuracy of touch detection is limited by sensor noise and camera sight lines. This approach has since been refined in a number of ways. Xiao, et al., for example, explore combining the depth image with the infrared image to enhance touch accuracy [34].

Depth cameras can also be used to perform a more general 3D reconstruction of the environment, possibly involving moving cameras or multiple cameras. Kinect Fusion leverages a voxel grid representation of the 3D surface to incorporate the observations of a moving depth camera [8]. Holoportation demonstrated the use of custom depth cameras to perform realtime voxel grid reconstruction [23]. The present work performs calculations in their original depth image representations, avoiding the computational expense of a voxel grid representation.

Depth cameras have been used to track people and compute their skeletal pose in realtime. The Xbox 360 Kinect uses inference based on random forests [25], but more recent pose estimation techniques such as OpenPose exploit deep learning to compute 2D pose from color images [2]. The Azure Kinect camera, used by the present work, uses deep learning techniques to infer 3D pose from depth and infrared images [16].

Depth cameras naturally lend themselves to building higher level interpretations of activities in room-scale environments. LightSpace demonstrated the use of multiple depth cameras and projectors to deliver interactive projection mapping experiences [32]. RoomAlive refined the calibration of multiple projectors and depth cameras as well as the rendering of interactive projection mapped experiences [9, 17]. RoomAlive has been shown to scale to as many as eight cameras in a conference room [31]. Lindlbauer, et al., uses a voxel grid representation to encode annotations throughout the physical environment [12]. The Proximtiy Toolkit enables developers to create applications that reason about multiple users and devices in the same space [13], while EagleSense focuses on the problem of inferring intent of multiple users from depth images [33].

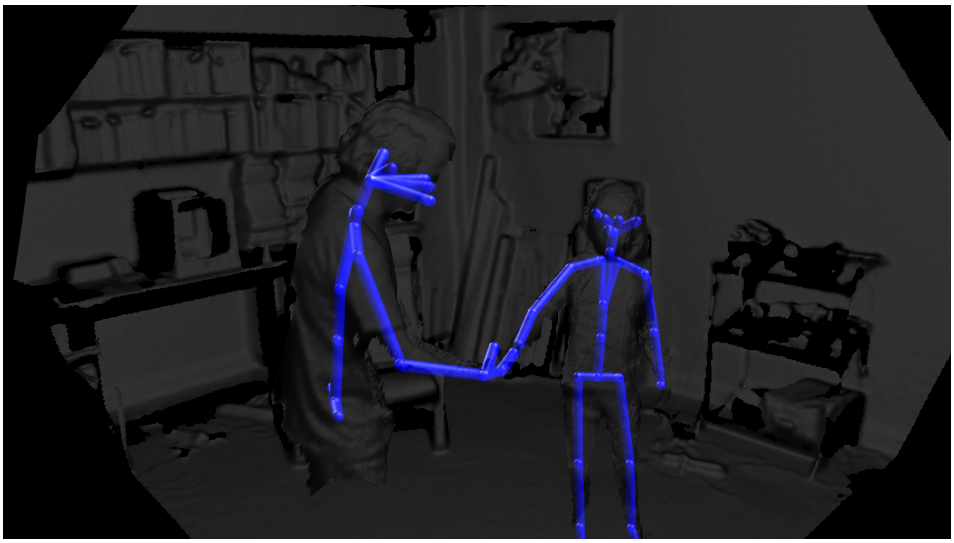

Fig. 2. Two people tracked by the Azure Body Tracking SDK shake hands, rendered over the depth image from a single camera.

3 DESIGN

We base our proposed interactions and visualizations on the capabilities of the Azure Kinect camera and its associated body tracking SDK [14–16]. The Azure Kinect camera includes a color image sensor and a time of flight depth image sensor that is similar to that of the Kinect for Xbox One.

3.1 Face Touching, Shaking Hands and Social Distance

The Azure Kinect body tracking SDK tracks multiple people and the 3D position $\vec{x}_j$ of up to 32 locations ("joints") on each tracked body. Joints include the head, pelvis, nose, hands, wrists, feet, etc. The body tracking SDK also reports a "confidence" value with each joint. A joint that is successfully tracked may have a high confidence value, while a joint that is occluded may have a low value. Unlike with earlier Kinect cameras, there is no specified limit to the number of people that can be simultaneously tracked by the Azure Kinect body tracking SDK.

While it is difficult or impossible to directly observe two objects touching by video cameras [30], a tracked person that is likely about to touch their face can be detected by simply calculating the distance between either hand joint and the nose joint. In our prototype system, an annoying alarm is sounded when this distance falls below a threshold $\tau = 150$mm: $\|\vec{x}_{hand} - \vec{x}_{nose}\| < \tau$. This threshold was tuned empirically to reduce *false positive* and *false negative* touch events (also see Section 5.3). Similarly, two tracked people about to shake hands can be detected by calculating the distance between their hands. See Figures 1, 2, and the accompanying video.

Finally, two people who are not maintaining appropriate social distance can be detected by calculating the distance between the pelvis of the two tracked bodies, the average joint position over each body, or any pair of joints between bodies. In this case it may be appropriate to rely on some means of notification other than an audible alarm, since, while it is easy for people to attribute a short alarm to either touching their face or shaking hands, it is more difficult to associate an alarm to violating an invisible distance threshold. We propose an alternative means of displaying social distance later.

3.2 Touching the Environment

If a 3D point (X, Y, Z) projects to image coordinates (u, v) in the depth camera, its depth Z is recorded in the depth image $\mathcal{D}(u, v) = Z$. The Azure Kinect SDK includes an API to calculate 3D position $\vec{x}_d(u, v) = (X, Y, Z)$ from $Z = \mathcal{D}(u, v)$. Meanwhile, the 3D position $\vec{x}_j$ of a joint j is reported in the depth sensor coordinate system, such that it can be related to geometry and objects that appear in the depth image. Specifically, we can compare joint positions to regions of a stored depth image to determine whether a person is touching or about to touch parts of their physical environment.

A static depth map $\bar{\mathcal{D}}(u, v)$ can be computed offline as the average of multiple depth maps or by other more sophisticated surface reconstruction techniques [8]. Because the body tracking SDK gives no information about the shape of the joint, and because joint position and depth images can be noisy, it is helpful to compare joint position and a point in the depth image under some model of uncertainty. We denote the multivariate normal distribution with a scalar σ as $\mathcal{N}(\vec{\mu}, \sigma^2)$ and compute the likelihood that a joint at $\vec{x}_j$ is currently touching a point $\vec{x}_d(u, v)$ in the depth image $\bar{\mathcal{D}}(u, v)$ as:

$$w_j(u, v) = \mathcal{N}(\vec{x}_d(u, v) - \vec{x}_j, \sigma^2)$$

Such touch events can be accumulated by an image $C(u, v)$ which is updated every frame for all joints that are considered likely to touch the environment (e.g., hands and feet):

$$C'(u, v) = \max\left(C(u, v), w_j(u, v)\right) \tag{1}$$

A high value in $C(u, v)$ encodes our belief that the corresponding region around $\bar{\mathcal{D}}(u, v)$ has been touched at some point in the past. This map of touch history can be of interest when considering likely fomites in the environment (see Figure 1).

3.3 A Simplistic Model of Spreading

Next we present a simplistic model of virus spreading that leverages the calculations developed on joint positions and depth images above[1].

First, all joints of the tracked people in the room are initially marked as contaminated or not. A joint is updated as "contaminated" if it is found to be touching another contaminated joint. For example, a contaminated hand will contaminate the person's nose if they touch their face with their hand, and one person's hand might contaminate another person's hand if they shake hands, as computed in Section 3.1.

The touch history image $C(u, v)$ can model the "scatter" of the virus and ongoing contamination of the physical environment if only contaminated joints are considered in its update (equation 1).

Finally, a joint j that is not marked as contaminated can "gather" the virus through a similar computation:

$$\hat{c}_j = \frac{\sum_{u,v} w_j(u, v) C(u, v)}{\sum_{u,v} w_j(u, v)} \tag{2}$$

The joint is considered contaminated if $\hat{c}_j$ exceeds some threshold.

In this very simple model of virus spreading, once contaminated, joints and regions of the physical environment never become uncontaminated. The contamination map $C(u, v)$ may need to be cleared periodically, or gradually. For example, the lifetime of the virus could be coarsely modeled by having values $C(u, v)$ slowly decay over the span of days. Similarly, a joint marked as a "cleaner" could be used to make local reductions in $C(u, v)$ using an update similar to equation 1.

[1]We warn the reader that this model has not been validated by epidemiologists in any way!

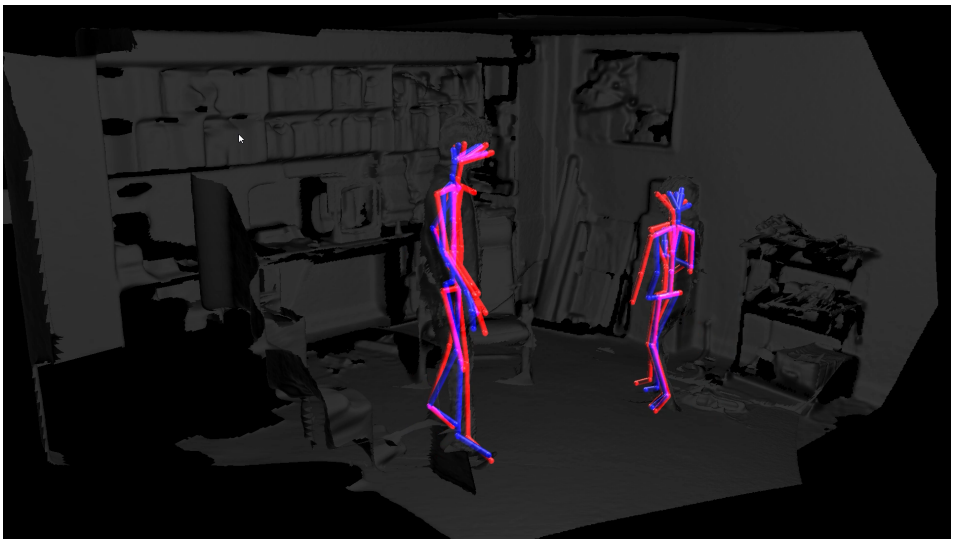

Fig. 3. Bodies tracked from two cameras (red, blue) are merged into a single body (purple) for further processing.

3.4 Multiple Cameras

Thus far we have only considered modeling the physical environment and tracking people's bodies using a single camera. This may be adequate for certain application scenarios, such as detecting when a computer user touches their face as they sit in front of a display, but in larger room-sized applications with perhaps multiple people, the line of sight of a single camera will be quite limiting. The addition of more cameras can address larger rooms or monitor the same critical area from multiple angles.

Multiple cameras can be supported by simply employing a static depth map $\bar{\mathcal{D}}(u,v)$ and contamination map $C(u,v)$ for each camera. In many cases it will be desirable to calibrate the pose of each camera to a common coordinate system. This can be done, for example, by sighting common fiducials in each camera (three will suffice). Knowing the pose of each camera allows transforming a joint position tracked in one camera to another camera's coordinate system. This can be advantageous, for example, when the body is successfully tracked from one camera but one of its joints touches a physical surface visible in another.

Calibrating multiple cameras to the same coordinate system also allows to infer that a body tracked in one camera is the same tracked body in another camera. A body model where multiple observations of the same body are merged into a single model can help in preserving joint contamination state as the body moves in and out of view of multiple cameras. Our prototype implementation merges bodies in a greedy fashion based on similar joint position in the common coordinate frame (see Figure 3). Average joint positions are calculated across all observations, considering only the joints that are marked with higher tracking confidence.

3.5 Projection Mapping

Visualizing the contamination map $C(u,v)$ can give insight into the formation of potential fomites in the physical environment. Hot spots in $C(u,v)$ could be targeted for extra cleaning, for example.

Fig. 4. Projection mapping illustrates areas of the physical environment that have been touched *in situ*. This includes parts of the chair which have been touched by the person's hands, as well as areas of the floor touched by the person's feet.

Because $C(u,v)$ and the depth map $\bar{\mathcal{D}}(u,v)$ are in the same image coordinates, it is trivial to render $\bar{\mathcal{D}}(u,v)$ as 3D geometry with per-vertex coloring drawn from $C(u,v)$.

We demonstrate using projection mapping techniques to display the contamination map *in situ*. Our prototype system uses a modified version of the RoomAlive Toolkit to calibrate two cameras and two projectors. Projection mapping uses a simple "surface shading" model in which the per-vertex colored geometry is rendered as usual but with a graphics view and projection matrix calculated from projector pose, focal length, etc. Unlike "holographic" projection mapping techniques where virtual 3D objects are rendered, this simple technique is not view-dependent, since the rendered geometry and the physical surface coincide (see Figure 4).

The Azure Kinect body tracking SDK reports a "body index" image $\mathcal{B}(u,v)$ that maps depth sensor coordinates (u,v) to a currently tracked body. The body index $\mathcal{B}(u,v)$ can be used to mask out parts of a live depth image that do not fall on the tracked bodies. When rendered as black in projection mapping, this masked live depth image can block projection of parts of $C(u,v)$ that are occluded by the body.

We can also use projection mapping techniques to provide a visual *in situ* notification when social distancing between two people has been violated. When detected, we use the "scatter" operation (equation 1) to construct an instantaneous map $\mathcal{S}(u,v)$ that is analogous to $C(u,v)$, but with a value of σ in proportion to social distancing guidelines. This new map is rendered in combination with the contamination map but with a different color. Finally, we perform the same operation on the corresponding body index masked images, such that, with enough projector coverage, each person violating social distance may see those people that are too close similarly illuminated (see Figure 5).

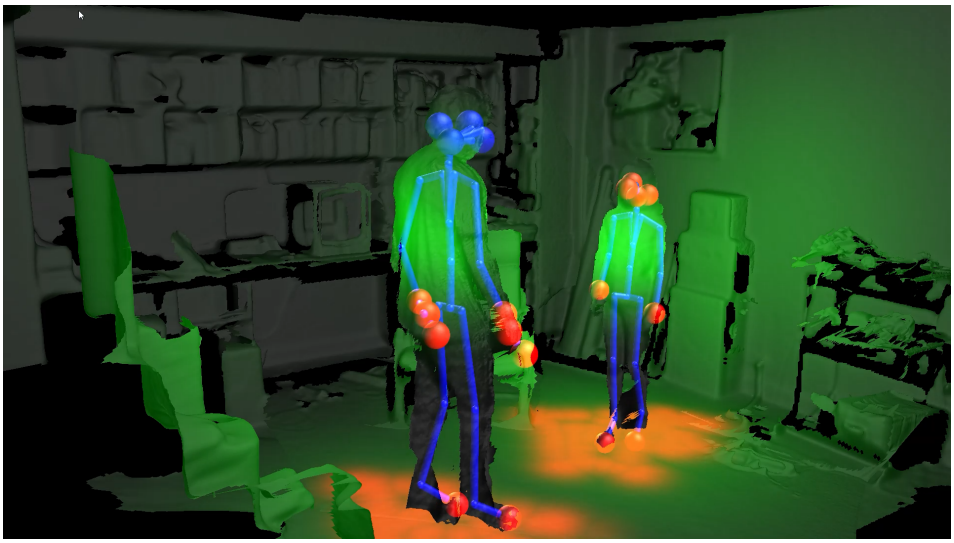

Fig. 5. Rendering of social distance calculation (green). Joints marked as contaminated are rendered as red spheres.

4 IMPLEMENTATION

Our prototype implementation of two Azure Kinect cameras and two projectors is hosted on a single Windows PC with two Nvidia GTX 1080Ti graphics cards. This was constructed in the author's basement during their institution's work-from-home orders.

All software is written in C++ and uses DirectX. Rendering and image operations are hosted on the first GPU (60Hz), while two instances of the Azure Kinect body tracker, one for each camera, are hosted on the second GPU (30Hz). Depth images, contamination maps $C(u, v)$, etc., reside in GPU memory for efficient manipulation and rendering. Deployments of more than two cameras will likely require hosting of individual cameras by remote PCs, as supported by the RoomAlive Toolkit.

Both "scatter" (equation 1) and "gather" (equation 2) operations are implemented as compute shaders. The summations in the numerator and denominator of equation 2 are awkward to compute on a GPU. Our implementation computes them by generating the full set of mip maps of a 2-channel floating point image (one channel for the numerator, one for the denominator). The final division is carried out on the host PC by reading back the last 1×1 level of the mip map and dividing one channel value into the other.

4.1 Computing 3D Coordinates

The Azure Kinect employs a pinhole camera model with radial and tangential lens distortion. With focal length f_x, f_y, principal point c_x, c_y, radial distortion coefficients $k_1, k_2, k_3, k_4, k_5, k_6$ and

tangential distortion coefficients p_1, p_2, a 3D point (X, Y, Z) is projected to an image point (u, v) by

$$
\begin{aligned}
x' &= \frac{X}{Z} \\
y' &= \frac{Y}{Z} \\
x'' &= x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \\
y'' &= y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \\
u &= f_x x'' + c_x \\
v &= f_y y'' + c_y
\end{aligned}
$$

where $r^2 = x'^2 + y'^2$.

The Azure Kinect camera produces a depth image $\mathcal{D}(u, v) = Z$. Calculating (X, Y, Z) from $\mathcal{D}(u, v)$ thus involves inverting the above projection equations, which generally requires iterative optimization (e.g., Newton-Raphson) due to its nonlinearity. However, noting that the lens distortion depends only on x' and y', we can pre-compute tables $x'(u, v)$ and $y'(u, v)$ such that if $Z = \mathcal{D}(u, v)$, then $X = Z \times x'(u, v)$ and $Y = Z \times y'(u, v)$. This yields an efficient means of calculating 3D coordinates from depth image values as a vector product, suitable for use in the "scatter" and "gather" compute shaders.

5 LIMITATIONS

The system presented in this paper has number of important limitations:

5.1 Camera Sight Lines

Computer vision-based approaches have the benefit that they do not require instrumenting the sensed surface, but they suffer from sight line limitations that may prevent the observation of touching certain parts of the environment and from certain body poses. The importance of such *false negative* events will depend on how the overall system is intended to be used. The RoomAlive Toolkit and the Azure Kinect are well-suited to deploying networks of calibrated cameras. We believe that for particularly vulnerable environments, the added infrastructure of multiple cameras may be worth the added expense and complexity. While the prototype presented in the paper uses two cameras, four cameras would be a more effective minimum configuration for a room. As ever with camera-based techniques, privacy will be concern.

5.2 Other Camera Limitations

Camera-based approaches to monitoring large spaces and people have some important limitations that may impact applicability. For example, depth cameras such as the Azure Kinect that use pulsing or structured infrared light typically perform poorly outdoors. Secondly, while person tracking has made great strides in recent years with deep neural networks, there are still configurations that are difficult to correctly track. While the Azure Kinect Body Tracking SDK often correctly tracks seated people, it can fail when the tracked person is partially occluded (e.g., in a bed).

5.3 Touch Precision

The computational model presented in this paper addresses the inherent uncertainty in detecting touch events with depth cameras. This means that the contamination map may in fact indicate

touch events when the contaminating joint only came very near the surface but did not touch. The importance of such *false positive* events will depend on how the overall system is to be used. One approach to improve touch accuracy is to consult the realtime depth map (and possibly body index $\mathcal{B}(u,v)$) to perform touch detection, as in Wilson's work on detecting touch [30]. Even here, the true size of the touching body part must be guessed or inferred. One idea is to fit a more detailed 3D model of the body to the tracked body, and compute intersections with the depth map by rendering the body in the depth camera coordinate frame with "greater than" z-buffering.

5.4 Moving Objects

The present prototype assumes that the physical environment is static and the only moving objects are the people within. For many realistic environments, this assumption will be unreasonable. While there are many obvious techniques to adaptively update the static model as it changes, preserving the information encoded in the contamination map $C(u,v)$ may involve more sophisticated tracking such as optical flow [7].

5.5 Surface Model

The "scatter" and "gather" operations presented model the spread of the virus between two 3D points: the 3D position of a joint, and the 3D position of a given pixel in the depth image. This neglects the shape of the surface around the point in the depth image, which might have some impact on how the virus is spread. One idea is to compute a local planar approximation of the surface and devise a more realistic "scatter" and "gather" operation based on a Gaussian spread around the point of contact with the plane.

5.6 Modeling the Spread of Coronavirus

The spreading model presented assumes that any surface or joint marked as contaminated will spread the virus once it comes in contact with another joint or surface. The probabilistic nature of the model accommodates camera-based systems that are unable to directly observe touch events. This will likely overestimate the spread of the virus. This may be acceptable for many highly critical, vulnerable settings.

However, this model neglects that the primary means of spreading the virus is by respiratory droplets or aerosols caused by coughing, sneezing, talking or breathing [4]. The present work may be extended to detect talking, coughing and sneezing events via the Azure Kinect microphone array, and, given the body tracking information, project the 3D cloud of water droplets and contaminated air onto the physical surface. The 3D position of the dominant sound source was demonstrated as the simple triangulation of sound source angles as reported from three Xbox 360 Kinect sensors [29]. It may be possible to correlate such triangulation results with the body tracking information to determine the position and angle of the event, and then use a "scatter"-like calculation to model the results on the physical environment [20]. Meanwhile, it may be possible to model the continual impact of breathing using the body tracking results.

6 USAGE CONSIDERATIONS

This paper presents a range of technical capabilities that can be combined to meet the needs of a given deployment. An important consideration, for example, is whether it is appropriate to provide real-time feedback indicating when someone is touching their face, shaking hands, touching a fomite or violating social distance. Our current implementation demonstrates using audible alarms and projection mapping to provide real-feedback to people in the room. If the goal is to discourage people from touching their face, it may be desirable, for example, to play an audible alarm even if the false negative rate is relatively high. Such an alarm may be ignored, however, if the false

positive rate is very high, or if there are so many people in the room that it is difficult for an individual to associate the feedback with their actions.

A related concern is whether the system should be used to predict, and therefore prevent, touching behavior before it occurs. Informal experience with our current implementation suggests that its performance is not good enough to predict touch events before they occur. Increasing the distance threshold τ helps anticipate touch events but allows for more false negative events. Such prediction capability might be supported by modeling joint position dynamics.

Some applications may not require real-time feedback. For example, it may be desirable to monitor the evolution of fomites throughout the course of a day, so that custodial staff can later perform more effective, targeted cleaning. In this scenario it may be advantageous for the custodial staff to use a headworn augmented reality display such as the Microsoft HoloLens, which is capable of directly displaying the same 3D contamination map used in the prototype system's projection mapping capability. Such a feature could could be easily implemented using the Holographic Remoting feature of the Windows Holographic API.

"Contact tracing" is thought to be an important way to limit the spread of coronavirus. In applications where it is possible and appropriate to know the identity of the people in the room, the system may support an automatic contact tracing process in which tracked individuals are automatically notified when they come into close contact with other individuals that are known to be sick. In this case it may be possible to estimate the likelihood of spread based on their distance apart, time spent near each other and the overall ventilation of the room.

Furthermore, the notion of "contact tracing" may be extended to include fine-grained interactions between people such as shaking hands, and touching fomites. Specifically, if a tracked body can be associated with a person's real world identification when they enter the room, interactions with other persons may be observed by the system and incorporated in their contact tracing record. Further, fomites in the environment can be associated with a person by recording the person's identification in a map that is analogous to the system's contamination map $C(u, v)$. A person's contract tracing record could then include the fact that they touched a fomite previously touched by someone with the virus. We leave this for future work.

7 CONCLUSION

This paper presents a suite of techniques that aim to leverage depth cameras such as the Azure Kinect to combat the spread of the COVID-19 virus. These range from simple calculations on the Azure Kinect body tracking results to determine whether a person is about to touch their face or shake hands with another, to a more general model of how fomites evolve in the physical environment, and possibly spread the virus as imaged by multiple depth cameras and displayed by multiple projectors. The paper provides technical detail to allow practitioners to replicate the system, and we plan on open-sourcing our implementation if there is sufficient interest.

Such techniques may find application in particularly vulnerable settings such as schools, long-term care facilities, physician offices, and other venues where the cost of installing a network of cameras or projectors may be negligible compared to the cost of the potential loss of life. Without a proper validation, readers should be cautioned against drawing conclusions as to how effective such techniques are in preventing virus transmission (and we note the challenge for most institutions in making any claim of efficacy in this context). Rather, we hope that they inspire further research into effective technological solutions in these troubling times.

REFERENCES

[1] Arduino Team. 2020. This pair of Arduino glasses stops you from touching your face. https://blog.arduino.cc/2020/03/10/this-pair-of-arduino-glasses-stops-you-from-touching-your-face/.

[2] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[3] Centers for Disease Control and Prevention. 2020. CDC updates COVID-19 transmission webpage to clarify information about types of spread. https://www.cdc.gov/media/releases/2020/s0522-cdc-updates-covid-transmission.html.

[4] Centers for Disease Control and Prevention. 2020. How COVID-19 Spreads. https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-covid-spreads.html.

[5] Xiang 'Anthony' Chen. 2020. FaceOff: Detecting Face Touching with a Wrist-Worn Accelerometer. arXiv:2008.01769 [cs.HC]

[6] Zhen-Dong Guo, Zhong-Yi Wang, Shou-Feng Zhang, Xiao Li, Lin Li, Chao Li, Yan Cui, Rui-Bin Fu, Yun-Zhu Dong, Xiang-Yang Chi, et al. 2020. Aerosol and surface distribution of severe acute respiratory syndrome coronavirus 2 in hospital wards, Wuhan, China, 2020. *Emerg Infect Dis* 26, 7 (2020).

[7] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. 2012. HoloDesk: Direct 3d Interactions with a Situated See-through Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2421–2430. https://doi.org/10.1145/2207676.2208405

[8] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew Fitzgibbon. 2011. KinectFusion: Real-Time Dynamic 3D Surface Reconstruction and Interaction. In *ACM SIGGRAPH 2011 Talks* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 23, 1 pages. https://doi.org/10.1145/2037826.2037857

[9] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 637–644. https://doi.org/10.1145/2642918.2647383

[10] Alicia NM Kraay, Michael AL Hayashi, Nancy Hernandez-Ceron, Ian H Spicknall, Marisa C Eisenberg, Rafael Meza, and Joseph NS Eisenberg. 2018. Fomite-mediated transmission as a sufficient pathway: a comparative analysis across three viral pathogens. *BMC infectious diseases* 18, 1 (2018), 540.

[11] Mary Y. Y. Lai, Peter K. C. Cheng, and Wilina W. L. Lim. 2005. Survival of Severe Acute Respiratory Syndrome Coronavirus. *Clinical Infectious Diseases* 41, 7 (10 2005), e67–e71. https://doi.org/10.1086/433186 arXiv:https://academic.oup.com/cid/article-pdf/41/7/e67/881171/41-7-e67.pdf

[12] David Lindlbauer and Andy D. Wilson. 2018. Remixed Reality: Manipulating Space and Time in Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 129, 13 pages. https://doi.org/10.1145/3173574.3173703

[13] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. 2011. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 315–326. https://doi.org/10.1145/2047196.2047238

[14] Microsoft. 2020. About Azure Kinect Sensor SDK. https://docs.microsoft.com/en-us/azure/kinect-dk/about-sensor-sdk.

[15] Microsoft. 2020. Azure Kinect DK. https://azure.microsoft.com/en-us/services/kinect-dk/.

[16] Microsoft. 2020. Quickstart: Set up Azure Kinect body tracking. https://docs.microsoft.com/en-us/azure/kinect-dk/body-sdk-setup.

[17] Microsoft Research. 2015. RoomAlive Toolkit. https://github.com/Microsoft/RoomAliveToolkit.

[18] Brian Moore Mike Bodge and Isaac Blankensmith. 2020. https://donottouchyourface.com/.

[19] MIT Media Lab. 2020. Project Saving Face. https://www.media.mit.edu/projects/saving-face/overview/.

[20] New York Times. 2020. This 3-D Simulation Shows Why Social Distancing Is So Important. https://www.nytimes.com/interactive/2020/04/14/science/coronavirus-transmission-cough-6-feet-ar-ul.html.

[21] New York Times. 2020. What's the Risk of Catching Coronavirus From a Surface? https://www.nytimes.com/2020/05/28/well/live/whats-the-risk-of-catching-coronavirus-from-a-surface.html.

[22] Mark Nicas and Daniel Best. 2008. A Study Quantifying the Hand-to-Face Contact Rate and Its Potential Application to Predicting Respiratory Tract Infection. *Journal of Occupational and Environmental Hygiene* 5, 6 (2008), 347–352. https://doi.org/10.1080/15459620802003896 arXiv:https://doi.org/10.1080/15459620802003896 PMID: 18357546.

[23] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. 2016. Holoportation: Virtual 3D Teleportation in Real-Time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing

Machinery, New York, NY, USA, 741–754. https://doi.org/10.1145/2984511.2984517
[24] Mark Rober. 2020. How To See Germs Spread (Coronavirus). https://www.youtube.com/watch?v=I5-dI74zxPg.
[25] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. 2013. Real-Time Human Pose Recognition in Parts from Single Depth Images. *Commun. ACM* 56, 1 (Jan. 2013), 116–124. https://doi.org/10.1145/2398356.2398381
[26] TechCrunch. 2020. Immutouch wristband buzzed to stop touching your face. https://techcrunch.com/2020/03/09/dont-immutouch/.
[27] Neeltje van Doremalen, Trenton Bushmaker, Dylan H. Morris, Myndi G. Holbrook, Amandine Gamble, Brandi N. Williamson, Azaibi Tamin, Jennifer L. Harcourt, Natalie J. Thornburg, Susan I. Gerber, James O. Lloyd-Smith, Emmie de Wit, and Vincent J. Munster. 2020. Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1. *New England Journal of Medicine* 382, 16 (2020), 1564–1567. https://doi.org/10.1056/NEJMc2004973 arXiv:https://doi.org/10.1056/NEJMc2004973
[28] Washington Post. 2020. Officials keep warning the public not to touch their faces – and then do just that. https://www.youtube.com/watch?v=mA1wqjaeKj0.
[29] Andrew Wilson, Hrvoje Benko, Shahram Izadi, and Otmar Hilliges. 2012. Steerable Augmented Reality with the Beamatron. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 413–422. https://doi.org/10.1145/2380116.2380169
[30] Andrew D. Wilson. 2010. Using a Depth Camera as a Touch Sensor. In *ACM International Conference on Interactive Tabletops and Surfaces* (Saarbrücken, Germany) *(ITS '10)*. Association for Computing Machinery, New York, NY, USA, 69–72. https://doi.org/10.1145/1936652.1936665
[31] Andrew D. Wilson. 2017. Fast Lossless Depth Image Compression. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 100–105. https://doi.org/10.1145/3132272.3134144
[32] Andrew D. Wilson and Hrvoje Benko. 2010. Combining Multiple Depth Cameras and Projectors for Interactions on, above and between Surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 273–282. https://doi.org/10.1145/1866029.1866073
[33] Chi-Jui Wu, Steven Houben, and Nicolai Marquardt. 2017. EagleSense: Tracking People and Devices in Interactive Spaces Using Real-Time Top-View Depth-Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3929–3942. https://doi.org/10.1145/3025453.3025562
[34] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (Niagara Falls, Ontario, Canada) *(ISS '16)*. Association for Computing Machinery, New York, NY, USA, 85–94. https://doi.org/10.1145/2992154.2992173
[35] Jijun Zhao, Joseph E Eisenberg, Ian H Spicknall, Sheng Li, and James S Koopman. 2012. Model analysis of fomite mediated influenza transmission. *PloS one* 7, 12 (2012).

Received June 2020; revised August 2020; accepted September 2020