TipTrack: Precise, Low-Latency, Robust Optical Pen Tracking on Arbitrary Surfaces Using an IR-Emitting Pen Tip

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

Tables are focus points for social interactions and support everyday activities, such as learning, crafting, or dining. These physical interactions on and around the table may be augmented with digital information and tools projected onto the tabletop. For interaction with such projected information, touch input suffers from technical and interactional limitations. Pen input is a more robust alternative that does not suffer from Midas-touch problems. We developed a system for tracking the position of an IR-emitting pen tip on a planar surface with sub-millimeter resolution and an end-to-end latency of less than 30 ms. Distinguishing between drawing and hovering states is done by combining a stereoscopic camera setup and a machine-learning classifier. We demonstrate practical performance, uses and limitations through multiple studies and examples.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; **Interaction devices**; *Mixed / augmented reality*.

KEYWORDS

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

ACM Reference Format:
Vitus Maierhöfer, Andreas Schmid, and Raphael Wimmer. 2024. TipTrack: Precise, Low-Latency, Robust Optical Pen Tracking on Arbitrary Surfaces Using an IR-Emitting Pen Tip. In *Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '24), February 11–14, 2024, Cork, Ireland*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3623509.3633366

Figure 1: TipTrack enables pen interaction, drawing and handwriting on unmodified tabletops via a camera-projector combination mounted above the table. Pen tracking works well enough for detailed drawings.

1 INTRODUCTION

Tables have been supporting everyday activities and work for millennia [2]. They are a natural focus point for individual and collaborative work, for dining, and for spending time together [38].

As physical and digital life have become ever more interwoven, it made sense to also bring digital tools to the physical table. In the early 2000s, several companies brought *interactive tabletops* to market - most notably Microsoft and Samsung with the *Surface* and *SUR40* devices. However, despite significant research, media attention, and investment, these interactive tabletops have failed to achieve widespread adoption. Interactive tabletops with built-in displays are used in special settings – such as museums – but have inherent limitations that make them unsuitable for use in home environments [34].

One of the first interactive tabletop prototypes used a different approach which seems more suitable for everyday use. Pierre Wellner's DigitalDesk [33] was made interactive using a combination of top-down projection and camera-based tracking of objects and interactions. This approach has several advantages over glass displays on four legs. It works with most unmodified tabletops which people already have in their homes, thus saving space, preserving the general affordances of a table, and allowing digital equipment to be updated independently from the furniture. Furthermore, the top-down approach stays out of the way and allows for projecting digital augmentations onto objects on the table. However, a major drawback of top-down projection and optical sensing is that the users' actions have to be inferred from captured camera frames. This is much more difficult than sensing touch and simple objects from below the surface using capacitive or optical touch screens.

The three most common input modalities on interactive tabletops are touch, tangibles, and stylus. As they have different advantages and limitations, ideally all three modalities should be supported. Tracking tangibles from above is relatively straightforward thanks to optical markers (fiducials) which can be added to arbitrary objects [11, 14]. For hand tracking and touch input, multiple camera-based approaches exist [32, 37]. While touch input is immediate and intuitive, pen input complements it and is better suited for some application scenarios and tasks. Touch interaction on tabletops

TEI '24, February 11–14, 2024, Cork, Ireland

ACM ISBN 979-8-4007-0402-4/24/02.
https://doi.org/10.1145/3623509.3633366

faces two inherent challenges, independent of sensing approach: the *fat-finger* problem – reduced precision due to occlusion of content below the hand [12] – and the *Midas-touch* problem: every time one touches the tabletop, a reaction from the system might be triggered.

Pen input avoids these challenges, allowing for simpler, more precise, and more robust pointing and selection: As only little of the underlying surface is occluded by a pen tip, pointing and writing is more precise with a pen than with a finger. If the system only reacts to pen input, and not to touch, the Midas-touch problem does not occur. People can work with physical objects on the table without having to worry that they might inadvertently trigger a digital, projected, object. Furthermore, as there is no fat-finger problem, UIs designed for pen input can be more condensed than touch-optimized UIs. Pens are also much better suited for handwriting or drawing than fingers. However, reliably tracking a pen tip and determining whether it touches a surface is not trivial.

In this paper, we describe a robust and fast approach for tracking pens on an unmodified tabletop. Our setup uses two IR cameras mounted above the tabletop and an active pen with an IR-emitting tip made from PMMA fiber.

2 REQUIREMENTS FOR A NATURAL WRITING EXPERIENCE

Physical pens have effectively zero latency, no offset, and work quite reliably. When the pen touches the paper, it immediately leaves a small ink stain at exactly the point of contact. These qualities are essential for a physical pen. On digital devices, users have gotten accustomed to the inherent latency and limited resolution of computers compared to pen and paper. However, pen input on interactive tabletops is so far in the physical domain that people expect a higher level of performance, comparable to a physical pen. The following requirements for digital pens in a physical world are derived from our own experience in building interactive hardware and software and from anecdotal observations reported in related work.

For handwriting or drawing, a temporal resolution of at least 100 Hz and a spatial resolution better than 0.5 mm seem essential for capturing smooth, undistorted strokes and letters of about 10 mm in height. Higher spatial resolution is always beneficial.

Latency should be as low as possible. While humans can compensate for latencies of up to several hundred milliseconds in a user interface [25], most people are able to perceive the effects of latency down to a few milliseconds [24]. In our experience, a latency of less than 50 ms feels acceptable – however, people notice and appreciate much lower latencies. It is important to not only optimize the latency of the tracking system but also the end-to-end latency of the whole system including camera and projector. An off-the-shelf display or projector typically has a latency of tens of milliseconds and the common refresh rate of 60 Hz already introduces an average latency of 8 ms. Thus, achieving low end-to-end latency inherently requires sensors and projectors with high frame rates and low internal latency.

The acceptable offset between pen tip and displayed line depends on the use case. For coarse strokes, an offset of multiple millimeters may be okay. However, for cursive handwriting or drawing, sub-millimeter offsets are desirable. We have observed that even small offsets make it hard to insert new details to an existing drawing or to write letters with curved lines.

A further essential requirement is that the tracking system needs to accurately determine whether the pen tip is in contact with the surface, or hovering slightly above it. The *minimum hover distance* is a measure for how close the pen can be to the surface without the tracking system registering a contact while still reliably reporting a contact while the pen tip actually touches the surface. For UIs where people only click buttons with a pen, and otherwise keep it far from the surface, a minimum hover distance of several millimeters may be acceptable. For handwriting, where people only slightly lift off the pen tip between strokes, a minimum hover distance of less than one millimeter seems essential. Otherwise, the text will look as if it was drawn without lifting the pen at all.

In addition to these essential requirements, a good pen tracking system should allow for simultaneously tracking multiple pens, distinguishing different pens or users and reporting the angle between pen and surface. The tracking system should be affordable and able to cope with changing lighting conditions, different users, and different sizes of tables. Pens should have an ergonomic shape and a small tip that occludes little of the surface. They should be low-cost and require little to no power.

In practice, implementations always have to find a trade-off between accuracy, latency, robustness, versatility, and cost.

3 RELATED WORK

TipTrack is a tracking technique for pens on arbitrary, unmodified planar surfaces. In this section, we discuss the state of the art in pen-tracking techniques on pen-sensitive and unmodified surfaces. For unmodified surfaces, we discuss both inside-out tracking (the pen senses its position) and outside-in tracking (sensors in the environment track the pen's position).

3.1 Pen tracking techniques for modified surfaces

3.1.1 Pen-sensitive screens. Pens (and pen-like devices) have been one of the earliest input devices for interactive digital systems, predating mouse and touch screen. The OA-1008 consoles developed for the SAGE system in the 1950s [9] allowed radar officers to select information on the screen via a light gun. Sutherland's seminal *Sketchpad* prototype [31] was operated via a light pen. The RAND tablet [6] used capacitive sensing for tracking a pen with high resolution.

Nowadays, the most common technologies employed in graphics tablets and pen-sensitive screens are resistive sensing, capacitive sensing, or electromagnetic resonance sensing. While these technologies offer high resolution and robust tracking at reasonably low cost, they require a sensing layer that covers the whole interactive surface. The pen must be in contact or close proximity to the sensing layer. Therefore, these technologies are not well suited for pen input on unmodified tabletops.

3.1.2 Pen tracking on non-screen modified surfaces. The *RetroDepth* system by Kim et al. [15] captures silhouettes of hands and objects in a 3D space using a stereo-pair of infrared cameras and multiple

IR-LEDs. The authors also demonstrate pen input by identifying the pen tip based on the pen outline. This approach necessitates a special retro-reflective material attached to the background surface, however.

InfinitePaint by Fender et al. [10] is a virtual reality application where users can create digital paintings using a real wet brush. A single camera tracks strokes of the brush on a special paper that temporarily turns black upon contact with water. This allows for precise stroke detection as long as the user does not paint over the same area twice in quick succession.

3.2 Pen tracking techniques for unmodified surfaces: Inside-out tracking approaches

A number of specialized sensing techniques have been developed for tracking pens on large, unmodified surfaces. They can be grouped into inside-out and outside-in tracking approaches. With inside-out approaches, a pen has an embedded sensor system inside that continuously captures some property of the environment around it and thus infers the pen's position.

A well-known example are pens with embedded cameras such as the *Anoto Livescribe* Smartpens[1], Neopen[2] or TipToi[3]. These commercial products contain a high-speed IR-sensitive camera that captures a unique dot pattern printed on a sheet of paper or any other surface [4]. From the arrangement and relative position of the dots, the sensor can determine the pen's position relative to the origin of the coordinate system in use. These pens offer high resolution and low latency but are relatively expensive and require special paper with a proprietary pattern printed on it. *PenLight* by Song et al. [30] features an Anoto pen in the context of a projected augmented reality tabletop. The authors explore possible use cases of a pen sized projector attached to the digial pen.

FlashPen by Romat et al. [27] contains a commercial mouse sensor in its tip which allows for tracking relative movement of the pen tip on arbitrary surfaces with very high resolution (micrometer range) and very low latency (low millisecond range). However, in addition to the sensors inside the pen, a separate external tracking system for determining the pen's absolute initial position is required.

DeltaPen by Lüthi et al. [18] is a pressure sensitive digital pen that allows for precise tracking of translation and rotation input on uninstrumented surfaces. A pressure sensor detects contact with the surface and two optical flow sensors measure motion and rotation. The device requires a permanent USB-connection to a computer for communication and power which limits its flexibility.

An inside-out tracking approach that does not rely on sensors in or near the pens tip is *PenSight* by Matulic et al. [22]. The system features a downward-facing fisheye-camera attached to the top of a pen. While this approach enables multiple tracking possibilities of the pen's surroundings, including the detection of the current user, as well as hand pose and gesture detection, it does not allow for the tracking of the pen-tip itself.

[1]https://www.anoto.com/solutions/livescribe/
[2]https://neosmartpen.com/
[3]https://www.ravensburger.de/de-DE/entdecken/tiptoi

3.3 Pen tracking techniques for unmodified surfaces: Outside-in tracking approaches

With *outside-in* tracking, sensors are mounted in the environment and capture the pen's position. The pen may either be unmodified or equipped with simple markers to aid tracking. While electromagnetic sensing techniques could be used in principle, the large surface and the presence of metallic objects on the table limits achievable resolution. Therefore, researchers have primarily pursued optical outside-in tracking approaches.

WebcamPaperPen by Pfeiffer er al. [26] tracks the tip of an unmodified ballpoint pen using a webcam. To distinguish between hover and contact, the distance between the pen tip and its shadow is measured. The setup requires a white surface as a backdrop to reliably find the pen tip and a single spot light to provide the shadow. It is therefore rather susceptible to changes in lighting conditions. Pfeiffer et al. seem to be the first to explicitly mention the *'serif' effect* that occurs when the minimum hover distance is to high. When the pen is lifted from the surface after a stroke, it still draws a small additional line connected to the last stroke. The authors do not report latency, minimum hover distance or spatial resolution.

Imad et al. [13] track the tip of a pen with colored markings at top and bottom end using a stereo camera setup. They report an achievable spatial resolution of 7 mm in the x axis and 1 mm in the y axis. No further performance metrics are reported. The figures in the paper indicate very coarse and noisy tracking, however.

3.3.1 Tracking approaches based on infrared light. Multiple research projects use the optical sensor in *Nintendo Wii Remotes* [5, 16] for tracking battery-powered pens with embedded IR LEDs. The sensor, made by PixArt, has an update rate of up to 100 Hz and an interpolated resolution of 1024x768 px. It reports the coordinates and brightness of up to four IR light sources. A pen with an internal IR LED as tip projects IR light onto close-by surfaces. When the tip is very close to the surface, the small hot spot generated by the pen can be tracked using the Wii Remote or any other IR-sensitive camera. Lee [16] required users to press a button on the pen to activate the LED. They do not report latency, update rate, or physical resolution.

By using a pen with a pressure sensitive tip, Chen et al.'s approach [5] 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. Hover events could not be detected. The processing pipeline had a significant latency of about 150 ms.

Lee at al. [17] placed an infrared camera next to a projector and used a custom infrared pen to turn the projection area on a wall into a digital whiteboard. The user needs to press a button on the pen to activate the IR LED. Their approach requires no hardware on or under the projection surface and could be adopted to a horizontal tabletop setting. Like the Wii-Remote-based solutions, this approach can not distinguish between then pen tip hovering above or touching the surface.

Margetis et al. [21] propose an approach that tracks an IR pen in a 3D space above a tabletop using stereoscopic images from two calibrated cameras. Resolution is about 2 mm, update rate is 30 Hz. Latency is not reported. The authors claim that hover/click states can be distinguished but do not go into detail. A user study

evaluated mainly the usability of the whole application. It is not reported whether the system supports handwriting.

In the paper *RetroSphere* [1] by Balaji et al., the authors present a pen-like passive controller called "RetroPen". The pen consists of two retroreflective markers that are tracked by AR glasses featuring a stereo pair of infrared blob trackers and infrared LED emitters. Contact to a surface can be detected due to a decrease in distance between the two markers because of a spring integrated into the pen. The authors claim a tracking accuracy of about 96.5% with errors around 3.5 cm over a 100 cm tracking range which does not allow for precise handwriting input that can be compared to using a normal ballpoint pen.

3.3.2 Marker-based tracking approaches. A common way to tag and optically track objects are 2D markers, such as *ArUco* [11] . They can even be made invisible to the human eye by 3D printing them using NIR-fluorescent filament [7, 8].

DodecaPen by Wu et al. [36] does not rely on an infrared-emitting pen. Instead, the rear end of the pen is augmented with a small dodecahedron containing printed ArUco markers. A 1.3 MP RGB camera captures the marker position and orientation with an update rate of 60 Hz. Through various measures, DodecaPen achieves a resolution of 0.5 mm. However, the position of the pen tip needs to be inferred from the position and orientation of the dodecahedron. The system requires good, constant lighting conditions for reliable marker tracking. End-to-end latency of the system is not reported. Wu et al. also give a short overview of further tracking techniques.

3.4 Summary

In summary, many fast and robust pen tracking solutions are available commercially. However, these typically require a sensing layer or an overlay with markers. Far fewer solutions – mostly research prototypes – support pen tracking on unmodified surfaces, such as tabletops. While tracking the tip of an unmodified pen with RGB(-D) cameras is possible, resolution and robustness are very limited. DodecaPen tracks markers on the far end of the pen which seems far more robust and accurate. However, for all of these systems, image processing is non-trivial and the system is very susceptible to changing light conditions or low ambient light. Therefore, many implementations use cameras to track hot spots generated by IR LEDs embedded into the pen. These systems allow for fast, robust tracking. However, they do not distinguish between hover and contact and offer only low spatial resolution.

Unfortunately, latency and minimum hover distance are rarely reported in previous work which makes it hard to determine practical performance. However, demo videos of these systems typically show end-to-end latencies of more than one hundred milliseconds.

4 TIPTRACK IMPLEMENTATION

Our goal with TipTrack was to design a reasonably robust, fast, and affordable system for tracking pen input on an unmodified tabletop. The focus is on offering an approach that is replicable, of immediate practical use, and can be adapted to different use cases. To this end, we made the following design decisions:

- only track the tip of the pen, not the whole pen
- only track on the surface, not in 3D space
- use a battery-powered, IR-light-emitting pen to ensure consistent tracking even in low-light conditions
- keep the IR LED on even when the pen is lifted from the table to support a *hover* state and remove the need for a switch within the pen which might alter the tactile properties of the pen tip
- use a combination of machine learning and heuristics for robust hover detection and tracking
- use two cameras on the left and right side of the table with overlapping fields of view in order to mitigate occlusion by the hand holding the pen , but do not necessarily use them for simultaneous stereo tracking
- use a PMMA tip (1 mm) in a commercial marker-pen case in order to offer a natural writing experience

These design decisions are explained in more detail later on. They also result in a number of inherent limitations which we discuss afterwards.

TipTrack operates as follows: An IR LED embedded within a battery-powered pen emits light. This light is emitted through the pen's lead which is made of clear PMMA. Most of the light is emitted through the front of the tip, some through the side. If the pen is close to a surface, the frontally-emitted light bounces off that surface. Two synchronized IR cameras mounted above the table capture an image of the reflected light. A Python script then processes each image pair in real-time. The pen's position is determined by the position of the bright spot of infrared light emitted by the pen. Around each bright point in the captured images, an area of 48 by 48 pixels is extracted (Fig. 7). A custom-trained convolutional neural network distinguishes between hover and touch states based on those regions of interest. Pen state and position are then sent to a simple renderer written in C++. Alternatively, input events are generated and passed on to the operating system via Linux' evdev framework. The overall architecture of the system is shown in Figure 4.

4.1 Hardware

For our reference implementation of TipTrack, we use a projected augmented reality setup with two infrared cameras, a 4k video projector, as well as our custom infrared pen, and a reasonably powerful PC. Cameras and projector are mounted above the tabletop on a mobile truss system (Fig. 2).

4.1.1 Projector/Camera Setup. We use an *Optoma UHZ50* projector[4] to project an interactive application onto a table's surface with the projection covering an area of 113 cm × 63 cm. This projector can produce 4k images (3840 × 2160 pixels) at 60 fps or FullHD images (1920 × 1080 pixels) at 240 fps. According to the manufacturer, this projector has a low input lag of 16 ms in 4k mode and 4.9 ms in FullHD mode. We separately verified these values.

To track the infrared pen on the table, we use two *FLIR BFS-U3-23S3M-C* cameras[5] with 8 mm lenses (45 mm full frame equivalent) and 850 nm IR pass-through filters. Cameras are mounted at a distance of 120 cm above the table and capture its surface from opposite sides. We use telephoto zoom lenses and mount the cameras as high

[4] https://www.optoma.com/ap/product/uhz50/
[5] https://www.flir.eu/products/blackfly-s-usb3/?model=BFS-U3-23S3M-C

Figure 2: Projected augmented reality setup for our reference implementation of TipTrack. Two cameras with infrared filters (1) and a projector (2) are mounted on a truss system. They are set up to project an interactive application onto a table (3) from above.

above the surface as possible because occlusion of the pen tip becomes less likely the tighter the field of view. Both cameras capture 1920 × 1200 pixel 8 bit monochrome images at 158 Hz. Camera triggers are synchronized via cable. By mounting 850 nm IR filters on the camera lenses, ambient visible light gets blocked almost entirely, so ideally only the infrared light from the pen tip is visible in the camera images. This way, artificial room light, as well as the projection do not influence pen tracking. While price and specifications of these cameras are higher than current consumer webcams, they are still rather affordable and representative of expected webcam performance in the near future.

4.1.2 System setup and calibration. To allow for precise projection of drawn content without offset, projector and cameras need to be calibrated to each other. We start by undistorting each camera frame after determining the intrinsic parameters of each camera with the help of a checkerboard calibration pattern. In the next step, we use a custom calibration tool which allows for selecting the four corners of the projection in the camera frame. This process is repeated for each of the two cameras. By using these points to calculate a homography, points can be transformed from the camera's coordinate system to the projection's coordinate system by multiplying their coordinates with a transformation matrix. While more sophisticated approaches for projector-camera calibration exist, this proved to be sufficient for our prototype. As long as the table stays approximately at the correct height, no further calibration is needed.

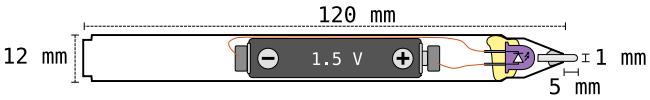

Figure 3: Schematic of the infrared pen. A battery-powered infrared LED shines light through a light guide which is used as the pen tip. All components are included in an off-the-shelf felt tip pen. The battery is connected to the LED with thin copper wires and magnetic contacts.

4.1.3 Infrared Pen. We built a pen emitting infrared light by placing an infrared LED[6] inside a black *edding 400* permanent marker case (Fig. 3). A PMMA light guide with a diameter of 1 mm and a sanded end is hot-glued inside the pen's tip. It directs the IR light towards the drawing surface, lights up itself, and provides a writing experience similar to that of an ordinary ballpoint pen. We compared side-emitting and conventional light guides and found that both work similarly well. 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. A fully charged battery lasts around eight hours before the LED becomes too dim and tracking gets less reliable. A *joule thief* circuit could be used to increase LED brightness for partially discharged batteries, however, it would have to be built small enough to fit inside the pen. We did not yet pursue this optimization.

4.2 Image processing

Both cameras capture images at 158 frames per second. TipTrack processes these images to find and extract each pen's spot so it can determine its coordinates and classify whether the pen is touching the surface or not. All image processing is implemented in Python 3.8 and *OpenCV 4* [3]. First, we find the brightest spots in each camera frame and extract a 48 × 48 pixel region around them. By applying a threshold keeping only the brightest pixels and calculating the center of gravity of the remaining shape, we determine the pen tip's position in the camera image.

We transform the resulting coordinates into the projection's coordinate system by multiplying them with the transformation matrix from the camera calibration. This procedure is significantly more time-efficient than applying the homography to the whole camera frames during runtime. Processing one set of frames from both cameras takes between 3.5 and 5.5 milliseconds on our system[7].

4.3 Pen State Classification

To distinguish between the pen states *draw* and *hover*, we use a convolutional neural network (CNN).

We trained a convolutional neural network (CNN) to distinguish between the pen states *draw* and *hover*. The network is implemented in *TensorFlow 2.0*[8] using the *Keras* API[9]. The network structure can be seen in Fig. 5. Hyperparameters of the network were optimized via grid search.

[6] SFH 4550, 850 nm, 2800 mW/sr, φ 3°
[7] Ubuntu GNU/Linux 20.04, Intel i9 9900k @3.6 GHz, NVIDIA GeForce RTX 2060
[8] https://www.tensorflow.org/
[9] https://keras.io/

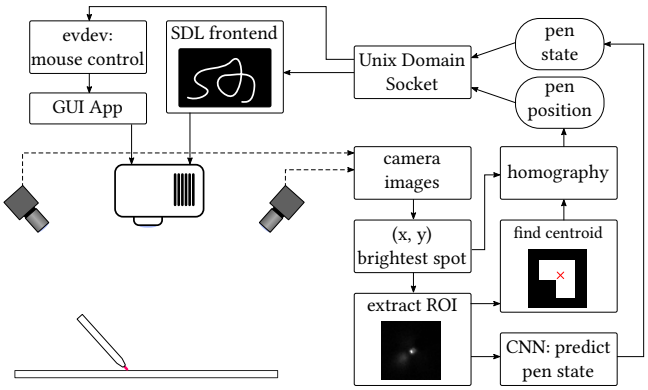

Figure 4: TipTrack processing pipeline. Cameras capture the infrared spot emitted from the pen's tip. An area around the resulting bright spot is extracted for pen state classification. Pen coordinates are calculated by finding the centroid of the bright spot, and then transferred to the output coordinate system. A CNN is used to classify the pen state as *draw* or *hover*. A drawing application renders drawn lines which are then projected onto a table.

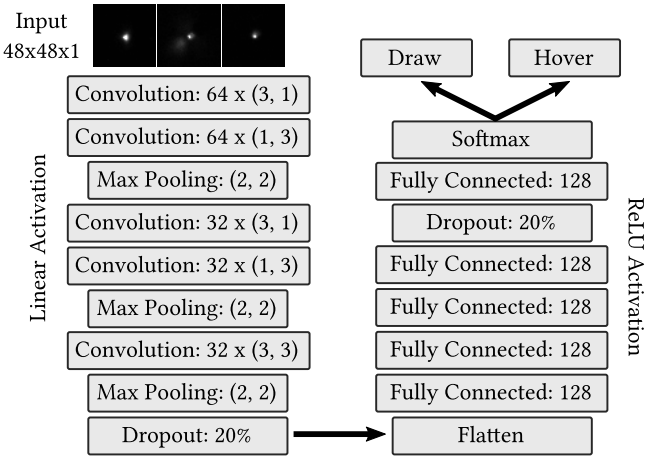

Figure 5: Network architecture of the convolutional neural network used for pen state classification.

Training data was acquired by manually moving the infrared pen across the table, continuously capturing images with both cameras. Those images are cropped to a 48×48 pixel region around the brightest spot and saved with the corresponding label *draw* or *hover*. Capturing a whole training data set of 12000 images this way takes about 10 minutes. Pen states are easier to distinguish on bright images as differences between the images become more apparent. However, with long exposure times, quickly moving the pen introduces motion blur which deteriorates classification. We found a sweet spot of 0.8 milliseconds exposure time and 18 dB sensor gain. With those settings, contrast is high enough while image noise is still acceptable. Before training, we randomly select 20% of the images for each category to be used as a validation data set. The remaining 80% of the images is augmented by rotation in 90° steps and flipping, which increases the amount of training data by a factor of eight.

Figure 6: TipTrack can reliably track the infrared pen on different flat and smooth surfaces, such as paper, cardboard, wood, and fabric.

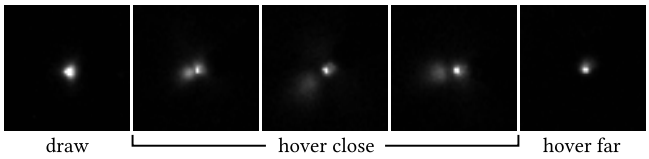

Figure 7: Cropped and re-scaled sensor images. If the pen is hovering over the surface (> 2 mm), a light cone is clearly visible. When touching the surface, the light cone disappears. Close hovering (< 2 mm) looks very similar to touching. Hovering far from the surface (> 15 cm) also looks similar, but the spot is smaller and less bright.

We trained our network with a batch size of 128 using an *Adam* optimizer with categorial crossentropy as the loss function. If validation loss stops decreasing for two epochs, learning rate is reduced by 80%. After six epochs, the model reached an accuracy of 97.85% on the validation data set. Using *TensorFlow lite*[10], the model achieves a mean prediction time of 1.3 ms with rare outliers.

For our final model, we collected all training data on a light, untreated wooden tabletop (see Fig. 2). To assess the model's performance on different surface materials without re-training, we informally tested 11 additional materials (six of them depicted in Fig. 6). We achieve identical tracking and hover-detection performance for the following surfaces: untreated dark wood, white fibreboard (acrylic paint), coated fibreboard with a wood texture, paper, cardboard, and a green cutting mat. A slight increase in misclassified pen events is visible on dark cotton fabric. Tracking does not work at all on transparent (glass, acrylic) or highly reflective materials (metal, dark wood with glossy coating).

[10] https://www.tensorflow.org/lite

4.4 Drawing Application and Human Interface Device

We implemented an application that displays lines drawn by users as a frontend for our system. We used C++ with the *SDL2*[11] framework as it adds only a small amount of latency even when rendering complex scenes [29]. This application receives all events via a UNIX domain socket[12], and draws points and lines on a black canvas, which is then projected onto the table.

To control arbitrary GUI applications with TipTrack, we implemented a program that translates the pen state to mouse events. Pen state and coordinates are used in combination with *python-evdev*[13] to simulate a virtual input device and control the mouse cursor. Hovering only moves the cursor, whereas a short press or dragging gesture triggers the left mouse button and a long press at a constant position triggers the right mouse button.

5 EVALUATION

In order to better understand and document the properties of our current TipTrack implementation and of the concept in general, we conducted three studies:

- In a lab study, we investigated subjective qualities of drawing with TipTrack.
- In an extensive technical benchmark, we measured latency and tracking resolution of our setup.
- Through multiple in-the-wild evaluations we gathered qualitative feedback and quantitative performance data. Participants were HCI researchers and practitioners at an academic conference as well as children and teenagers at two Open Lab Days.

5.1 Subjective Qualities

In order to learn more about how writing with TipTrack feels, we asked 15 participants to draw simple and complex shapes with TipTrack. Participants were aged 24–33 years (mean 28); two were left-handed. All participants were unfamiliar with the system and we asked them to pay attention to the system's performance and give us feedback afterwards.

At the beginning and the end of the session, participants had to trace a number of crosses with the pen which were projected onto the tabletop. This was intended to gather data on accuracy and precision of the system. As we significantly improved TipTrack performance afterwards, these metrics are obsolete. Thus, we do not report them.

Then we displayed a random phrase from MacKenzie and Soukoreff's phrase set [19] and asked participants to copy it inside a projected box in the center of the table. This process was repeated five times each for three differently sized boxes (5 cm, 4 cm, and 3 cm height) to stimulate size variation in user's handwriting. Finally, participants had three minutes to freely draw on the table. After all tasks were finished, we asked participants to provide feedback on our system in a short interview.

At this stage, our system had a slight offset between physical pen and tracked position, which got stronger near the table's borders (up to about 5 millimeters). Twelve of our 15 participants found this offset distracting. However most of them got used to it quickly.

A small yellow circle indicated the pen's current position while it hovered above the table. Eleven participants found this feature useful, for example to counteract said offset, three were indifferent and one participant found it irritating.

For eight participants, the shadow cast by the drawing hand occluding the projection was a problem when using the system. This was especially apparent for left-handed users while writing, and for right-handed participants during the cross-hatching tasks, as the crosses were projected onto their hands instead of the table.

Additionally, one participant occluded the camera's line of sight with their head and two participants noticed that hovering the pen very closely above the surface could wrongly be detected as drawing.

Seven of the fifteen participants liked how writing feels with our system. Four found it acceptable and four would have preferred a smoother surface. We asked which physical pen our IR pen resembled most closely (multiple answers possible). Most participants compared it to a felt-tip pen because of the haptics (7) or a ball-head pen because of the rough writing experience (6). Other pen types mentioned were a fountain pen (4), tablet (2), a sharpie (1) or a gel pen (1). Participants also asked for more drawing features, such as erasing drawn lines or changing line weight and color. Figure 8 shows examples of the phrases and sketches written and drawn by participants.

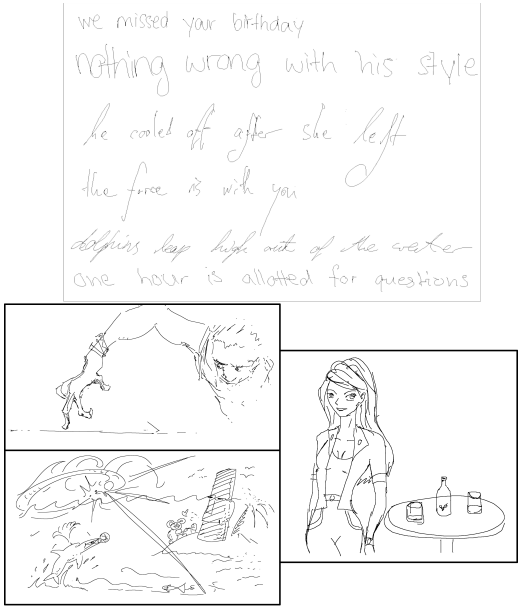

Figure 8: Examples for TipTrack's capabilities for drawing and handwriting. Top: phrases written by users during the handwriting task. Bottom: pictures created by participants during the drawing task. We increased the line weight for better visibility at this small size.

[11] https://github.com/libsdl-org/SDL
[12] https://man7.org/linux/man-pages/man7/unix.7.html
[13] https://python-evdev.readthedocs.io/

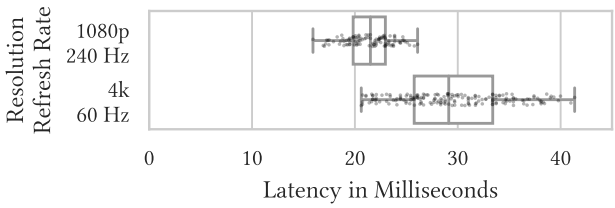

Figure 9: End-to-end latency of TipTrack with different projector settings.

5.2 Latency Measurements

To measure the end-to-end latency of our system, we used a modified version of YALMD, an open-source latency measuring device [28]. For each measurement pass, YALMD turns on an infrared LED placed on the tabletop and starts a microsecond-resolution timer. Once the slightly modified TipTrack software detects the bright spot from the LED in the camera image, it conducts through all of the image processing and classification steps described in section 4.2. During our latency measurements, TipTrack projects a large white rectangle instead of a pen stroke. A photo resistor in YALMD detects the change in brightness once this rectangle is projected onto the surface and stops its internal timer. This measurement is repeated 1,000 times for each measurement run.

We measured a mean end-to-end latency of 21.4 ms (16.0 – 26.0 ms) in 1080p 240 Hz mode and a mean end-to-end latency of 29.6 ms (20.6 – 41.3 ms) in 4K 60 Hz mode (Fig. 9). In comparison, state-of-the-art research prototypes which combine high-end projectors and cameras reach an end-to-end latency below 10 ms (e.g., 6 – 7 ms, [23]). For most interactive systems based on commercial off-the-shelf hardware, end-to-end latency is not reported at all. For SciSketch (2014, [5]), Chen et al. report a latency of 150 ms, which they attribute in part to a slow Bluetooth connection and software.

To find performance bottlenecks, we measured partial latencies in our system. We used the *LeoBodnar Video Signal Input Lag Tester*[14] to measure the response time of our projector. In 4k 60 Hz mode, the projector has an input lag of 16 milliseconds.[15] In FullHD 240 Hz mode, the projector has an input lag of 4.9 milliseconds. Those measurements match the manufacturer's specifications.

To determine the latency introduced by the cameras, we measured the end-to-end latency of the camera-based setup as described above. Then we conducted the same measurements again but instead of capturing camera frames, we triggered the processing pipeline via auto-generated input events via a USB mouse [35]. We used a *Logitech G5*, which has a rather consistent latency of 2.2 milliseconds (std: 0.2 ms) [35]. On average, latency with the camera was 5 milliseconds higher and there was no influence on latency variance, resulting in an estimated latency of 7 to 7.4 ms for the camera. As the camera produces a new image every 6.33 milliseconds, this value seems plausible.

[14] www.leobodnar.com/shop/index.php?main_page=product_info&products_id=212
[15] As this measuring device only outputs a 1080p signal, latency for actual 4K input may be higher.

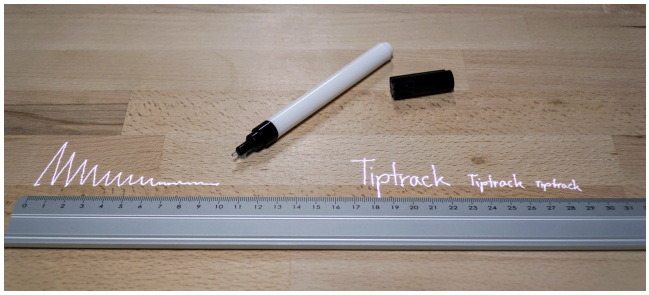

Figure 10: Text written with TipTrack remains readable up to a minimum letter height of approximately 4 mm.

The remaining 6 – 7 milliseconds of latency are caused by predictions of the CNN (mean: 2.6 ms for predicting two detected pen events), image processing, and rendering.

5.3 Tracking Resolution

With the current system, fluid handwriting on the surface is possible up to a minimum letter height of approximately 4 mm (see Fig. 10). In this section, we present a systematic benchmark of TipTrack's horizontal and vertical tracking resolution.

To measure the spatial tracking resolution of our system, we used an *AxiDraw V3*[16] robot to move the pen across the surface in a controlled manner (Fig. 11, left). The *AxiDraw* has a resolution of 88 motor steps per millimeter, which is more than sufficient for this test. For our test, we used the setup depicted in Fig. 2 with the projection covering an area of 113 cm × 63 cm, resulting in a pixel size of 0.29 mm × 0.29 mm. The cameras were zoomed in to just cover the projected area. We moved the pen horizontally for 24 steps in different increments: 10 mm, 5 mm, 2 mm, 1 mm, 0.5 mm, 0.2 mm, 0.1 mm. We then calculated the offset between the pen's theoretical position and the measured position (Fig. 11, right). To this end, we normalized all measured trajectories by setting their start and end points to 0 respectively 100 and calculating the distance from each measured point to a straight line between start and end. We found that spatial tracking works robustly down to 0.5 mm steps. However, even for 0.1 mm – the smallest step size we used – relative error peaks at around 40%, which is 40 micrometers off.

The vertical resolution, i.e., the minimum distance between pen and surface that is reliably classified as *hovering*, is important for use cases such as writing and drawing. To measure it, we mounted the *AxiDraw* vertically so it could move the pen back and forth, as well as up and down with sub-millimeter precision. We attached our infrared pen to the robotic arm so that it just touched the table's surface (Fig. 12, left). We programmed the *AxiDraw* to raise the pen in 0.1 mm steps and draw a short line at every vertical position. As there is slight mechanical play in the joints and belts of the AxiDraw when mounted vertically, it would be difficult to precisely calibrate a zero point at which the pen tip just touches the surface. Therefore, we used a contact microphone attached to the table to exactly determine the moment when the pen leaves the surface. This microphone registers the sound of the pen scratching along

[16] https://shop.evilmadscientist.com/productsmenu/846

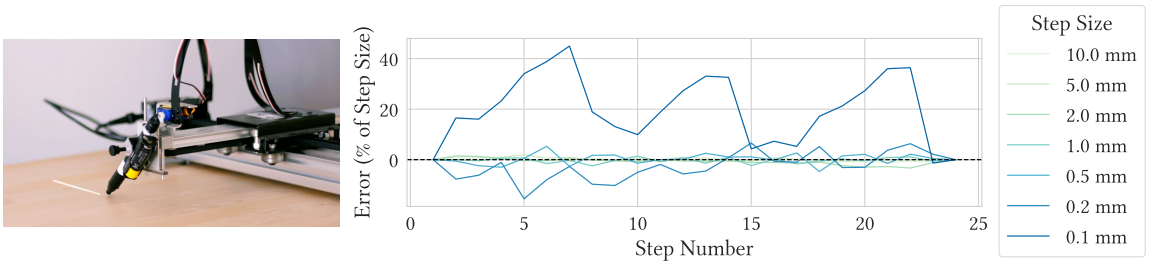

Figure 11: Measurements of spatial tracking resolution. Left: AxiDraw drawing robot moving the pen across the table for our resolution measurements. Right: Relative deviation of the pen's measured position from a straight line when moving the pen across the table in different step sizes. The fluctuations in accuracy for small step sizes indicate that the pen was moved in increments smaller than the system's tracking resolution.

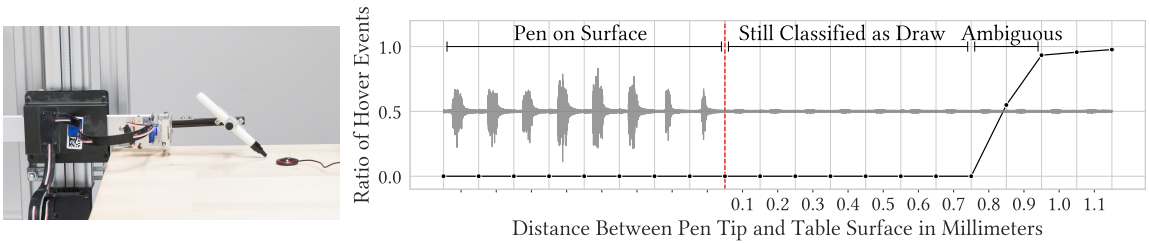

Figure 12: Results of our test for vertical accuracy. The wave in the center represents scratching sounds of the pen on the table's surface. Once the pen physically leaves the table surface, the scratching sounds stop. However, the pen has to be 0.9 mm above the table's surface to be confidently classified as hovering.

the table's surface. Once the pen no longer touches the surface, the audio amplitude decreases significantly. We define the first vertical position where no scratching sound is recorded anymore as the starting point for our measurements. By incrementally raising the pen and checking our system's predictions, we could determine at which vertical position TipTrack starts to classify the pen state as *hover*. Our measurements show that up to a vertical distance of 0.8 mm, the pen is registered as touching the surface, i.e., its state is classified as *draw* (Fig. 12, right). Between 0.8 and 0.9 mm, there is ambiguity in predictions with an almost even distribution of *draw* and *hover*. Above 0.9 mm, TipTrack consistently classifies the pen state as *hover*.

5.4 In-the-wild Evaluation via Public Demonstrations

Whether an input device feels just right or too sluggish and imprecise is ultimately a subjective impression. To gather subjective feedback on TipTrack, we presented the prototype at three public events with different audiences:

- an open lab day with about 250 students between eight and sixteen (July 2022)
- an open lab day with about 300 students and preschoolers aged between five and sixteen (July 2023)
- the demo session of an HCI conference where over 100 attendees interacted with TipTrack (September 2022)

The goal was to see how users interact with the system, how robust it is in practice, and to gather qualitative feedback on possible improvements and use cases.

5.4.1 Open Lab Day 2022. TipTrack was one of several interactive exhibitions at an open lab day which 250 students from several local schools attended. More than one hundred students visited our demo. A drawing application (MyPaint[17]) was shown in full-screen mode. Students could draw and scribble on the table using the pen. While we had also prepared alternative demos, visitors were so engaged with the drawing application that we never switched demos. Even though the setting did not allow for structured observations, we observed that students enjoyed drawing on the table. They had a strong sense of agency and were proud of things *they* produced. As there was only one pen available, students interacted with each other and took turns drawing and watching.

As our TipTrack setup was located in a room with large windows during this event, lighting conditions were changing continuously over the course of the day. Therefore, tracking quality varied significantly and we had to re-train the classifier multiple times to adapt to new conditions.

5.4.2 Open Lab Day 2023. Similar to the Open Lab Day of the previous year, we exhibited TipTrack for about 300 students from several local schools and a local preschool. In addition to a simple drawing application that allowed to playfully try out the system, we also prepared a short dexterity game which we also used to

[17] http://mypaint.org/

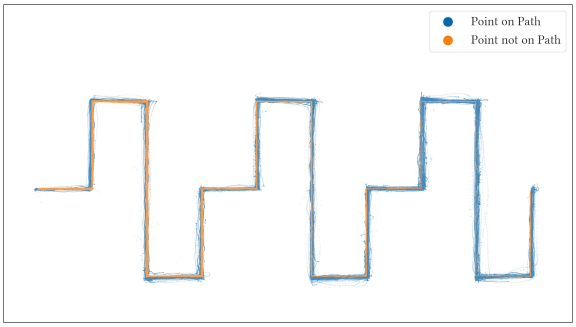

Figure 13: Aggregated paths drawn by participants in our 2023 demo. The canvas represents the whole projected area in 4K resolution. The path to follow gets smaller from left to right.

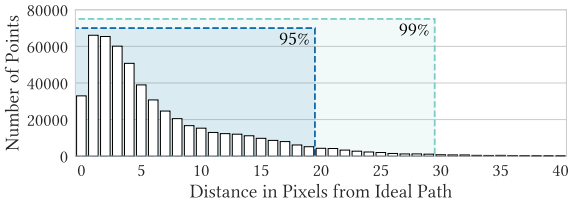

Figure 14: Euclidean distance of all points in paths drawn by participants to the ideal path. One pixel is 0.29 mm wide. Therefore, 95% of points are less than 6 mm away from the ideal path.

collect data about the system's performance. The game's objective was to follow a given path from start to finish as accurately and quickly as possible. The path was projected onto the tabletop and consisted of three connected segments with progressively decreasing path widths (16 px, 8 px and 4 px at 0.29 mm per pixel). Current accuracy and a timer were displayed on top. The timer starts with the player drawing a line through the start region at the left hand side and stops when the pen reaches the target region on the right side. During this game, we recorded all pen events to later evaluate TipTrack's tracking accuracy in a real-world setting. We asked students for their age but not for any further demographic information as it was not feasible to obtain informed consent in this setting.

In total, 65 students (41 children, 5 – 11 years and 24 teenagers, 12 – 16 years) played the game. Players were rewarded with sweets for their participation, regardless of their final score. Task completion time varied greatly among participants, especially between children (M: 101.41 s, SD: 61.36 s, range: 10.14 – 249.59 s) and teenagers (M: 32.60 s, SD: 36.70 s, range: 9.67 – 184.37 s). While most children clearly focused on following the path as accurately as possible, teenagers tried to finish the task quickly while only roughly following the path. Notably, the deviation of drawn lines from the ideal path did not change as the task's difficulty increased. Figure 13 depicts all paths drawn over the course of the day.

To evaluate how precisely novice users can follow a path using TipTrack, we calculated the euclidean distance between each point

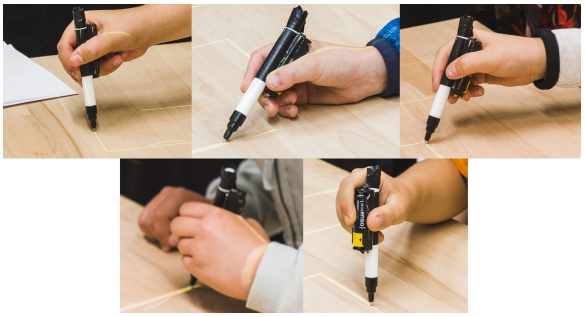

Figure 15: We observed that some children held the pen in idiosyncratic ways, for example at the rear end or in very steep angles. Even though the external battery of the pen we used in our field study certainly influenced users' grip, we also observed this behavior when using a pen with an internal battery.

drawn by participants to an ideal path with a width of one pixel (Fig. 14). We found that 95% of points were within a distance of 20 px to the ideal path and 99% of points were within a distance of 30 px. As one pixel is approximately 0.29 mm wide, 95% of points were less than 6 mm away from the ideal path.

One important observation we could make was that young children in particular often hold the pen very differently from adults (nearly perpendicular to the surface or really close to the pen tip, see Fig. 15) causing additional occlusion of the pen tip. As this leads to reduced tracking accuracy, these cases represent a worst case scenario for TipTrack's performance.

5.4.3 Conference Demo Session. We presented TipTrack at the demo session of 'Mensch und Computer 2022', the largest German HCI conference [20]. Over 100 attendees interacted with our system which gave us the opportunity to get feedback in conversations with computer scientists, HCI researchers, and designers. Most visitors were impressed by the system's low latency and precise tracking. Some of them thoroughly tested how robust the tracking was by covering the pen with their hands, quickly moving the pen, trying to draw very precise lines, or hovering the pen very closely above the surface. Many attendees asked for a fully-featured drawing application: first and foremost a way to erase lines, but also to be able to change line width and color. A few people mentioned the problem of user's hands occluding the projection. Some of them proposed using a short-throw projector at the opposite side of the table to reduce occlusion. As there were no windows near our system, tracking worked very robustly for the full three hours of the event. However, the system crashed occasionally, requiring a restart of the Python backend. We fixed this issue after the event. TipTrack won the 'Best Demo' award at this conference by a wide margin with more than twice the votes of the second-placed demo. However, a contributing factor might have been that the demo was very accessible compared to more complex applications.

While feedback at exhibitions can not replace the results from controlled lab studies, it provides insights into public acceptance, usability problems, and suitability for different user groups. Positive reception by children, teenagers, and HCI experts suggests that

TipTrack might be useful and enjoyable for wider audiences. Our prototypical implementation proved to run very reliably through all three exhibitions. The technical issues we encountered were either fixed afterwards or suggest directions for future work (e.g., accidental occlusion of the camera's view or of the projection).

6 LIMITATIONS, FUTURE WORK, AND APPLICATIONS

We could show that TipTrack is capable of robust pen tracking even to the degree of enabling fine grained input needed for handwriting and drawing. However, the system still has some limitations that we hope to address in future iterations.

6.1 Current technical limitations of TipTrack

TipTrack's current implementation assumes a planar surface. When drawing on a non-planar surface or on objects above the tabletop (e.g., a book lying on the table), the projected strokes will be offset. As the hover detection is based on the IR hotspot, it still works reliably in these cases. In order to support drawing on non-planar or vertically offset surfaces, the vertical position of the pen would need to be tracked via a stereoscopic camera setup or other sensing methods. TipTrack already employs two cameras in order to provide reliable tracking even when one camera's view is occluded. In order to offer reliable 3D tracking, at least three cameras should be used.

Even though using a CNN for pen state classification yielded significantly higher accuracy than earlier experiments with sophisticated heuristics or support vector machines, this approach has a number of drawbacks. Firstly, the model is optimized for the exact cameras we use in our reference implementation of TipTrack. With other cameras, a different network architecture might perform better. Thus, the model's hyperparameters have to be optimized again when using different cameras. Additionally, as deep neural networks are a black box, the system is hard to debug when tracking accuracy starts to deteriorate during operation. From our experience, this happens often when the setup is exposed to direct sunlight. In this case, collecting a new set of training data and retraining the network can solve the problem temporarily. However, this is a time-consuming process blocking the system for at least 15 minutes and can be a show-stopper for practical applications where the setup is unattended.

Even though we could show that our TipTrack implementation supports handwriting, projection resolution and tracking accuracy are still limiting factors for how small one can write (see Fig. 10). Especially when writing very small, hovering very closely above the surface being wrongly classified as drawing can lead to unwanted lines within and between letters.

With our reference implementation of TipTrack, our goal was to find out how fast and accurate such a system can be built with today's hardware. Therefore, we used specialized (and therefore moderately expensive) hardware and did not focus on replicability in the first place. Even though it is possible to build a TipTrack setup with significantly cheaper components, this comes with the cost of higher latency and less accurate tracking. In future work, we will investigate how TipTrack can be implemented with cheaper components and how to compensate the loss of spatial and temporal resolution. One option would be to predict the pen's trajectory to reduce perceived latency.

6.2 Plans for future iterations of TipTrack

Though it has not been explicitly mentioned in this paper, the current version of TipTrack supports simultaneous operation with multiple pens.

As long as a pen does not leave the tracking area, its ID will not change. However, when a pen leaves and re-enters the tracked area, it will be assigned a new ID and all potential metadata, such as color and stroke width, gets lost. This problem could be addressed with more intelligent pens, for example by modulating the LED's brightness in unique patterns. However, this solution could negatively influence tracking accuracy and might require higher camera frame rates.

As we currently use standard AAAA batteries without a power supply circuit to power the infrared LEDs within TipTrack's pens, LED brightness decreases when the battery discharges. Battery life could be increased by including additional sensors, such as a pressure sensitive tip that only activates the LED once the pen touches the surface. Adding a switch or force sensor to the pen might also allow for easier detection of a hover state. A force sensor or deformable tip might also be used to add pressure-sensitivity to pen input, e.g., for drawing applications. However, this approach might introduce new problems, such as requiring more pressure on the surface and thus affecting the writing experience. Another solution would be to use a boost converter IC in combination with rechargeable batteries. Even though this does not solve the problem of having to regularly switch and/or recharge batteries, the LED's brightness would remain constant regardless of the battery's charge. We will evaluate different solutions to counteract battery drain and its effect on tracking quality in future studies.

TipTrack's capability to track the pen in a hover state could be used more extensively for things like preview and menu selection, similar to the hover feature of the Apple Pencil[18].

We currently use the camera frames solely to distinguish between draw and hover events. It seems possible to extract additional information, such as pen tilt and surface material, from the reflection produced by the pen's IR LED on the surface.

6.3 Applications

TipTrack is intended to act as a generic input device for tabletops. It supports both a hover and a touch state, and can emit pointer events to applications running on the same computer. Therefore, it can be used to control any application that can be operated by mouse, touch, or pen input. In addition, TipTrack supports drawing on the tabletop or arbitrary flat surfaces lying on the table.

Thus, TipTrack can be used in many different application scenarios, such as:

- collaborative interaction with digital content
- creating digital art
- interacting with and annotating physical documents and objects on the table
- interacting with projected interfaces on a workbench

[18] https://developer.apple.com/news/?id=23ksaoks

7 CONCLUSION

With TipTrack, we propose and characterize an architecture for fast and robust optical tracking of a pen tip on a tabletop. Our open-source proof-of-concept implementation has an end-to-end latency of about 20 – 30 ms which is much lower than previous approaches for tabletops and in the same order of magnitude as current state-of-the-art inductive graphics tablets. Most of this latency is caused by cameras and projector. Therefore, upcoming hardware generations with higher update rate and resolution will lower latency further.

Source code and technical documentation for TipTrack are available on GitHub[19] under open source licenses.

ACKNOWLEDGMENTS

This project was funded by the Federal Ministry of Education and Research (BMBF) via VIGITIA (16SV8287) . We thank the reviewers of this and previous submissions for their helpful comments and suggestions.

REFERENCES

[1] Ananta Narayanan Balaji, Clayton Kimber, David Li, Shengzhi Wu, Ruofei Du, and David Kim. 2022. RetroSphere: Self-Contained Passive 3D Controller Tracking for Augmented Reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (Dec. 2022), 1–36. https://doi.org/10.1145/3569479

[2] The Holy Bible. n.d.. *Exodus 25,23*. URL: https://www.kingjamesbibleonline.org/Exodus-25-23/.

[3] Gary Bradski. 2000. The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25, 11 (2000), 120–123. Publisher: Miller Freeman Inc..

[4] Peter Brandl, Michael Haller, Michael Hurnaus, Verena Lugmayr, Juergen Oberngruber, Claudia Oster, Christian Schafleitner, and Mark Billinghurst. 2007. An Adaptable Rear-Projection Screen Using Digital Pens And Hand Gestures. In *Proceedings of the 17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*. 49–54. https://doi.org/10.1109/ICAT.2007.12

[5] 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

[6] M. R. Davis and T. O. Ellis. 1964. The RAND tablet: a man-machine graphical communication device. In *Proceedings of the October 27-29, 1964, fall joint computer conference, part I on XX - AFIPS '64 (Fall, part I)*. ACM Press, San Francisco, California, 325. https://doi.org/10.1145/1464052.1464080

[7] Mustafa Doga Dogan, Raul Garcia-Martin, Patrick William Haertel, Jamison John O'Keefe, Ahmad Taka, Akarsh Aurora, Raul Sanchez-Reillo, and Stefanie Mueller. 2023. BrightMarker: 3D Printed Fluorescent Markers for Object Tracking. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, San Francisco CA USA, 1–13. https://doi.org/10.1145/3586183.3606758

[8] Mustafa Doga Dogan, Ahmad Taka, Michael Lu, Yunyi Zhu, Akshat Kumar, Aakar Gupta, and Stefanie Mueller. 2022. InfraredTags: Embedding Invisible AR Markers and Barcodes Using Low-Cost, Infrared-Based 3D Printing and Imaging Tools. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3491102.3501951

[9] R. G. Enticknap and E. F. Schuster. 1959. Sage data system considerations. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics* 77, 6 (Jan. 1959), 824–832. https://doi.org/10.1109/TCE.1959.6372899 Conference Name: Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics.

[10] Andreas Rene Fender, Thomas Roberts, Tiffany Luong, and Christian Holz. 2023. InfinitePaint: Painting in Virtual Reality with Passive Haptics Using Wet Brushes and a Physical Proxy Canvas. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–13. https://doi.org/10.1145/3544548.3580927

[11] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (June 2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005 01071.

[19] https://github.com/PDA-UR/TipTrack

[12] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. ACM Press, Atlanta, Georgia, USA, 581. https://doi.org/10.1145/1753326.1753413

[13] Fadi Imad. 2018. Real-Time Pen Input System for Writing Utilizing Stereo Vision. *Journal of Computers* (2018), 1000–1009. https://doi.org/10.17706/jcp.13.9.1000-1009

[14] Martin Kaltenbrunner and Ross Bencina. 2007. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*. ACM Press, Baton Rouge, Louisiana, 69. https://doi.org/10.1145/1226969.1226983

[15] David Kim, Shahram Izadi, Jakub Dostal, Christoph Rhemann, Cem Keskin, Christopher Zach, Jamie Shotton, Timothy Large, Steven Bathiche, Matthias Nießner, D. Alex Butler, Sean Fanello, and Vivek Pradeep. 2014. RetroDepth: 3D silhouette sensing for high-precision input on and above physical surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1377–1386. https://doi.org/10.1145/2556288.2557336

[16] 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.

[17] 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

[18] Guy Lüthi, Andreas Rene Fender, and Christian Holz. 2022. DeltaPen: A Device with Integrated High-Precision Translation and Rotation Sensing on Passive Surfaces. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–12. https://doi.org/10.1145/3526113.3545655

[19] 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 '03*. ACM Press, Ft. Lauderdale, Florida, USA, 754. https://doi.org/10.1145/765891.765971

[20] Vitus Maierhöfer, Andreas Schmid, and Raphael Wimmer. 2022. Demonstration of an Infrared Pen as an Input Device for Projected Augmented Reality Tabletops. In *Proceedings of Mensch und Computer 2022 (MuC '22)*. Association for Computing Machinery, New York, NY, USA, 584–586. https://doi.org/10.1145/3543758.3547519

[21] 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

[22] Fabrice Matulic, Riku Arakawa, Brian Vogel, and Daniel Vogel. 2020. PenSight: Enhanced Interaction with a Pen-Top Camera. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376147

[23] Leo Miyashita, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2018. MIDAS projection: markerless and modelless dynamic projection mapping for material representation. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 196:1–196:12. https://doi.org/10.1145/3272127.3275045

[24] Albert Ng, Michelle Annett, Paul Dietz, Anoop Gupta, and Walter F. Bischof. 2014. In the blink of an eye: investigating latency perception during stylus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1103–1112. https://doi.org/10.1145/2556288.2557037

[25] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, Pittsburgh PA USA, 187–196. https://doi.org/10.1145/1570433.1570469

[26] Gustavo Thebit Pfeiffer, Ricardo Guerra Marroquim, and Antonio Alberto Fernandes de Oliveira. 2014. WebcamPaperPen: A Low-Cost Graphics Tablet. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. 87–94. https://doi.org/10.1109/SIBGRAPI.2014.54 ISSN: 2377-5416.

[27] Hugo Romat, Andreas Fender, Manuel Meier, and Christian Holz. 2021. Flashpen: A High-Fidelity and High-Precision Multi-Surface Pen for Virtual Reality. In *Proceedings of the 2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, Lisboa, Portugal, 306–315. https://doi.org/10.1109/VR50410.2021.00053

[28] Andreas Schmid and Raphael Wimmer. 2021. *Yet Another Latency Measuring Device*. preprint. Open Science Framework. https://doi.org/10.31219/osf.io/tkghj

[29] Andreas Schmid and Raphael Wimmer. 2023. Measuring the Latency of Graphics Frameworks on X11-Based Systems. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23)*. ACM, Hamburg Germany, 1–7. https://doi.org/10.1145/3544549.3585779

[30] Hyunyoung Song, Tovi Grossman, George Fitzmaurice, François Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. 2009. PenLight: combining

a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Boston MA USA, 143–152. https://doi.org/10.1145/1518701.1518726

[31] Ivan E. Sutherland. 1964. Sketchpad a Man-Machine Graphical Communication System. *SIMULATION* 2, 5 (May 1964), R–3–R–20. https://doi.org/10.1177/003754976400200514

[32] Andrey Vakunov, Chuo-Ling Chang, Fan Zhang, George Sung, Matthias Grundmann, and Valentin Bazarevsky. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality (CV4ARVR)*. Seattle, WA, USA. https://doi.org/10.48550/arXiv.2006.10214

[33] Pierre Wellner. 1993. Interacting with paper on the DigitalDesk. *Commun. ACM* 36, 7 (July 1993), 87–96. https://doi.org/10.1145/159544.159630

[34] Raphael Wimmer. 2017. Some Thoughts on Ergonomic, Practical, and Economic Properties of Interactive Tabletops. In *Workshop "The Disappearing Tabletop" at ISS 2017*. https://epub.uni-regensburg.de/36401/

[35] Raphael Wimmer, Andreas Schmid, and Florian Bockes. 2019. On the Latency of USB-Connected Input Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300650

[36] Po-Chen Wu, Robert Wang, Kenrick Kin, Christopher Twigg, Shangchen Han, Ming-Hsuan Yang, and Shao-Yi Chien. 2017. DodecaPen: Accurate 6DoF Tracking of a Passive Stylus. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, Québec City QC Canada, 365–374. https://doi.org/10.1145/3126594.3126664

[37] 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*. ACM, Niagara Falls Ontario Canada, 85–94. https://doi.org/10.1145/2992154.2992173

[38] Tim zum Hoff, Sabrina Großkopp, Robin Neuhaus, Marc Hassenzahl, and Majella Mirjam Lilith Vincent. 2022. Interactive Tables for Social Experiences at Home. In *Proceedings of the 16th International Conference on Tangible, Embedded, and Embodied Interaction (TEI '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3490149.3501325