

Using an Infrared Pen as an Input Device for Projected Augmented Reality Tabletops

Vitus Maierhöfer
University of Regensburg
Regensburg, Germany
vitus.maierhoefer@ur.de

Andreas Schmid
University of Regensburg
Regensburg, Germany
andreas.schmid@ur.de

Raphael Wimmer
University of Regensburg
Regensburg, Germany
raphael.wimmer@ur.de

ABSTRACT

Interactive tabletops do not only offer a large surface for collaborative interaction. They also offer quick access to digital tools directly at the table - where a large number of everyday activities take place. Tabletops with an embedded display are generally less flexible and more fragile than ordinary massive tabletops. Physical objects on the tabletop occlude the digital content. In contrast, top-down-projected interfaces using an overhead camera-projector system allow for augmenting arbitrary tables and the object lying on them. However, detecting pointing input only via a camera image captured from above requires robustly recognizing whether a finger or pen touches the tabletop or whether it hovers slightly above it. In this paper, we present a solution for reliably tracking a pen on arbitrary tabletop surfaces. The pen emits infrared light via a tip made of optical fiber. A camera captures position and shape of the light point on the surface. Our open-source tracking algorithm combines heuristics and a machine learning model to distinguish between drawing and hovering. A pilot study with 7 participants shows that that this system can be reliably used for drawing and writing on tabletops. However, occlusion by users' hands can deteriorate tracking of the pen.

KEYWORDS

projected augmented reality, input device, pen input, computer vision

1 INTRODUCTION

Interactive tabletops facilitate collaborative use with intuitive interaction techniques. Therefore, they are used in a wide range of applications [16], such as interactive dining tables [4], augmenting an office desk [6], assistive systems for workplaces [5], collaborative work [3], playful interaction [17], smart kitchens [9], or music making [7, 8]. Because of their high price, fragile surface, and oftentimes bulky form factor, interactive tables with built-in displays or back-projection are not suitable for everyday use [15]. Projected augmented reality (PAR) tackles this problem as it can be used in combination with uninstrumented surfaces to create interactive tabletops. For example, it is possible to project onto objects on top of a table and the surface can still be used for its designated purpose.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MuC'22, 04.-07. September 2022, Darmstadt

© 2022 Copyright held by the owner/author(s).

<https://doi.org/10.18420/muc2022-mci-ws04-274>

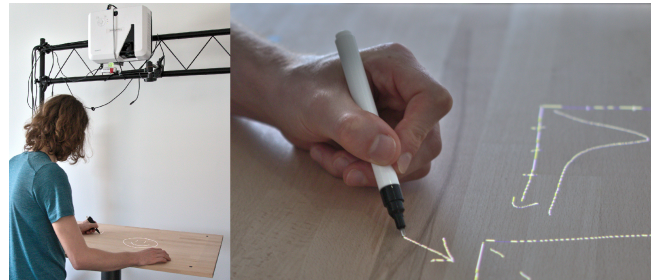


Figure 1: Left: Our projected augmented reality system. An infrared camera captures light emitted by the IR pen. A top-mounted projector displays drawn lines. Right: The IR pen can be used to draw arbitrary shapes and text on the table.

This is especially important for applications that are used during tasks such as crafting, where a robust surface is required.

However, if the interaction surface is uninstrumented, reliably tracking user's input becomes a challenge. For traditional pointing input, it is necessary to facilitate input by precisely determining position and form of interaction, such as clicking or dragging. Using only sensors mounted above the tabletop, it is hard to implement touch input as the system would have to reliably distinguish between hovering closely over the surface and touching it.

In this paper, we present an input form for projected augmented reality (PAR) tabletops based on an infrared (IR) emitting pen and an IR camera mounted above the table. Our work consists of

- a low-cost and easily replicable prototype for an IR pen
- an algorithm to precisely determine the pen's position from an IR camera's image
- a machine learning model that robustly detects whether the pen is touching a surface or not
- an evaluation of our system through a small user study (n=7)

2 RELATED WORK

Using infrared pens as an input device for interactive tabletops has been investigated in numerous research projects.

*Anoto Livescribe Smartpens*¹ are commercial ballpoint-pens with an embedded high-speed IR camera to capture a dot pattern that is printed directly on the paper. It allows instant digitization of user's handwriting, but requires special paper with a propriety pattern printed on it. Whereas in this case the duality of having an analog transcript and its digital representation is desirable, other use cases require the ability to write on surfaces and objects without permanently altering them.

¹<https://www.anoto.com/solutions/livescribe/>

A common approach to track the position of pens on interactive surfaces is to integrate tracking hardware directly into the tabletop. Vandoren et al. [14] present a digital painting system using a paint brush with a built-in infrared LED and light conducting bristle fibers. Users can paint on a special surface that uses rear projection. A camera with an IR band pass filter tracks the IR brush from below the semi-transparent tabletop. The system can detect the brush hovering above the surface and can estimate the brushes contact surface depending on the applied pressure.

If uninstrumented tabletops are required, tracking infrastructure has to be placed away from the table. To our knowledge, only a small amount of research on pen-like input methods for such systems has been published.

Lee et al. [11] placed an infrared camera next to a projector and used a custom infrared pen to turn the projection area on the wall into a digital whiteboard. While their approach has no hardware on or under the projection surface and could be adopted to a horizontal tabletop setting, it is limited by the fact that the user needs to press a button on the pen to activate the IR LED. This makes interaction less natural than writing with an ordinary pen. Additionally, this approach can not distinguish a pen's state between hovering or touching the surface, providing no feedback about the tracked position to users.

Multiple research projects use *Nintendo Wii Remotes* [2, 10] to track custom infrared pens. Lee [10] also required users to press a button on the pen to activate the IR LED. By using a pen with a pressure sensitive tip, Chen et al.'s approach [2] removes the need to actively push a button while drawing, but requires users to constantly push the pen onto the surface, which seems to impair writing in a natural way. Furthermore, hover events could not be detected directly and their processing pipeline had a significant latency of about 150 ms.

Margetis et al. [13] propose an approach that tracks an IR pen in a 3D space above a tabletop using stereoscopic images from two calibrated cameras. Even though their system can reliably distinguish between hovering and drawing, it can achieve a maximum of 29 Hz which is insufficient for natural handwriting detection.

3 TRACKING INFRASTRUCTURE

We extend the existing body of research with a tracking method for IR pens that works on arbitrary surfaces, allows for natural input with high temporal and spatial resolution and reliably distinguishes between hovering and drawing.

3.1 Hardware

The system hardware consists of an infrared camera, a 4k video projector, our custom infrared pen and a computer.

3.1.1 Projector/Camera Setup. Images are captured with a *RealSense D435*² stereoscopic depth camera. We only use one of its two infrared sensors, which can provide monochrome frames with a resolution of 848x480 pixels at a high frame rate of 90 fps. As the *RealSense* camera is slightly sensitive in the visible spectrum, we attached an infrared band pass filter (850 nm) to the lens to block remaining ambient light, including the projector's output.

²<https://www.intelrealsense.com/depth-camera-d435/>

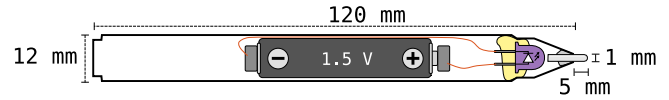


Figure 2: Schematic of the infrared pen. A battery-powered infrared LED shines light through a light guide which is used as the pen tip. All components are included in an off-the-shelf felt tip pen. The battery is connected to the LED with thin copper wires and magnetic contacts.

A generic *ViewSonic LS700-4K* projector is used in the setup. While lower resolution projectors could be used as well, the high resolution of 3840 × 2160 px enables more precise drawing and writing due to sharper lines.

3.1.2 System setup and calibration. We used a mobile truss system (Fig. 1, left) to mount camera and projector above the tabletop. To allow for precise projection of drawn content without any offset, projector and camera need to be calibrated to each other. We created a calibration tool which allows users to select the four corners of the projection in the camera frame. While more sophisticated approaches for projector-camera calibration exist, ours proved to be more than sufficient for our prototype. As long as the hardware is not moved, no further calibration is needed.

3.1.3 Infrared Pen. We built a pen emitting infrared light by placing an infrared LED³ inside a black *edding 400* permanent marker (Fig. 2). A fiber-optic light guide with a sanded end is hot-glued inside the pen's tip to lead the IR light towards the drawing surface and makes the device's haptic similar to a normal pen. We compared side-emitting and conventional light guides and found that both work with our system. A 1.5 V AAAA battery inside the pen powers the IR LED. Because there is no room for a proper battery holder inside the pen, we used magnetic contacts and thin copper wires to connect the LED to the battery.

3.2 Image processing

The *RealSense* camera produces infrared frames at 90 fps. To get frames that are perfectly aligned with the projection, we extract the relevant area from each image with a perspective transformation based on four corner points collected in the calibration step. We process those extracted images with Python 3.8 and *OpenCV* 4 [1]. First, we remove potential dim areas caused by sunlight with a binary threshold. For each remaining region above the threshold, we extract a region of interest (ROI) of 48 × 48 px (Fig. 3) and pass it to the classifier (section 3.3), which returns a prediction of the pen's state (*hover* or *draw*). The camera's low resolution of 848 × 480 would prevent us from using the full potential of the projector's high resolution. Therefore, the region around a detected pen spot is scaled up with linear interpolation to match the projector's 4k output. By calculating the centroid of the resulting blob, we can detect the pen's position with sub-pixel precision. A *PyQt5*⁴ application receives all events and draws points and lines on a black

³SFH 4550, 850 nm, 2800 mW/sr, φ 3°

⁴<https://www.riverbankcomputing.com/static/Docs/PyQt5/>

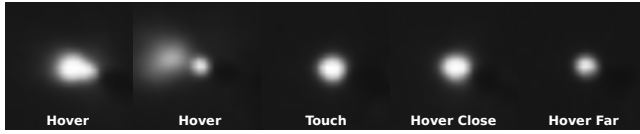


Figure 3: Cropped and re-scaled sensor images. If the pen is hovering over the surface (> 2 mm), a light cone is clearly visible. When touching the surface, the light cone disappears. Close hovering (< 2 mm) looks very similar to touching. Hovering far from the surface (> 15 cm) also looks similar, but the spot is smaller and less bright.

canvas, which is then projected onto the table. Processing a single frame takes around 9 milliseconds on our system⁵.

3.3 Pen State Classification

We use a convolutional neural network (CNN) to distinguish between the pen states *draw* and *hover*. The network was implemented in *TensorFlow 2.0*⁶ using the *Keras API*⁷. It consists of three 2D convolutional layers ($64 \times 3 \times 3$ kernels, $32 \times 5 \times 5$ kernels, $16 \times 7 \times 7$ kernels) with linear activation, each followed by a max pooling layer. After flattening, there are four dense layers with 128 neurons each and *ReLU* activation. A softmax layer classifies the result into one of the two states *draw* or *hover*. Hyperparameters of the network were optimized via grid search.

Training data was acquired by manually moving the infrared pen across the table, continuously capturing images with the *RealSense* camera, and saving those images in a directory with the corresponding label *draw* or *hover*. Capturing 500 training images this way takes about 30 seconds. To increase robustness of the network, this process was repeated under different lighting conditions (e.g. different times of day, open/closed window blinds) and with different exposure settings for the camera (exposure: 10 ms; gain: 16, 32, 64). This way, we could acquire a data set of 3000 8 bit grayscale images with 848×480 pixels. As the pen's light cone only covers a small part of those images, they were pre-processed by cropping to the brightest 48×48 pixel region. Before training, each of those cropped images was augmented by rotation in 90° steps, mirroring, and varying its brightness in 6 steps (70% – 120% brightness). 80% of this final data set were used to train the CNN, the remaining 20% were used for validation.

We trained our network with a batch size of 32 using an *Adam* optimizer with categorical crossentropy as the loss function. After three epochs, the model reached an accuracy of 97.85% on the validation data set.

Using *TensorFlow lite*⁸, the model achieves a mean prediction time of 2.6 ms (std: 1.94 ms, max: 6.8 ms).

4 EVALUATION

To evaluate the performance of our system, we recruited seven people (aged 21–36 (mean 25.7), three male/four female, four right handed/three left handed) to participate in a short user study. All

of our participants were unfamiliar with the system and we asked them to pay attention to the system's performance and give us feedback afterwards. The study consisted of four tasks. With the first task, we determined the accuracy of our tracking system by counting the number of wrongly detected segments. To this end, we projected a pattern with big and small crosses, as well as dots, on the table. Participants were asked to trace those symbols with the IR pen. They could not see the drawn lines in this task to avoid adaption to the system. With the second task, we evaluated how well people can write with the IR pen. We displayed a random phrase from MacKenzie and Soukoreff's phrase set [12] and asked participants to copy it inside a projected box in the center of the table. This process was repeated five times each for two differently sized boxes (3 cm and 4.5 cm height) to stimulate size variation in user's handwriting. For the third task, participants had three minutes to freely draw on the table. This way, we gave them the opportunity to explore the system's capabilities and adapt to its behavior. Afterwards, we repeated the first task to see whether participants had adapted to the system and the number of correctly recognized lines would increase. After all tasks were finished, we asked participants to provide feedback for our system in a short interview.

5 RESULTS AND DISCUSSION

Over all, participants liked interacting with our system and described interaction as natural, similar to a normal pen. There were tracking problems for all participants, mainly caused by user's hands occluding the line of sight between the camera and the pen's infrared spot, especially when holding the pen very steeply with a closed grip. Therefore, tracking worked better on the left hand side of the table for left handed users and vice versa. Participants reported that they quickly adapted to the system by holding the pen less vertically with a more open grip once they realized this behavior. All but one participant had better results in the second cross-hatching task (mean accuracy: 75% $\rightarrow$ 82.1%, Fig. 4), confirming that they adapted to the system within the short time of the study.

All phrases written during the second task were easily legible (Fig. 5). To test for machine-readability, we used handwriting recognition software⁹ on all written phrases and counted the number of wrongly detected characters. Out of the 70 total phrases written, only 1.2% of characters were wrongly detected.

During development and evaluation of our system, we observed that hovering the pen very closely to the table's surface is oftentimes misinterpreted as drawing. Even though this could lead to two lines unintentionally being connected, we found that this was rarely a problem even when writing small text. Additionally, we found that our system works robustly for slight changes in lighting conditions. However, for a more severe change in lighting, for example when moving the setup from a dark corner to a window, re-calibrating the camera's exposure settings and/or re-training the model might be necessary.

⁵Ubuntu GNU/Linux 20.04, Intel i9 9900k @3.6 GHz, NVIDIA GeForce RTX 2060

⁶<https://www.tensorflow.org/>

⁷<https://keras.io/>

⁸<https://www.tensorflow.org/lite>

⁹<https://lens.google/>

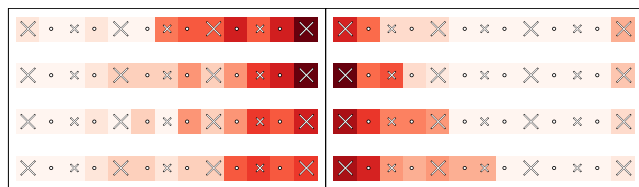


Figure 4: Results of the cross-hatching task described in section 4. Left: results for left handed participants. Right: results for right handed participants. The background color indicates how well our system could recognize user's input. White regions were detected for all participants, deep red regions were rarely detected. It can be seen that occlusion by the writing hand impairs tracking.

The assignment is due today
 a feeling of complete exasperation
 exercise is good for the mind
 the objective of the exercise

Figure 5: Sentences written by users during the handwriting task.

6 CONCLUSION AND FUTURE WORK

We could show that infrared pens can be used as input devices for projected augmented reality tabletops. In contrast to other pen tracking methods for PAR applications (such as Margetis et al. [13]), our system is able to precisely determine the pen's position and state at 90 fps, making natural handwriting on the table possible. Results of our study indicate that user's intuitively know how to interact with the device.

The main problem of our tracking method is occlusion by user's hands. We plan to integrate a second camera into our system to counteract this issue. As a side effect, this might also help our pen state classifier to distinguish between close hovering and drawing more reliably. In future work, we will implement more applications using our new input method and evaluate the whole system in an extensive user study.

The complete source code can be found in our GitHub repository <https://github.com/vigitia/IRPenTracking>.

ACKNOWLEDGMENTS

This project is funded by the Bavarian State Ministry of Science and the Arts and coordinated by the Bavarian Research Institute for Digital Transformation (bidt).

REFERENCES

- [1] G. Bradski. 2000. The OpenCV Library. *Dr. Dobbs' Journal of Software Tools* (2000).
- [2] Rui Chen, Po-Jui (Ray) Chen, Rui Feng, Yilin (Elaine) Liu, Andy Wu, and Ali Mazalek. 2014. SciSketch: a tabletop collaborative sketching system. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction - TEI '14*. ACM Press, Munich, Germany, 247–250. <https://doi.org/10.1145/2540930.2540973>
- [3] Florian Echtler. 2018. SurfaceStreams: A Content-Agnostic Streaming Toolkit for Interactive Surfaces. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings - UIST '18 Adjunct*. ACM Press, Berlin, Germany, 10–12. <https://doi.org/10.1145/3266037.3266085>
- [4] Florian Echtler and Raphael Wimmer. 2014. The Interactive Dining Table, or Pass the Weather Widget, Please. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces - ITS '14*. ACM Press, Dresden, Germany, 419–422. <https://doi.org/10.1145/2669485.2669525>
- [5] Markus Funk, Sven Mayer, and Albrecht Schmidt. 2015. Using In-Situ Projection to Support Cognitively Impaired Workers at the Workplace. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*. ACM Press, Lisbon, Portugal, 185–192. <https://doi.org/10.1145/2700648.2809853>
- [6] John Hardy. 2012. Experiences: a year in the life of an interactive desk. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. Association for Computing Machinery, Newcastle Upon Tyne, United Kingdom, 679–688. <https://doi.org/10.1145/2317956.2318058> ZSCC: 0000012.
- [7] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07)*. Association for Computing Machinery, New York, NY, USA, 139–146. <https://doi.org/10.1145/1226969.1226998>
- [8] Sergi Jordà, Martin Kaltenbrunner, Günter Geiger, and Marcos Alonso. 2006. The reacTable: a tangible tabletop musical instrument and collaborative workbench. In *ACM SIGGRAPH 2006 Sketches on - SIGGRAPH '06*. ACM Press, Boston, Massachusetts, 91. <https://doi.org/10.1145/1179849.1179963>
- [9] Thomas Kosch, Paweł W. Woźniak, Erin Brady, and Albrecht Schmidt. 2018. Smart Kitchens for People with Cognitive Impairments: A Qualitative Study of Design Requirements. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, Montreal QC, Canada, 1–12. <https://doi.org/10.1145/3173574.3173845>
- [10] Johnny Chung Lee. 2008. Hacking the Nintendo Wii Remote. *IEEE Pervasive Computing* 7, 3 (July 2008), 39–45. <https://doi.org/10.1109/MPRV.2008.53> Conference Name: IEEE Pervasive Computing.
- [11] Kuk-Seon Lee, Sang-Heon Oh, Kuk-Hui Jeon, Seong-Soo Kang, Dong-Hee Ryu, and Byung-Gyu Kim. 2012. A Study on Smart Touch Projector System Technology Using Infrared (IR) Imaging Sensor. *Journal of Korea Multimedia Society* 15, 7 (July 2012), 870–878. <https://doi.org/10.9717/KMMS.2012.15.7.870>
- [12] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 754–755. <https://doi.org/10.1145/765891.765971>
- [13] G. Margetis, A. Ntelidakis, X. Zabulis, S. Ntoa, P. Koutlemanis, and C. Stephanidis. 2013. Augmenting physical books towards education enhancement. In *2013 1st IEEE Workshop on User-Centered Computer Vision (UCCV)*. 43–49. <https://doi.org/10.1109/UCCV.2013.6530807>
- [14] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Chris Raymaekers, and Frank Van Reeth. 2008. IntuPaint: Bridging the gap between physical and digital painting. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. IEEE, Amsterdam, 65–72. <https://doi.org/10.1109/TABLETOP.2008.4660185>
- [15] Raphael Wimmer. 2017. Some Thoughts on Ergonomic, Practical, and Economic Properties of Interactive Tabletops. Brighton, UK. https://thedisappearingworkshop.files.wordpress.com/2017/06/disappearingtabletop2017_paper_5.pdf
- [16] Raphael Wimmer and Florian Echtler. 2019. VIGITIA: Unterstützung von alltäglichen Tätigkeiten an Tischen durch Projected AR. In *Mensch und Computer 2019 - Workshopband*. Gesellschaft für Informatik e.V., 6. <https://doi.org/10.18420/muc2019-ws-620> Publisher: Gesellschaft für Informatik e.V..
- [17] Qin Wu, Jiayuan Wang, Sirui Wang, Tong Su, and Chenmei Yu. 2019. MagicPAPER: tabletop interactive projection device based on tangible interaction. In *ACM SIGGRAPH 2019 Posters on - SIGGRAPH '19*. ACM Press, Los Angeles, California, 1–2. <https://doi.org/10.1145/3306214.3338575>