Feasibility Analysis of Detecting the Finger Orientation with Depth Cameras

Sven Mayer
University of Stuttgart
Pfaffenwaldring 5a
Stuttgart, 70569, Germany
Sven.Mayer@vis.uni-stuttgart.de

Michael Mayer
University of Ulm
Helmholtzstr. 16
Ulm, 89069, Germany
michael-1.mayer@uni-ulm.de

Niels Henze
University of Stuttgart
Pfaffenwaldring 5a
Stuttgart, 70569, Germany
Niels.Henze@vis.uni-stuttgart.de

MobileHCI '17, September 04–07, 2017, Vienna, Austria.

ACM ISBN 978-1-4503-5075-4/17/09.
https://doi.org/10.1145/3098279.312212

Abstract
Over the last decade, a body of research investigated enriching touch actions by using finger orientation as an additional input. Beyond new interaction techniques, we envision new user interface elements to make use of the additional input information. We define the fingers orientation by the pitch, roll, and yaw on the touch surface. Determining the finger orientation is not possible using current state-of-the-art devices. As a first step, we built a system that can determine the finger orientation. We developed a working prototype with a depth camera mounted on a tablet. We conducted a study with 12 participants to record ground truth data for the index, middle, ring and little finger to evaluate the accuracy of our prototype using the PointPose [3] algorithm to estimate the pitch and yaw of the finger. By applying 2D linear correction models, we further show a reduction of RMSE by 45.4% for pitch and 21.83% for yaw.

Author Keywords
Finger orientation; yaw; pitch; modeling; depth camera; touch; mobile devices.

ACM Classification Keywords
H.5.2 [User Interfaces]: Prototyping

Figure 1: The prototype with the 8.4-inch tablet with the depth camera.

Introduction & Background

Touchscreens are used in a large number of devices where they replaced physical buttons, sliders, and knobs. Moreover, a number of laptops with touchscreens are available where the touchscreen can be used as an alternative input to mouse and keyboard. However, today's touchscreens are mostly limited to 2D input. A body of research aimed to extend the throughput of a single touch action.

Colley and Häkkilä [1] investigated to use the finger type as additional information. In their prototype setup, they used a LEAP Motion to detect by which finger the phone was touched to trigger different actions based on the finger type.

Work by Xiao et al. [5] and Kratz et al. [3] both proposed to use the finger orientation as additional information. However, while Xiao et al. used the capacitive image from the touchscreen to estimate the pitch and yaw of the touching finger, Kratz et al. [3] used a depth camera attached above the touch screen to gain additional information about the finger orientation. One downside of the method by Xiao et al. [5] is that the device's operating system needs to be modified. In contrast, the method by Kratz et al. [3] can be applied to any touchscreen by using a depth camera. Xiao et al. [5] reported their method leads to a pitch error of $9.7°$ and a yaw error of $26.8°$ for their prototype. Kratz et al. [3] did not report an error; rather they reported the variation of the measurements within 7.5 seconds in which the participants were asked not to move their finger. Thus a comparison is not possible. With this work we determine the accuracy of the PointPose method proposed by Kratz et al. [3]. Further, we incorporate the idea by Colley and Häkkilä [1] into the PointPose evaluation. Thus we investigated not the accuracy using the index finger, but also determine the accuracy of the finger orientation using the middle, ring and little finger.

Based on previous work, our aim is it to build a low-cost prototype which can detect the pitch and yaw of the finger. Such a system is needed when designing and testing new UI elements which take the finger orientation and finger type into account. We envision the proposed apparatus to complement paper prototypes to investigate the possible use and feasibility of pitch and yaw already in the design process. Further, this can help to model the correlation between the finger orientation and the touch point as shown by Holz and Baudisch [2]. By using a depth camera, we can detect the fingers orientation on any flat surface. In contrast to vague and imprecise interaction (e.g. interaction with a secondary task), fine-grained interaction needs high precession. Therefore, we studied the accuracy of PointPose [3] and show how to further improve the accuracy. Thus, this work contributes (1) an analysis of PointPose's using the index, middle, ring and pinky finger, and (2) an offset model to reduce the error of PointPose.

Prototype

For our prototype, shown in Figure 1, we use a Samsung Galaxy Tab Pro 8.4 which offers $2560 \times 1600\,px$ on an 8.4-inch screen resulting in 359.39 *PPI*. As a depth camera, we use an Intel RealSense F200. The camera has a minimum sensing distance of $20\,cm$ and a resolution of $640 \times 480\,px$ at 120 *FPS*. We use the RealSense F200 due to its small minimum distance in comparison to other available depth sensors. However, we needed to overcome the $20\,cm$ between the tablets screen and the depth sensor. Therefore, we 3D printed a mount for the tablet and laser cutted a connection plate to attach the camera to the tablet. We firmly connected the parts using metal screws.

Experiment

To evaluate the accuracy of our setup with the algorithm proposed by Kratz et al. [3] we collected ground-truth data

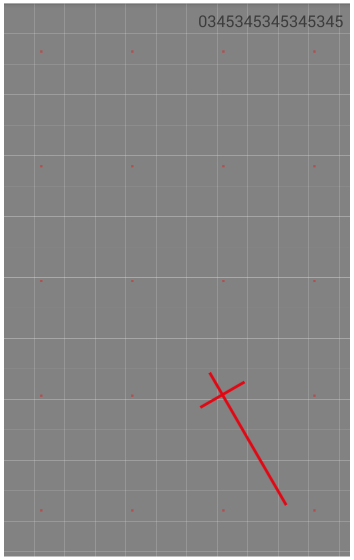


Figure 2: The study app is showing instructions to perform a 30° input at one specific position.

by conducting an experiment. The ground-truth was determent by three RGB cameras to always get a clear view.

Apparatus

We record the ground-truth data using three RBG cameras which we fixed on a wooden frame. We mounted one camera on top of the tablet, one on the left and one on the right (see Figure 3). The top camera was used to determine the fingers yaw while the left and right were used to determine the fingers pitch. We needed two cameras to determine pitch because when we insert extreme yaw angle one camera was always covered by the rest of the hand. We used three Microsoft Lifecam HD 3000 which recorded with 1280×720 @ 30 *FPS*. We replaced the flexible parts of the original camera mount with a non-flexible plastic connector (see Figure 3). The three camera streams were used to later determine the real finger orientation through a manual labeling process. We developed an Android application, which displays red crosshairs indicating the touch position. The crosshair further indicated which finger yaw angle the participant should perform, see Figure 2.

Design & Task

We designed the study using a repeated-measures design with four independent variables (IVs): TARGETS, YAW, PITCH, and FINGER. We randomized the order of FINGER, and within FINGER we randomized TARGETS and YAW. To cover a broad range of possible positions, we used 20 TARGETS arranged in a 4×5 grid on the tablet. The targets further represented five PITCH input angles: 15°, 30°, 45°, 60° and 75°. Xiao et al. [5] found that a pitch of 90° cannot be detected with long nails. Thus we did not investigate angles steeper than 75°. Further, we used five YAW input angles: -60°, -30°, 0°, 30° and 60°. Mayer et al. [4] found a comfort input zone of yaw for the right hand ranging from -33.75° to 101.25°, for the right hand. To not stress the

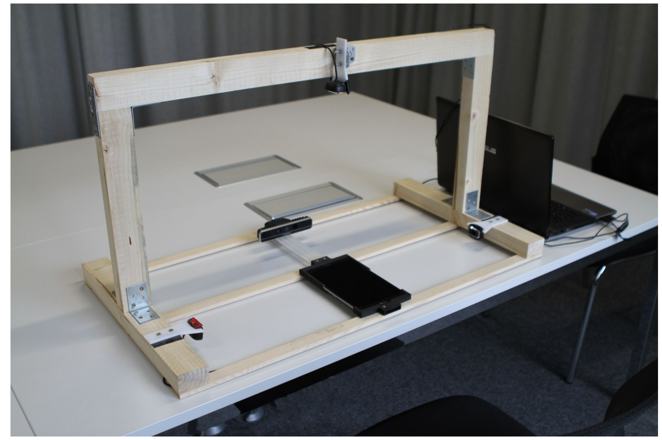


Figure 3: The wooden frame with the attached web cameras which we used for ground truth recording in our study.

participates too much we limited the range to -60° and to built a symmetric model to 60° on the other extreme. Further, all input tasks were performed with four FINGERS: index, middle, ring and little finger. Thus we had a design with $20 \times 5 \times 5 \times 4 = 2000$ conditions.

Performing a specific pitch angle is not easy. To overcome this issue Xiao et al. [5] used laser cutted stabilizers which they placed below the participants' finger. However, this is not possible using the camera based approach, as the stabilizers would influence the depth image. Thus we decided to let participants input a movement and determine the PITCH angle in the post processing. Therefore we asked half of the participants to start with a pitch close to 0° and then change the pitch of the finger up to a steep angle close to 90°. The other half was asked to move from 90° down to 0° pitch. We specified these two movements to reduce an effect of the finger moving only in one direction.

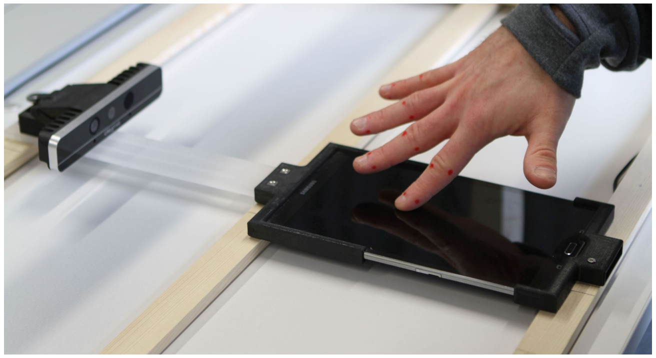


Figure 4: A participant while performing the task.

Procedure
First, we welcomed our participants and informed them about the procedure of the study. Second, we asked them to fill in a consent form and a questionnaire with demographic data. Afterward, the experimenter marked each finger with two red dots on the left and right side and on top of the finger to later calculate the finger orientation. Then, we explained that they have to touch the center of the red crosshair while aligning the finger with the longer red line indicating the yaw angle; they moved the finger slowly up or down to input several pitch angles (see Figure 4). We reimbursed the participants with €5.

Participants
We recruited 12 participants (3 female) which were aged between 22 and 35 (M $= 25.83$, SD $= 3.31$). All participants used their right hand.

Results
First, we corrected the camera lens distortion for the three recorded RGB-camera streams. Second, we manually labeled the finger posture with the help of the red markers

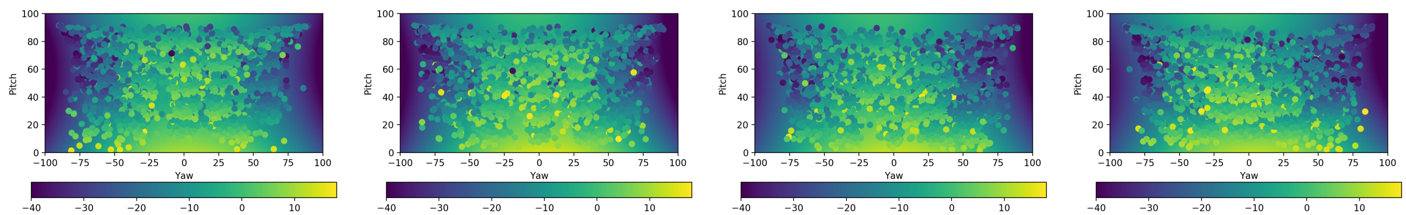
	Pitch			**Yaw**		
	RMSE	**M**	**SD**	**RMSE**	**M**	**SD**
Index	15.7	-10.8	11.4	11.9	2.8	11.7
Middle	$17.$	-10.5	13.4	14.7	3.4	14.3
Ring	13.8	-7.4	11.7	11.2	3.5	10.7
Little	14.8	-9.8	11.1	10.8	$3.$	10.4
Mean	15.4	9.6	11.9	12.2	3.16	11.8

Table 1: The RMSE and standard division for pitch and yaw per finger.

on the finger for each of the five PITCH angles. Due to the continuous change of the pitch angle, we were able to label accurate PITCH angles. However, for the YAW angles,s we were bound to the participants' accuracy (M $= 3.1°$, SD $= 9.9°$). For the modeling, we used the yaw angles actually performed by the participants, not the initial categories.

Using the depth images, we determined the pitch and yaw with the PointPose algorithm [3]. Due to the manual labeling and noise in the depth data, we removed outliers where the distance between ground-truth and predicted angles is more than two standard deviations away from the average. This was done for pitch and yaw individually. In total, we removed 8.2% of the data. Then, we calculated the root-mean-square error (RMSE) for each finger, see Table 1. The average RMSE is $15.4°$ for pitch and $12.2°$ for yaw.

The PointPose algorithm [3] was evaluated regarding precession over time. The evaluation of Kratz et al. [3] used an alignment task where the target was presented and the participant had to move a cursor to overlap with the target. Whereby the cursor could be manipulated through either pitch or yaw input. Accuracy was determined by measuring the variation of a 7.5 sec recording where the participants had to hold the alignment.



(a) Index finger **(b)** Middle finger **(c)** Ring finger **(d)** Little finger

Figure 5: The scatterers are showing the points where we gained data samples from the study. The underlining plain represents the correction model for the pitch correction based on pitch and yaw of the depth camera.

$$f(\alpha, \beta) = a\alpha^2 + b\beta^2 + c\alpha\beta + d\alpha + e\beta + f$$

Equation 1: The modeling function, α and β are the pitch and yaw values of result gain by the PointPose algorithm using the depth camera point cloud.

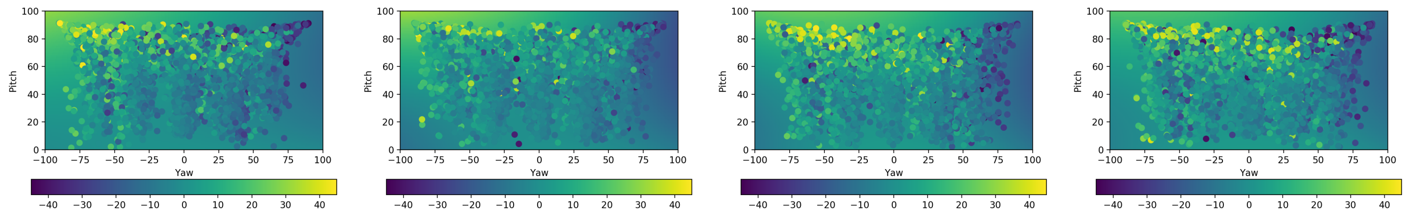
Comparing our results with the results reported by Kratz et al. [3] is hard because Kratz et al. averaged over 7.5 seconds whereby we used a concrete error not the variance while holding the finger. Further, Kratz et al. [3] used 7 steps for yaw ranging from $-30°$ to $30°$ and 5 steps for pitch ranging from $50°$ to $75°$. On average they reported a change in variation of M $= -.92°$ (SD $= 6.36°$) for pitch and M $= -2.52°$ (SD $= 14.67°$) for yaw.

Modelling

In the following, we present our model to reduce the error through offset correction. We modeled the offset with a full second order two-dimensional polynomial, as in Equation 1. We choose Equation 1 after a one-dimensional polynomial fitted less accurate and the visual inspection suggested a more complex underlying behavior. Furthermore, a more complex function led to overfitting. The pitch and yaw offset corrections are modeled independently from each other. Thus we fitted 8 functions, 4 fingers $\times$ 2 degrees of orientation (pitch/yaw). However, the correction model for pitch and yaw is based on both pitch and yaw angles gained from the depth camera as the α and β input for the Equation 1. Whereby we used the predicted angles by the depth camera for α and β, results are shown in Figure 5 for the pitch correction and in Figure 6 for the yaw correction. We validated the improvements for all functions by the use of leave p out cross validation with $p = 3$, which is a split of $75\% : 25\%$ for train and test.

	Pitch in $\%$	**Yaw in** $\%$
Index	41.8	15.2
Middle	$43.$	16.5
Ring	40.4	13.9
Little	41.5	13.2
Mean	41.7	14.7

Table 2: The reduction of RMSE when applying the correction models to pitch and yaw.



(a) Index finger **(b)** Middle finger **(c)** Ring finger **(d)** Little finger

Figure 6: The scatterers are showing the points where we gained data samples from the study. The underlining plain represents the correction model for the yaw correction based on pitch and yaw of the depth camera.

For the pitch correction, we achieved an average reduction of the RMSE of 41.7%, all results are listed in Table 2. The overall remaining pitch error improved from M $= -9.6°$ without correction model to M $= -.9°$ with correction model. For the yaw correction, we achieved an average reduction of the RMSE of 14.7%, all results are listed in Table 2. The overall remaining yaw error improved from M $= 3.2°$ without correction model to M $= .2°$ with correction model.

For the final model we used the training and test data to fit the model, we achieved an RMSE reduction by 45.4% for pitch and 21.83% for yaw. Further the fitness of the pitch correction functions for the four fingers functions is $R^2 = [.50 \;\; .45 \;\; .49 \;\; .54]$ (see Figure 5), and for the four yaw functions the fitness is $R^2 = [.63 \;\; .58 \;\; .67 \;\; .76]$ (see Figure 6).

Discussion and Implications

In a first step, we recorded ground truth data pitch and yaw to determine the accuracy of PointPose. In a second step, we applied offset models to correct the mean error of the PointPose method.

We showed that the root-mean-square error without correction is $11.75°$ for pitch. This results in an offset of 13.1% of the possible pitch input range which is from $0°$ to $90°$. Further, in our study, we explored the yaw range between $-60°$ and $60°$ resulting in an RMSE of $8.74°$ and an average offset of 7.3%. Thus high precision input is not possible with the proposed method. Even for imprecise input, we see a lack of feasibility to use this method.

We also show that the predicted results are more accurate close to the center of the observed input space (pitch $= 70°$ and yaw $= 0°$), see Figure 5 and Figure 6. For the pitch correction, we can see an overall trend of a larger pitch error with yaw values away from the center. Further, we can observe that this is radially symmetric (see Figure 5). Also for the yaw correction, we see an overall drift in the mean data (see Figure 6). To correct the drift and improve the predicted accuracy, we applied one offset model per pitch/yaw and finger and thereby reduction of RMSE for pitch by 45.4%, and for yaw by 21.83%.

When comparing our results with correction and the results reported by Xiao et al. [5], we achieve a similar pitch error and smaller yaw error. Their method leads to a pitch error of $9.7°$ while our method achieved $11.75°$. For the yaw error, Xiao et al. [5] reported $26.8°$ while our method achieved a three times smaller yaw error of $8.74°$.

We used the tablet to display targets, not for the actual recognition nor the model. Thus, the touch position was not taken into account in the analysis nor the offset correction. Doing so allows using the depth camera also without a tablet. Thus, mounting the depth camera onto a not touch sensitive surface is possible; this can be useful for system prototyping when building a first fully functional apparatus. We envision using our approach even in earlier stages e.g. when designing new UI interfaces using paper prototypes. Here, the behavior of the finger orientation can be observed, and the UI can be designed adaptive to the input.

Conclusion

As a first step, we build a prototype to determine the accuracy of the algorithm by Kratz et al. [3]. As a second step, we showed a reduction of RMSE by 45.4% for pitch and 21.83% for yaw with our offset correction model. Further, we showed that the algorithm proposed by Kratz et al. [3] also could determine the pitch and yaw of the middle, ring, and little finger with an equal accuracy.

We build a prototype which can be applied to any flat surface to determine the yaw and pitch of the finger. This is especially handy when working with paper prototypes to investigated new interaction techniques that take the finger orientation into account.

With Google's Project Tango[1] we saw the first mobile device equipped with a built-in depth camera. Even if their camera is a back-facing camera, they showed the feasibility of the hardware setup and further that today's mobile devices have the computational power to process depth camera data in real time. Thus building a compact mobile device with a front facing depth camera would be the next step to gain the finger orientation for additional input.

[1] https://get.google.com/tango/ (last acceded: 06-23-2017)

Future Work

Based on the new offset models the next step is to evaluate these with specific applications. The aim is to determine if our setup can be used in prototypes to investigate possible features of future screens. We especially see potential by enhancing paper prototypes to observe how the finger orientation changes with different UI's and how feasible it is to input specific pitch and yaw postures in UI's which make use of this kind of input. The problem of determining the finger type is one open issue which we aim to address in the future.

Acknowledgments

This work is partly financially supported by the German Research Foundation (DFG) within Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart.

REFERENCES

1. Ashley Colley and Jonna Häkkilä. 2014. Exploring finger specific touch screen interaction for mobile phone user interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures the Future of Design (OzCHI '14)*. ACM Press, New York, New York, USA, 539–548. `DOI: http://dx.doi.org/10.1145/2686612.2686699`

2. Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*

(CHI '11). ACM, New York, NY, USA, 2501–2510. DOI: http://dx.doi.org/10.1145/1978942.1979308

3. Sven Kratz, Patrick Chiu, and Maribeth Back. 2013. PointPose: Finger Pose Estimation for Touch Input on Mobile Devices Using a Depth Sensor. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. ACM, New York, NY, USA, 223–230. DOI: http://dx.doi.org/10.1145/2512349.2512824
4. Sven Mayer, Perihan Gad, Katrin Wolf, Paweł W. Woźniak, and Niels Henze. 2017. Understanding the Ergonomic Constraints in Designing for Touch Surfaces. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/3098279.3098537
5. Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 47–50. DOI: http://dx.doi.org/10.1145/2817721.2817737