Reducing the Latency of Touch Tracking on Ad-hoc Surfaces

NEIL XU FAN, The University of British Columbia, Canada
ROBERT XIAO, The University of British Columbia, Canada

Touch sensing on ad-hoc surfaces has the potential to transform everyday surfaces in the environment - desks, tables and walls - into tactile, touch-interactive surfaces, creating large, comfortable interactive spaces without the cost of large touch sensors. Depth sensors are a promising way to provide touch sensing on arbitrary surfaces, but past systems have suffered from high latency and poor touch detection accuracy. We apply a novel state machine-based approach to analyzing touch events, combined with a machine-learning approach to predictively classify touch events from depth data with lower latency and higher touch accuracy than previous approaches. Our system can reduce end-to-end touch latency to under 70ms, comparable to conventional capacitive touchscreens. Additionally, we open-source our dataset of over 30,000 touch events recorded in depth, infrared and RGB for the benefit of future researchers.

CCS Concepts: • **Human-centered computing** → **Interaction techniques**.

Additional Key Words and Phrases: touch detection, latency reduction, ad-hoc surfaces

ACM Reference Format:
Neil Xu Fan and Robert Xiao. 2022. Reducing the Latency of Touch Tracking on Ad-hoc Surfaces. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 577 (December 2022), 11 pages. https://doi.org/10.1145/3567730

1 INTRODUCTION

Touch interfaces are now ubiquitous thanks to the prevalence of smartphones in everyday life. However, touch interactions are still largely confined to small devices which are suitably instrumented with touch sensing capabilities; large touch interfaces, such as television-sized or wall-sized touch interfaces are still rare, in part due to the cost and complexity of adding touch sensing instrumentation over a large surface area. To counter this, researchers have proposed various strategies for performing touch sensing over large, uninstrumented surfaces, enabling ordinary surfaces in the environment to be used as tactile touch interfaces.

One of the most promising touch sensing techniques is the use of a depth camera sensor to sense the position of the fingertip relative to the surface [29]. However, due to the long capture and processing times of the depth sensor, depth-camera based touch sensing systems typically suffer from high touch latency. Furthermore, the low depth resolution of contemporary depth sensors render them unable to accurately discern the fingertip's depth near the touch surface, resulting in low touch detection accuracy.

In this work, we tackle these twin issues of latency and touch detection accuracy by using a model pipeline which exploits the dynamic features of the fingertip at touch moments. This approach improves on both latency and touch event detection accuracy as compared with prior works. Additionally, we contribute a dataset of raw depth + infrared + RGB videos of touch events collected

Authors' addresses: Neil Xu Fan, fanxu104@cs.ubc.ca, The University of British Columbia, Department of Computer Science, Vancouver, BC, Canada; Robert Xiao, brx@cs.ubc.ca, The University of British Columbia, Department of Computer Science, Vancouver, BC, Canada.

2573-0142/2022/12-ART577 $15.00
https://doi.org/10.1145/3567730

under a variety of situations, referenced to a ground-truth capacitive touch sensor underneath the touch surface, to aid future researchers in improving the performance of touch interactions on arbitrary surfaces.

2 RELATED WORK

Touch sensing on minimally-instrumented or uninstrumented surfaces has a long history in the research literature. For example, touch can be detected by using inertial measurement units or acoustic sensors on the user's fingers [27] [12], wrists [21], or on the surface itself [10] to sense the vibrations caused by the contact of the finger with the surface. Instrumenting the periphery of the surface can also be used for touch sensing; for example, infrared sensors can be placed along the periphery, or the surface can be coated with a conductive paint and surrounded by electrodes on the periphery for electrical sensing of touches [34].

2.1 Vision-Based Touch Sensing

More directly related to our technique are systems which employ cameras for touch sensing.

The color change of the fingernail when pressing on a surface can be used to identify a touch event [19, 26]. However, this method relies on touch pressure and stable lighting conditions. Tsuji et al. [28] introduced a novel approach using a projector as a programmable light source to capture objects close to the surface by Slope Disparity Gating.

Vision-based methods can capture not only the finger but also the casted shadows. Niikura et al. [23] demonstrated a method to detect touch when the finger and its shadow converge together. Matsubara et al. [20] built a prototype system with two IR lights at the sides of the operating region. The distance of the hand to the surface is estimated by measuring the distance of the shadows cast by the IR lights. Echtler et al. [7] built a multi-touch table based on FTIR (frustrated total internal reflection) to detect touch and hover.

A thermal camera can also be used to sense touch, by detecting the temperature change left on the surface after touching [15, 16]. However, this method only senses touch after removing the finger, resulting in high latency. LIDAR can also be used to detect the object near a surface. Digital Playgroundz [11] attaches a LIDAR sensor onto a wall to detect objects touching the wall. SurfaceSight [17] attaches to the base of IoT devices to sense objects touching the surface.

2.2 Depth Camera-Based Accuracy

The most common method is to use a depth camera, which senses the distance to each pixel in the field-of-view.

Wilson first demonstrated a method of using a Microsoft Kinect depth camera as a touch sensor, which detects a touch if the distance from the finger to the surface is smaller than a threshold value [29]. Since the introduction of this method, accuracy has been an active research area. Touch accuracy consists of *location accuracy* and *event-detection accuracy*.

Location accuracy is well explored. DIRECT [31] merges depth and infrared data in order to resolve the position of the fingertips more precisely than prior touch tracking methods, which keeps the average error within 5mm. MRTouch [32] extends this technique to the depth camera built into a head-mounted augmented reality device. Both DIRECT and MRTouch were optimized for touch position accuracy, and visually suffer from high latency, with MRTouch citing a latency of 180ms. Additionally, both systems have high error rates, with DIRECT undercounting multitouch contacts 22% of the time, and MRTouch overcounting contacts 21% of the time.

However, event-detection accuracy, in our opinion, is not sufficiently studied. Many works used the single threshold method, as Wilson did, in their studies [4, 6, 9, 22]. Due to noise from depth cameras, this method suffers from false positives. An improved method is to use a pair of thresholds

with hysteresis; see Section 5.1 for a more detailed description. This is currently the most popular method used in research with depth cameras [1, 33].

2.3 Latency Reduction

Researchers have investigated the effects of latency on touchscreen applications, as well as ways to reduce latency. Hinckley et al. utilize low-level capacitive information from a touchscreen to detect fingers hovering over a touchscreen, in order to predict touch down events before they happen [14]. Cattan et al. used short-term finger movement prediction to reduce touch latency on touch devices [2]. HACHIStack uses a two-layer photo-sensor-based structure to measure the velocity of a finger when it is close to the surface [13], enabling the system to predict the touch event before the touch physically occurs. Predicting in advance of the physical event allows the system to compensate for latency occurring later in the input pipeline.

Many researchers explored the possibility of predicting the touch using the finger's trajectory. Zero-Latency Tapping uses a Vicon system to track the users' fingertip and use the fingertip trajectory to predict the location and time for tapping [30]. Similarly, Cattan et al, localize the fingertip by optical tracking system with a marker. With continuous prediction of the touch point they can reduce the latency of touch interactions [3]. However, a vision tracking system with instrumented fingertips is not widely usable in normal life.

Another method is adding a color camera. Hybrid HFR Depth combines the ordinary color camera with the depth camera to synthesize low-latency and high-frame rate depth images [18]. Song et al. puts a RGB camera on one side of an interactive display and uses a convolutional neural network to predict the location of the touch and reduce the latency of a touch screen [25].

3 DATA COLLECTION

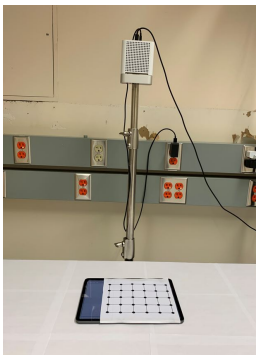
We take a data-driven machine learning approach to predicting the moment of touch. We recruited 16 participants to our initial data collection and released the dataset for future researchers to use[1].

3.1 Data Collection Setup

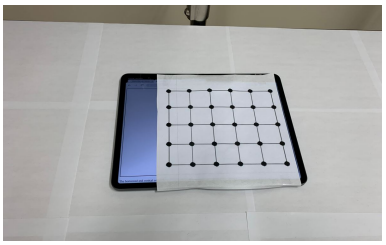
A Microsoft Azure Kinect DK depth camera is used to capture the users' hand movement. During the data collection session, we used this camera to record the RGB, infrared and depth video of users' hand movement. The videos were captured at 30 fps. A 12.9-inch iPad Pro is used to collect the users' touch interactions. To better simulate a surface, the iPad screen is covered with a thin sheet of white printer paper and attached with tape at the edge, which does not affect the capacitive touch sensor. Unlike the glass surface of the iPad, the printer paper exhibits specular and textural qualities more similar to ordinary surfaces such as wood. The cover paper has a grid of dots printed on it as a guide. Participants are asked to tap the dots one by one to get an even and widely spread distribution of the location of touches.

The iPad runs a full-screen application which detects and records the touch interactions on the screen to a file. The recorded data file contains the class of the touch event (e.g. touch down, touch move, or touch up), screen coordinates, touch identifier and timestamp. The screen turns from white to black whenever it detects a touchdown and turns back to white upon touch up. This color change information can be used to synchronize the data recorded from the iPad with the camera.

[1]The data is released at: https://github.com/UBC-X-Lab/DepthTouchSensing



(a) The camera setup.



(b) The iPad setup.

Fig. 1. Data collection experiment setup.

3.2 Data Collection Procedure

The data collection session is planned in a way to capture varying touch conditions. This dataset covers tapping, dragging, and multi-touch gestures, although only tapping data is used directly in our system.

16 participants (10 female) were recruited for this data collection session. Each participant was paid $15 for an 80 minute long session. Participant skin colours ranged from Type I to Type V on the Fitzpatrick scale [8] (I: 3; II: 5; III: 4; IV: 3; V: 1). They were randomly assigned to 4 balanced groups (4 people per group). The camera was set up at a different height and angle for each group. Within each group, the iPad is set up at a different location for each participant, to cover a larger touch area. All the participants were asked to do the same task during the study. The data was collected in a room with constant indoor artificial light condition (about 200 lux).

Each participant recorded 6 sessions. Since tapping is the most common touch interaction, 4 sessions were designed to collect tapping data. Each tapping session is broken into 4 subsections. In these 16 subsections, 3 tapping conditions are explored: different tapping speed (participants are asked to follow a metronome played at 80 vs. 120 taps per minute), different finger lifting distance (1-2 vs. 7-8 cm), and different angle between finger and surface (30 vs. 60 degrees). For the finger lifting distance and finger angle, participants are given a chance to practice before the data collection. The tapping conditions are constantly monitored by the researcher. A reminder would be given to the participants when the condition is not met. Each condition has two levels, and each combination of conditions is recorded twice ($2 \times 2 \times 2 \times 2 = 16$). In total, each participant collected on average 2400 taps. The remaining two sessions recorded dragging and multi-touch interactions. In the dragging session, participants were asked to draw lines in different directions. In the multi-touch session, participants were asked to perform different multi-touch gestures including pinches and multi-finger taps. The dragging and multi-touch sessions are inserted between the tapping sessions, as sessions 2 and 5 respectively, to break the tapping pattern and reduce monotony. Dragging and multi-touch data were collected and provided for future research, but were not used in this project.

4 METHODS

As mentioned before, the touch sensing problem can be split into touch location and touch moment detection sub-problems. The former sub-problem is well explored in previous studies (2.2). However, very few works addressed the latter problem of detecting the touch moment accurately, which

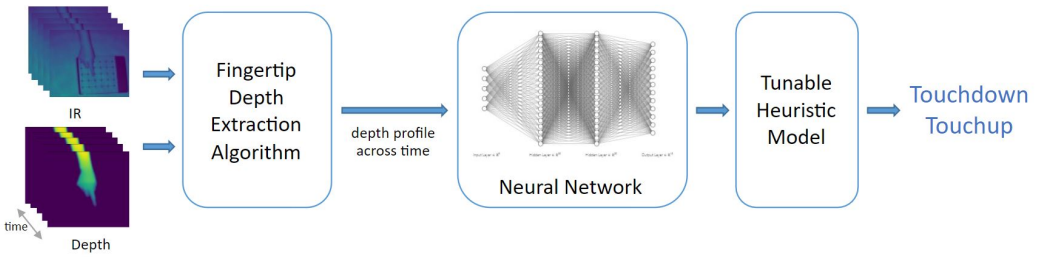


Fig. 2. The model pipeline.

we focus on. Our touch moment detector is inspired by the idea of using inertial measurement units (IMUs) to detect the finger's approach towards the surface and abrupt stop when the finger contacts the surface. Instead of using heuristics to capture the fingertip dynamic changes, we take a data-driven machine learning approach.

The system tracks touches in three stages, as shown in Figure 2. In the first stage, candidate fingertip locations are extracted from the synchronized infrared and depth frames. In the second stage, a neural network is used to capture and classify the dynamics (depth, speed, acceleration) of a fingertip. As mentioned in Section 2.2, previous works only used the depth information. We believe that capturing the fingertip velocity change will be beneficial to identify the touch moment. In the final stage, the current and past output of the classification network is fit into an adjustable heuristic model to identify the touch moment.

4.1 Fingertip Depth Profile Extraction

We adapted a similar method as DIRECT [31] to capture the accurate location of the fingertip. The first step is to remove the background from both depth and IR images. When the system starts up, we extract the first two seconds of the depth and IR video and compute the mean and standard deviation of each pixel. For each incoming depth image, the background will be removed by zeroing pixels which differ from the background by less than two standard deviations (95% confidence level).

After removing the background, we generate binary masks for the depth and IR images by thresholding. Then, the IR mask is fused with the depth mask using bitwise-OR, allowing the depth image to fill in the area where the hand shares a similar IR profile as the background. Additionally, the IR image can mitigate multipath interference in the depth image which manifests as a "denting" of the observed surface near a finger (see e.g. [24] for more details).

To locate the finger, we assume that the arm intersects the edge of the image, and identify the center point of this intersection. We then identify the furthest masked point from this edge point, which is assumed to be the outstretched fingertip of the user. To smooth out noise, we averaged nonzero values around the detected fingertip with a radius of 10 pixels. We accumulate the averaged depth value of the fingertip in each frame to generate the depth curve across time.

4.2 Classification Network

4.2.1 Data synchronization. For training and evaluation, we need to synchronize the video data with the ground truth collected from the iPad. The signal used for synchronization is the screen color change mentioned in section 3.1. After synchronization, the ground truth touch down and touch up event times are converted to the nearest depth camera frame numbers. Since the Azure

Kinect camera runs at 30 fps and the iPad has a 120 Hz refresh rate, there could be a synchronization error up to 15 ms, which we treat as negligible relative to the Kinect frame time (30 ms).

4.2.2 Frame labeling. We treat the touch interaction as a state machine, where the touch progresses through one frame per state, and certain states correspond to the touch down or touch up event. 14 state labels are designated for the touch process. State 7 is the label for the ground truth touch down event. Nearby frames (six frames before and two after) are labelled as 1-9 in sequence - for example, frames that occur six frames (200 ms) prior to a touch down event will be labelled 1. Similarly, state 12 labels the ground truth touch up event, and the nearby frames (two before and two after) are labelled 10-14. In the event of an overlap, the temporally closer ground truth event has priority. The asymmetry in state labelling reflects the greater priority of touch down events. Thus, the nine frames near the touch down event will be labelled 1, 2, 3, 4, 5, 6, 7 [touch down event], 8, 9 in sequence; frames near the touch up event will be labelled 10, 11, 12 [touch up event], 13, 14.

Frames that are not near any touch event (i.e. no finger or finger moving on the surface) are very numerous, and were not used for training to avoid severe class imbalance. At inference time, these frames are typically predicted as states 1 or 14.

Using our state-based approach, we cast the problem of detecting or predicting a touch as simply predicting the label for a touch point, given data from the preceding frames. A prediction of label 2, for instance, indicates an estimate that a touch down event will occur 5 frames in the future, whereas a prediction of label 7 indicates that the finger has contacted the surface in this frame.

There are two main reasons to treat the touch procedure as having multiple states, instead of just having touch-down and down-up. The first is that the sequential relationship between states can help with improving touch detection accuracy. Even if the exact touch event label is missed, we can still infer the touch by seeing multiple labels adjacent to the exact touch event label - for instance, seeing predicted states 6 and 8 (one frame before/one frame after) allows us to infer a touch down event. The second reason is that having multiple states allows us to predict the touch events by declaring a touch event at an earlier state (e.g. in state 4, which predicts a touch event within three frames), which can be used to lower the latency.

4.2.3 Model selection and training. One of the key ideas of this work is to exploit motion dynamics such as the velocity and acceleration of the fingertip to predict touch motion. We tried applying heuristics to the dynamics, as well as SVM, neural network, RNN, and LSTM models, and found that a simple NN is sufficient to capture the dynamics and shows the best result without overfitting or excessive training requirements. Our model consists of a lightweight classification neural network, with 2 hidden layers. The input dimension is 5, which is the length of the time window (about 150ms). This time window is just enough to capture the dynamics before a touch event. We also explored a time window width of 10, but found that it provided no performance benefit while being slower to train due to added parameters. The output dimension is 14, corresponding to the number of labels. The dimension for both the hidden layers is chosen to be 20, with 740 learnable parameters in total. The activation layer is ReLU. The model was trained with a batch size of 2000, a learning rate of 0.0001, with the Adam optimizer, for 100 epochs on a NVIDIA GeForce RTX 2060 SUPER GPU.

4.3 Tunable Heuristic Model

To convert the state labels into a touch classification, we defined a tunable heuristic model. The Basic Heuristics model produces a down or up event if it sees a ground truth event frame label (7 or 12). It also produces a down event if it saw label 5 or 6 in the previous frame and label 8 in the current frame. This Basic Heuristic model can be tuned by shifting all the label heuristics forward to make early predictions (e.g. by predicting a touch when it sees label 6 to predict one frame in

Table 1. Touch event accuracy (F-scores) and touchdown latency.

Models	Td	Tu	Td latency in frames (ms)
Basic Heuristic	0.9300	0.9306	-0.18 (−5.94 ms)
-1 Heuristic	0.9018	0.9108	-1.09 (−35.97 ms)
-2 Heuristic	0.8779	0.9007	-2.05 (−67.65 ms)
-3 Heuristic	0.8253	0.8922	-3.13 (−103.29 ms)
-4 Heuristic	0.7150	0.8673	-3.84 (−126.72 ms)
Threshold	0.8927	0.8867	-0.71 (−23.43 ms)
Hysteresis	0.9006	0.8884	-0.50 (−16.50 ms)

advance). All models employ a simple state machine to avoid triggering multiple down events with no intervening up events, or vice versa.

5 RESULT AND DISCUSSION

5.1 Comparison Model

As mentioned in Section 2.2, there are two methods for detecting touch events which almost all previous depth camera-based works used. We compare our result against them.

The first one is single thresholding. This method reports a touchdown if it detects the fingertip is below a threshold. A touch up will be reported if the fingertip is higher than the threshold. We built a system following the description of [29]. To maximize the performance of this method, we optimized the threshold using a grid search, with a step width of 0.3 mm around the average depth value of the fingertip between the ground truth touch down and touch up moments.

The second method is a double-threshold method, also known as hysteresis. A touch-down event is identified when the finger is below the lower threshold, and is held until the finger rises above the higher threshold, preventing rapid state changes from noise near a single threshold. We built this system following the description of [33]. We optimized these thresholds using a 10×10 grid search, with a step width of 0.3 mm, near the average depth value of the fingertip at the ground truth touch down and touch up moments.

5.2 Measurement

We are using two main factors to measure the performance of a touch model: touch event detection accuracy and latency. Accuracy is measured by computing the F-score. A false negative is reported if there are no predicted touch events within 7 frames before or 5 frames after a ground truth label. Note that the fastest tap speed in our dataset is 2 taps per second, so taps are typically separated by at least 15 frames. Likewise, a false positive is reported if there are no ground truth touch events near a prediction. A true positive is reported when a prediction shows up near the ground truth. The latency is measured by computing the average frame difference between the matched ground truth and predicted touch event.

To show the model's cross-user performance, we performed a leave-one-user-out cross validation evaluation. All results are averaged over all sixteen folds.

In Table 1, Td and Tu stand for touch down and touch up accuracy (F-scores). The Basic Heuristic model is the proposed touch down model with no prediction. -N Heuristic models refer to models which predict the touch down event N frames in advance (N = 1, 2, 3, 4). All heuristic models use the same touch up detection model, but touch up accuracy can be affected if the corresponding touch down event is not detected. The final column, Td latency, displays the average difference

between the model's predicted touch moment and the ground truth touch moment in units of depth camera frames (33 ms per frame); a negative touch down latency indicates an early prediction.

Our Basic and -1, -2 Heuristic models outperform the Threshold and Hysteresis methods in terms of average accuracy, with accuracy dropping for the more aggressive heuristic models. In general, our system outperforms prior systems; for example, both DIRECT and MRTouch significant rates of touch errors (over 20% in certain conditions), although no F-scores were reported.

We ran a paired t-test on the touch down accuracy of each camera position group against all the data. None of the differences can be considered statistically significant, which suggests that the system is tolerant to different camera physical positions.

5.3 Latency-accuracy Trade-off

That depth camera systems suffer from high latency is a well-known problem. One of the main goals of this project is to predict the touch down event earlier to compensate for this latency. Since our system can treat the touch procedure as a state machine, we can tune our system to predict touch down events before they happen. Because this is a prediction, the earlier we predict, the less confident the system will be. In Table 1, we explore the latency-accuracy trade off for our method.

To assess the effect of prediction on end-to-end latency, we ran the -3 heuristic system in real time and set up a slow-motion camera (iPhone Xs) at 240 fps to capture both the fingertip from the side and the computer monitor. Latency in the system comes from the depth camera, image processing delay, prediction algorithm, and computer display latency. The display is a standard LCD computer monitor with 60 Hz refresh rate and 5 ms response time. The procedure is illustrated in the Video Figure.

We measured the latency by counting the number of frames between the real touch down and the computer display's response. We measured 20 touchdown events for both the threshold method and the earlier heuristic. The threshold method shows an average latency of 237.5 ms, with a standard deviation of 35 ms. Our predictive heuristic method produces an average latency of just 69.2 ms, which makes it competitive with the 50 ms level of an Apple iPad capacitive screen [5]. However, the standard deviation is quite high, about 80 ms. This is because it is difficult for the system to make a stable prediction at a specific time before the touchdown. Having an unpredictable latency reduction is a limitation of the system. However, across the 20 measurement sample, the worst (temporally latest) prediction still produced a 41ms latency reduction compared to the average thresholding method.

Another interesting result is that the latency reduced in the real-time measurement (about 170 ms) is much higher than the result shown in Table 1 (about 100 ms). An explanation for this is that the latency reduction in table 1 is an average improvement among all tapping conditions, including some difficult situations (e.g. small movement and fast tapping speed) where the system cannot predict as far in advance. By contrast, in the brief real-time test, the taps were generally slower, making it easier to predict the tap in advance.

6 LIMITATIONS

6.1 Background IR Profile

Although we fuse the information from both the depth and IR camera, we still rely on the IR image seeing the accurate location of the fingertip to remove the depth image fingertip denting effect. If the users' fingertip shares a very similar IR profile with the background, we will not be able to capture its accurate location. Then, the performance of the system might be greatly jeopardized.

6.2 Limited Testing Conditions

Our system was tested in laboratory conditions with consistent indoor lighting, which may not reflect all possible use-cases. Since our system uses active infrared sensing exclusively, we expect it will be portable to a wide range of indoor lighting scenarios where infrared illumination is sparse, but system performance outdoors may not be consistent.

The touch surface, a piece of paper covering an iPad, provides an IR and depth profile comparable to that of a wooden table, however, it may not reflect the range of table materials used in the wild (e.g. vinyl, melamine, particleboard, granite, etc.). Performance on different materials may be subject to the infrared reflectivity of the surface; surfaces which are very infrared-absorbing may result in poor system performance. Anecdotally, we did not observe significant performance differences between the controlled iPad setup and touching directly on the table surface.

6.3 Noise Level

One large factor that affects the accuracy is the noise level. In the small finger movement session, some participants only lifted the fingertip up by about 5 mm, while the depth camera has a depth accuracy of 1 mm. It is difficult to further reduce the noise level under this fingertip extraction method. Thus, detecting small distance taps is a very challenging task for this system. In future work, we could explore the use of a CNN-based model to classify sequences of touch images instead of the averaged depth value, improving noise robustness.

7 CONCLUSION

In this paper, we proposed a lightweight touch detection model based on the users' fingertip kinetics to detect touch events on non-instrumented surfaces. Although there is a trade-off between accuracy and latency reduction, the system shows an improvement on both accuracy and latency compared to previous systems, improving the practical usability of depth camera based touch tracking systems. Our work also highlights areas of future improvement: broader testing with a wider range of materials, environments and conditions would improve the robustness and applicability of the system, while more sophisticated CNN models may be able to further extract the touch signal from the noise. Although our work focused primarily on detecting the precise moment of touch, it could potentially be applied to improving touch location detection in the future. Overall, our dataset and system contributions will help to make touch interactions on ad-hoc surfaces more reliable.

8 ACKNOWLEDGEMENT

This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2019-05624 and by Rogers Communications Inc. under the Rogers-UBC Collaborative Research Grant: Augmented and Virtual Reality. We thank Jinhao Lu for video assistance and useful discussions. We thank Dr. Dongwook Yoon (The University of British Columbia) for supporting equipment.

REFERENCES

[1] Arturo Cadena, Rubén Carvajal, Bruno Guamán, Roger Granda, Enrique Peláez, and Katherine Chiluiza. 2016. Fingertip detection approach on depth image sequences for interactive projection system. In *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*. 1–6. https://doi.org/10.1109/ETCM.2016.7750827

[2] Elie Cattan, Amélie Rochet-Capellan, and François Bérard. 2015. A Predictive Approach for an End-to-End Touch-Latency Measurement. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. Association for Computing Machinery, New York, NY, USA, 215–218. https://doi.org/10.1145/2817721.2817747

[3] Elie Cattan, Amélie Rochet-Capellan, Pascal Perrier, and François Bérard. 2015. Reducing Latency with a Continuous Prediction: Effects on Users' Performance in Direct-Touch Target Acquisitions. In *Proceedings of the 2015 International*

Conference on Interactive Tabletops & Surfaces (ITS '15). Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/2817721.2817736

[4] Zhi Chai and Roy Shilkrot. 2018. Enhanced Touchable Projector-depth System with Deep Hand Pose Estimation. *arXiv:1812.11090 [cs]* (Dec. 2018). http://arxiv.org/abs/1812.11090 arXiv: 1812.11090.

[5] Jonathan Deber, Bruno Araujo, Ricardo Jota, Clifton Forlines, Darren Leigh, Steven Sanders, and Daniel Wigdor. 2016. Hammer Time! A Low-Cost, High Precision, High Accuracy Tool to Measure the Latency of Touchscreen Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2857–2868. https://doi.org/10.1145/2858036.2858394

[6] Andreas Dippon and Gudrun Klinker. 2011. KinectTouch: accuracy test for a very low-cost 2.5D multitouch tracking system. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. Association for Computing Machinery, New York, NY, USA, 49–52. https://doi.org/10.1145/2076354.2076363

[7] Florian Echtler, Manuel Huber, and Gudrun Klinker. 2008. Shadow tracking on multi-touch tables. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*. Association for Computing Machinery, New York, NY, USA, 388–391. https://doi.org/10.1145/1385569.1385640

[8] Thomas B. Fitzpatrick. 1988. The Validity and Practicality of Sun-Reactive Skin Types I Through VI. *Archives of Dermatology* 124, 6 (June 1988), 869–871. https://doi.org/10.1001/archderm.1988.01670060015008

[9] Yuxiang Gao and Chien-Ming Huang. 2019. PATI: a projection-based augmented table-top interface for robot programming. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. Association for Computing Machinery, New York, NY, USA, 345–355. https://doi.org/10.1145/3301275.3302326

[10] Jun Gong, Aakar Gupta, and Hrvoje Benko. 2020. Acustico: Surface Tap Detection and Localization using Wrist-based Acoustic TDOA Sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 406–419. https://doi.org/10.1145/3379337.3415901

[11] Dan Gregor, Ondrej Prucha, Jakub Rocek, and Josef Kortan. 2017. Digital playgroundz. In *ACM SIGGRAPH 2017 VR Village (SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/3089269.3089288

[12] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. https://doi.org/10.1145/3332165.3347947

[13] Taku Hachisu and Hiroyuki Kajimoto. 2013. HACHIStack: dual-layer photo touch sensing for haptic and auditory tapping interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1411–1420. https://doi.org/10.1145/2470654.2466187

[14] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2869–2881. https://doi.org/10.1145/2858036.2858095

[15] Daisuke Iwai and Kosuke Sato. 2005. Heat sensation in image creation with thermal vision. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '05)*. Association for Computing Machinery, New York, NY, USA, 213–216. https://doi.org/10.1145/1178477.1178510

[16] Daniel Kurz. 2014. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 9–16. https://doi.org/10.1109/ISMAR.2014.6948403

[17] Gierad Laput and Chris Harrison. 2019. SurfaceSight: A New Spin on Touch, User, and Object Sensing for IoT Experiences. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300559

[18] Jiajun Lu, Hrvoje Benko, and Andrew D. Wilson. 2017. Hybrid HFR Depth: Fusing Commodity Depth and Color Cameras to Achieve High Frame Rate, Low Latency Depth Camera Interactions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5966–5975. https://doi.org/10.1145/3025453.3025478

[19] Joe Marshall, Tony Pridmore, Mike Pound, Steve Benford, and Boriana Koleva. 2008. Pressing the Flesh: Sensing Multiple Touch and Finger Pressure on Arbitrary Surfaces. In *Pervasive Computing*, Jadwiga Indulska, Donald J. Patterson, Tom Rodden, and Max Ott (Eds.). Vol. 5013. Springer Berlin Heidelberg, Berlin, Heidelberg, 38–55. https://doi.org/10.1007/978-3-540-79576-6_3 Series Title: Lecture Notes in Computer Science.

[20] Takashi Matsubara, Naoki Mori, Takehiro Niikura, and Shun'ichi Tano. 2017. Touch detection method for non-display surface using multiple shadows of finger. In *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. 1–5. https://doi.org/10.1109/GCCE.2017.8229364

[21] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 519–528. https://doi.org/10.1109/VR50410.2021.00076 ISSN: 2642-5254.

[22] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. 2012. Extended multitouch: recovering touch posture and differentiating users using a depth camera. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. Association for Computing Machinery, New York, NY, USA, 487–496. https://doi.org/10.1145/2380116.2380177

[23] Takehiro Niikura, Takashi Matsubara, and Naoki Mori. 2016. Touch Detection System for Various Surfaces Using Shadow of Finger. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. Association for Computing Machinery, New York, NY, USA, 337–342. https://doi.org/10.1145/2992154.2996777

[24] Vivian Shen, James Spann, and Chris Harrison. 2021. FarOut Touch: Extending the Range of ad hoc Touch Sensing with Depth Cameras. In *Symposium on Spatial User Interaction (SUI '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3485279.3485281

[25] Ziwei Song, Yuichiro Kinoshita, Kentaro Go, and Gangyong Jia. 2021. Touch Point Prediction for Interactive Public Displays Based on Camera Images. In *2021 International Conference on Cyberworlds (CW)*. 133–136. https://doi.org/10.1109/CW52790.2021.00029 ISSN: 2642-3596.

[26] Naoki Sugita, Daisuke Iwai, and Kosuke Sato. 2008. Touch sensing by image analysis of fingernail. In *2008 SICE Annual Conference*. 1520–1525. https://doi.org/10.1109/SICE.2008.4654901

[27] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. 2020. TelemetRing: A Batteryless and Wireless Ring-shaped Keyboard using Passive Inductive Telemetry. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 1161–1168. https://doi.org/10.1145/3379337.3415873

[28] Mayuka Tsuji, Hiroyuki Kubo, Suren Jayasuriya, Takuya Funatomi, and Yasuhiro Mukaigawa. 2021. Touch Sensing for a Projected Screen Using Slope Disparity Gating. *IEEE Access* 9 (2021), 106005–106013. https://doi.org/10.1109/ACCESS.2021.3099901 Conference Name: IEEE Access.

[29] Andrew D. Wilson. 2010. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. Association for Computing Machinery, New York, NY, USA, 69–72. https://doi.org/10.1145/1936652.1936665

[30] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency tapping: using hover information to predict touch locations and eliminate touchdown latency. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/2642918.2647348

[31] 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 (ISS '16)*. Association for Computing Machinery, New York, NY, USA, 85–94. https://doi.org/10.1145/2992154.2992173

[32] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (April 2018), 1653–1660. https://doi.org/10.1109/TVCG.2018.2794222

[33] Lixing Zhang and Takafumi Matsumaru. 2016. Near-Field Touch Interface Using Time-of-Flight Camera. *Journal of Robotics and Mechatronics* 28, 5 (Oct. 2016), 759–775. https://doi.org/10.20965/jrm.2016.p0759 Publisher: Fuji Technology Press Ltd..

[34] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3025453.3025842