PenSight: Enhanced Interaction with a Pen-Top Camera

Fabrice Matulic[1], **Riku Arakawa**[1,2], **Brian Vogel**[1], **Daniel Vogel**[3]
[1]Preferred Networks Inc., [2]The University of Tokyo, [3]University of Waterloo
{fmatulic, vogel}@preferred.jp, arakawa-riku428@g.ecc.u-tokyo.ac.jp, dvogel@uwaterloo.ca

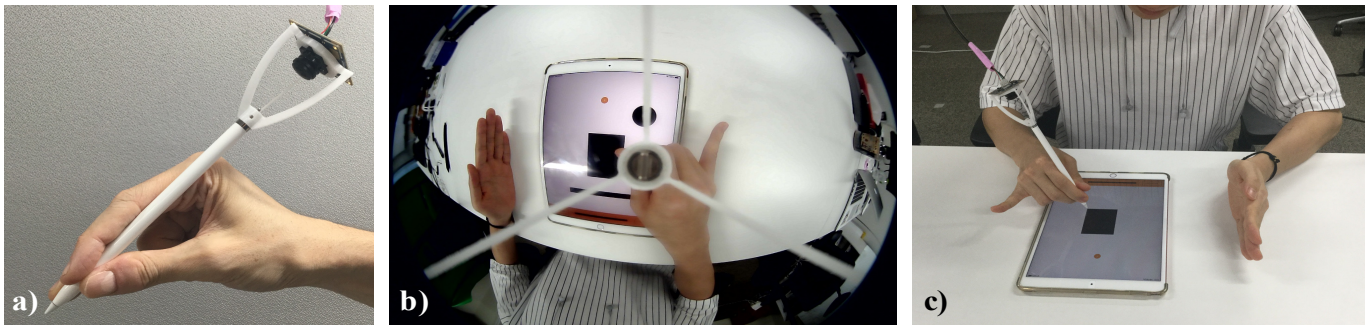

Figure 1. PenSight concept: (a) prototype pen input device with pen-top downward-facing fisheye camera; b) example wide-angle view captured by camera; c) corresponding interaction postures formed by the user with their pen-holding hand and resting hand.

ABSTRACT

We propose mounting a downward-facing camera above the top end of a digital tablet pen. This creates a unique and practical viewing angle for capturing the pen-holding hand and the immediate surroundings which can include the other hand. The fabrication of a prototype device is described and the enabled interaction design space is explored, including dominant and non-dominant hand pose recognition, tablet grip detection, hand gestures, capturing physical content in the environment, and detecting users and pens. A deep learning computer vision pipeline is developed for classification, regression, and keypoint detection to enable these interactions. Example applications demonstrate usage scenarios and a qualitative user evaluation confirms the potential of the approach.

Author Keywords

Pen input; tablet input; hand pose estimation

INTRODUCTION

Increasing the interaction vocabulary of pen input is an important goal. Commercial pens have features to switch modes, like tapping the pen barrel with the second generation Apple Pencil, or methods to enter a temporary "quasimode" [41], such as using the eraser end of the Microsoft Surface Pen. Researchers have proposed many more sensing and interaction techniques in the pursuit of this goal, such as using tilt [50], controlled barrel rolling [6], motion [21], and how fingers grip the pen [46, 48]. Pen manipulations can further be combined with multitouch input on the tablet to create a hybrid "pen+touch" interaction vocabulary, for instance, using coordinated touch input with the other hand [23, 31, 7], or even detecting different hand postures while holding the pen, such as extending the pinkie finger against the multitouch surface [8]. However, the potential pen manipulations, grips, and hand postures are limited by the sensing capabilities of the pen and the multitouch device. Techniques using grip sensors on the pen can only rely on how fingers are pressed against the barrel [46, 48], while the tablet can only detect touch contact patterns on the screen [8].

Using a camera as a sensor makes it possible to capture more diverse hand poses, including in-air postures [47], and hand actions away from the tablet or the pen in the surrounding environment [55]. However, mounting a camera for such situations is challenging. Placing it in the environment, like an overhead camera mounted on the ceiling, may provide a broad view of both hand and surrounding area, but fine-grained tracking is problematic and mobile pen and tablet usage is not supported. Placing a camera on the tablet preserves mobility, but even wide-angle or omnidirectional lenses have a restricted view defined by the plane of the device [58, 55].

To achieve both breadth of view and high mobility, we propose mounting a camera on the pen itself. Unlike pens with cameras in the tip for localisation (e.g. Anoto Livescribe [2]) or pens with cameras facing out from the side of the barrel (so-called "spy camera pens"), we fix a camera with a wide-angle "fisheye" lens *above* the top end of the pen, facing downward. This creates a unique and practical viewing angle to capture both the pen-holding hand and immediate surroundings, including the other hand (Figure 1). We call this approach PenSight.

In this paper, we describe the fabrication of a prototype PenSight device, and use it to explore the interaction design space

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376147

enabled by a pen-top camera viewpoint. This includes dominant and non-dominant hand pose recognition for mode activation and action triggers, tablet grip detection, hand gestures for continuous parameter control, capturing of physical content in the environment, and detecting users and pens. To enable these interactions, we leverage deep learning and computer vision techniques: classification to detect postures, regression to determine relative distances between hand and pen, and keypoint detection to identify a fingertip. Finally, we provide example application demonstrations and usage scenarios, with a qualitative user evaluation confirming the potential of this approach.

Our contributions are: (1) using a novel downward-facing pen-top camera viewpoint; (2) an exploration of the associated design space with applications; (3) the detection pipeline.

RELATED WORK

Our work relates to literature on enhanced pen input, hand pose estimation, and around-device interaction.

Enhanced Pen Input

Multiple methods to enhance direct single-point pen input have been proposed and developed. Touchscreens with separate digitisers or recognisers differentiating pen and touch input have inspired combinations of the two modalities. In the context of bimanual interaction, touch input performed by the non-dominant hand can be used to efficiently assist or complement the dominant hand manipulating the pen [23, 7, 39, 34]. For instance, different modes can be associated with different hand or finger contact patterns of the non-dominant hand on the surface to support rapid switching [31].

Hybrid pen and touch interaction has also been applied to single-hand pen input with the fingers of the pen-holding hand itself forming the mode-determining touch patterns [8].

To track the actions of the pen and the hand manipulating it, sensors can be directly integrated into an "active pen". This enables sensing movements like pen motion and tilt, which has been used to support various orientation-based interactions [6, 50, 21, 24, 19]. Menus can be invoked and controlled above the surface using hovering [16]. Grip sensors can detect how users grasp the pen [46, 48], with the combination of grip and motion sensing enabling context-based interactions and dynamic interface adaptations [22, 61]. FlexAura senses grasp with hand proximity, but the range is limited to 30mm [30]. Beyond grip, the pen can be made deformable to add expressivity through tactile manipulations [15]. Finally, VersaPen is a modular pen with attachable parts integrating different types of sensors to customisable input capabilities [49].

All the above techniques involve some form of grip or close-range sensing on the pen or the tablet. This precludes gestures outside that scope or interaction with the surrounding environment, both of which can be useful in a mobile context. Aslan et al. explored mid-air gestures of the non-dominant hand to assist pen input using a Leap Motion placed next to the tablet [4], but the tracking range is limited by the stationary sensor. With PenSight, we seek to support mode-triggering postures with both the pen-holding hand *and* the other hand, which are not limited by grip and where the postures can be formed rapidly and comfortably from normal hand-writing and hand-resting poses.

Mid-air Hand Interaction

Techniques have been proposed to create "around-device interaction" by detecting mid-air gestures using cameras built in, or attached to, a mobile device [47, 12, 58]. Using an omnidirectional lens has also been shown to extend the view to include the surrounding environment [55]. While effective for broad contextual sensing, a camera mounted on the mobile device itself still has a limited view angle and cannot track hands moving across devices and media.

Cameras or sensors worn directly on the hand or the arm allow more ubiquitous sensing, but at the cost of instrumenting the user. Wearable sensors used for hand pose estimation include mini cameras [10], infrared [26, 35], accelerometers [52, 54], pressure [13], electrical impedance tomography [60], ultrasound [25, 36], and electromyography (EMG) [43, 59]. These methods seem suited to detecting pen-holding poses, but they have mostly been used to recognise penned content such as handwriting [29, 56, 45]. We attempted to recognise pen-grip postures using the Myo commercial EMG armband, however we only achieved moderate success due to the fidelity of the sensor [33]. But even high-precision wearable sensors are limited to signals directly emanating from the body. PenSight, with its pen-top camera, does not require user instrumentation, and it can recognise both hands as well as the surrounding context wherever the pen is taken. This means it is not limited to a particular setting or device, nor is it limited to a single pen as the mount can be detached and reattached to other pens or pen-like instruments.

HARDWARE CONCEPT AND PROTOTYPE DESIGN

We describe our prototypes created to realise the PenSight concept, followed by depictions of potential future designs.

Proof-of-Concept Prototypes

A key requirement for an active pen is that it should still feel comfortable to grip and manipulate. It should not be excessively large, have weight imbalances or be tethered for power or data transfer. Given those considerations, we initially looked for a miniature wireless wide-angle camera that could be fixed at a moderate height above the top end of the pen via a 3D-printed mounting piece. We created a first prototype using an Aobo wireless camera combined with a 165° clip-on lens for mobile phones to increase the field of view. Although this was untethered, the setup significantly increased the pen length (8.5cm) and weight (52g), and streaming over WiFi showed a noticeable lag that could be detrimental to user experience. We therefore opted for a more lightweight design using a single mini USB camera module with a 180° fisheye lens. Without considering the cable, this reduced the added pen length to 6.2cm and weight to 18g. The camera streams video at 30 fps in 1920×1080 resolution over USB2.

The 3D-printed mount has a lower ring that fastens to the end of the pen and a top platform holding the camera in a downward direction (Figure 1). The camera platform must

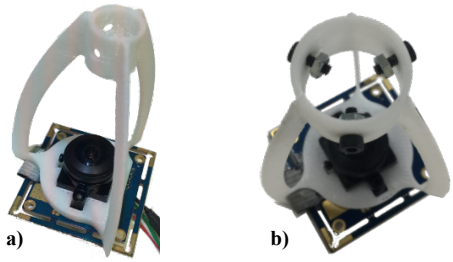

Figure 2. 3D-printed mounts: (a) with a ring sized for an Apple pencil; (b) using lateral screws in the ring to adapt to different pen diameters.

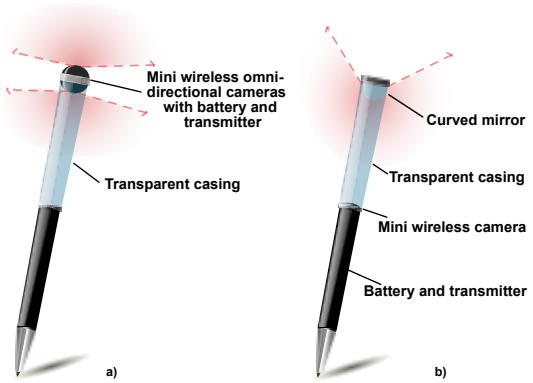

Figure 3. Future PenSight concept designs: (a) omnidirectional camera in the pen top; (b) catadioptric configuration with a camera in the barrel and curved mirror in the pen top.

be "raised" above the pen-top using a structure that is robust and stable, while minimising occlusion of the camera view. Through trial-and-error we arrived at a compromise with three equally spaced 1mm thin blades (Figure 2). The pen-fastening ring can be printed in different diameters. We printed versions for a Wacom stylus, an Apple Pencil, and a larger ring with lateral screws to adapt to other pen barrels.

Future Concept Designs

The 3D-printed mount adds length and weight to pens, which raises the centre of gravity. To remedy these issues in the future, we propose some concept ideas of possible PenSight designs with pens and cameras manufactured for this purpose. For example, a design with a mini wireless omnidirectional camera consisting of two camera units, each with a 180° field-of-view, one facing downwards towards the interactive surface, and one facing upwards to capture the environment above the pen including the user's face (Figure 3a). This would deliver fully circular image frames with equal scene coverage similar to commercial 360° cameras. Instead of an opaque mount, the upper part of the pen barrel could be made transparent to eliminate peripheral occlusions caused by support structures.

An alternative design for pens could be based on a catadioptric configuration with a camera located inside the lower part of the pen barrel (also containing the battery and the transmitter) and facing upwards towards a curved mirror fixed to the pen top (Figure 3b). Curved mirrors for catadioptric systems can support very wide viewing angles so that coverage similar to fisheye lens and omnidirectional cameras can be achieved [38]. In both future concept configurations, there is a tradeoff between achieving the necessary height of the captured view for sufficient coverage of the hands and the environment with minimal occlusions versus a practical pen length and weight.

INTERACTION TECHNIQUES

Since our focus is on exploration rather than exhaustiveness, we examine a selection of example interactions from different situations of pen and tablet use. These are representative of a broader support for many related grips, postures, and positions adopted by users in each case. The interactions that we consider can be divided into three categories: hand-posture-based interactions, interaction with the environment and identification. We first give a description of these interactions and later propose potential associations with actual modes and interface actions in example applications.

Hand Posture-Based Interactions

Prior work showed that hand postures can be used to efficiently switch pen modes or trigger interface actions upon detection [23, 31, 48, 46, 22]. These shortcut-like actions avoid time-consuming round-trips of the pen between the main workspace and mode-selection widgets commonly placed on the edges of the interface. Trigger postures can be formed by the non-writing hand (non-dominant hand) or the pen-holding hand (dominant hand). A popular theoretical framework grounding the design of interactions in the former case is the kinematic chain or asymmetric division of labour [17], where the non-dominant hand sets the frame of reference in which the dominant hand operates. For bimanual pen interaction, this can translate to non-dominant hand postures defining so-called quasimodes, defined as modes that remain active as long as the corresponding posture is formed [41]. For example, the pen is in selection mode while four fingers of the other hand are touching the screen [31]. For unimanual interaction, where mode-switching postures are formed by the pen-holding hand using different pen grips, quasimode triggers are also possible. However, for some postures and grips, the dexterity, switching speed, and fatigue may make this less appealing [8]. When using variations of pen grips for temporary mode changes, it is better to associate them with instant actions [33].

We support postures made *both* by the pen-holding hand *and* the other hand. To avoid the above issues, we assign postures of the other hand to quasimodes and pen-gripping postures to single actions. This clear separation of the hand roles should also reduce confusion for the user. In addition to interactions associated with either the pen hand or the other hand, we also propose a novel category of techniques combining both hands to form single postures.

Normal Hand-Resting Poses

When using a tablet with a pen on a desk, the other hand is mostly inactive, except when used for occasional multitouch actions like pinch-to-zoom. When not active, the hand is often resting on the table or on the user's lap [51]. We are not aware of any study investigating typical poses of the passive non-dominant hand and arm during pen tasks on tablets. Our informal observations suggest many people place their hand near the device, either on the opposite side or below, with the palm down, flat on the surface, or with lightly curled fingers

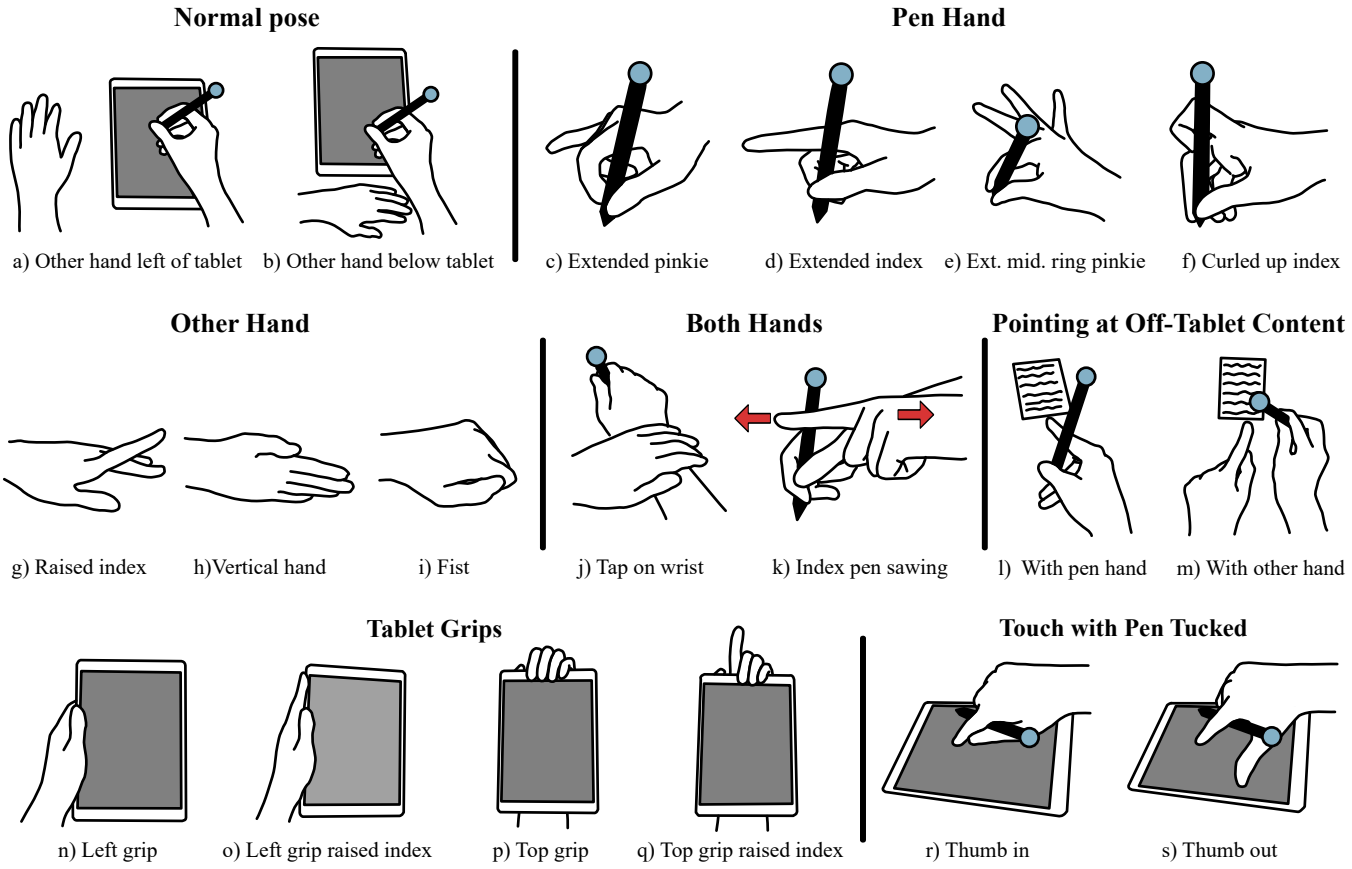

Figure 4. Supported hand postures, interactions, and tablet grips.

(Figure 4a,b). We build on these normal resting poses of the other hand to create our mode-setting postures.

Pen Hand Distinct Actions

Considering the position of the camera immediately above the pen hand, interactions based on gripping postures formed by this hand were a natural first choice in our exploration. Our previous study of alternative pen grips for mode-switching offer a number of possible candidates [33]. However, the viewing angle of the camera limits postures to those that can be distinguished from the top. Together with the requirement that it should be possible to form these postures rapidly for instant action triggers while maintaining pen grip stability, this leaves us mostly with poses consisting of extending one or more fingers. For our pen grips, we therefore consider extended index finger and extended pinkie, to which we add two new postures: extended middle, ring and pinkie and index curled up (Figure 4c-f).

Touch with Pen Tucked

Interaction with the dominant hand is not limited to pen input if the tablet also supports touch. For instance, pen interaction is often interleaved with touch input for panning and zooming. To perform touch operations with the pen-holding hand, the pen can be temporarily tucked between the fingers. There are several pen-stowing strategies, which depend on user preferences and context [22]. PenSight can be used to detect such postures. We show an example using two variations of one popular grip: tucking the pen with the index finger, or with adducted or abducted thumb (Figure 4r,s). The position of the thumb can be used to control an additional mode when performing touch operations with the tucked pen, such as locking 2D panning to one direction.

Other Hand Quasimode Postures

The wide viewing angle of the fisheye camera can extend posture-based interaction to the other hand when located near the device. As described above, if the tablet is placed on a table, the non-dominant hand has a more or less passive role depending on how frequently it is involved in touch operations. In our case, the non-dominant hand actively participates in the interactions by triggering quasimodes using particular postures. Since quasimodes are by definition maintained, we seek non fatiguing postures, where the hand still rests on the surface, but can be differentiated from casual resting poses for robust detection. Ideally, transitions from normal resting (passive) poses and mode-setting (active) postures should be smooth and effortless to support rapid switching [34]. Following these considerations, we selected three mode-setting postures: a fist and a vertical hand, which have been used previously for touch input on tabletops [53, 32] and mid-air interaction [5], and a resting hand with a raised index finger, which we believe to be new (Figure 4g-i). There are of course other possible postures that also fulfil the desired requirements.

Tablet Grips

Another important role of the non-dominant hand in tablet interaction is holding the device in mobile situations. Previous work showed that grip patterns provide useful contextual information about tablet use for situated interactions [57, 22, 61]. If the hand is holding the tablet from one of the sides, then the pen-top camera can capture it and recognise the gripping posture. Furthermore, unlike tablet prototypes with grip sensors that are limited to recognising touch patterns, PenSight can detect mid-air variations of grip postures, such as raising a finger, to support additional actions triggered by the tablet-gripping hand. We propose to use the index finger for such actions as it is arguably the most capable of independent movement in these scenarios. We use index-based triggers as extensions of two tablet gripping poses: left and top grips (Figure 4n-q).

Both Hands

Both hands can also be combined to form single bimanual postures. These are better suited for instant actions, since maintained poses would require moving the other hand along with the pen hand. As an example, we propose a gentle tap of the other hand on the wrist of the pen hand (Figure 4j). This action requires little effort if the other hand is resting below the tablet (Figure 4b).

Interactions are not limited to triggering single modes or discrete actions, but can also be used to control continuous parameters. One idea, proposed by Aslan et al. [3], is to use the distance between the other hand and a sensor close to the side of the tablet. Reliably inferring distance on the depth axis using only an RGB camera is challenging considering the different sizes and appearances of hands. Distances can be more robustly measured from a top view, and in our context this means that a hand moving to modify a continuous parameter would do so just under the camera. Therefore, we propose a gestural action in which the extended index finger of the other hand orthogonally contacts the pen and the distance between the fingertip and the pen (i.e. the centre of the image) determines the value of the continuous parameter to control. To increase or decrease a value, the user rubs their finger against the pen towards or away from them in a "sawing" motion (Figures 4k and 6c).

Interaction with the Environment

Apart from the hands, PenSight can also "see" the environment, including objects around the tablet, such as documents, which can be involved in the task. The camera can further be used to identify the user and the pen it is attached on.

Capturing Off-Tablet Content

Perhaps the most straightforward way to capture content in the environment would be to pick up the tablet and use its camera. If only a portion of the image is required, it can be cropped out using any standard photo editing application. However, this operation can be tedious for repeated acquisitions and when the captured content is used as a basis for further operations, like image search or OCR. A pen with a camera seems like a convenient tool to capture and reuse, or "pick-and-drop" [42] selected elements in the vicinity. For example, the user might want to circle items in a magazine to copy or search for, or underline a word in a book, to look up the definition. Unfortunately, the camera is unable to see the pen tip, as it is occluded by the holding hand. The pen, therefore, cannot be directly used to point at and select artifacts.

As an alternative method, we consider pointing with the index finger of the pen-holding hand or the other hand. Pointing with the pen hand requires extending the index while tilting the pen forward and above the content so that the camera can capture it (Figure 4l). Since we use a camera with a fisheye lens, the desired content can still be fully framed, even at a low tilt angle. This creates a pose similar to a touch with pen-tucked posture. If pointing with the index finger of the other hand, the pen hand simply needs to be held above or close to the content to capture (Figure 4m). The advantage is that each hand has a separate role: one points, the other frames, but at the same time, both hands are monopolised for the interaction. As for the action to execute the image capture, a trigger posture, such as a tap on the wrist, can be used in the first case and an extended finger in the second.

Pointing with the index finger only provides a single x-y point in the image and it does not explicitly define a complete region of interest for the desired content. There are many ways to extend single-point selections to lines and even bounding boxes. Among possible interactions that we considered are underlining with the finger, using three fingers together, or detecting three successive taps using the same finger to define the three corners of a rectangular region. All these options are feasible, but they either require more complex positioning of the hands and fingers or time-based detection. In our initial proof-of-concept implementation, we use a single-finger position captured in a single image frame to select content with well-defined boundaries, such as a photograph or isolated text.

Detecting Users and Pens

The camera can identify its own context as well. For example, anticipating that people have different preferences with regard to postures and their mappings to modes and shortcuts [33], a shared PenSight pen can detect which person is currently using it. This way, personalised preferences and custom settings can be loaded accordingly. Another application of user identification could be to prevent a pen from being used by unauthorised people.

One possible and immediate biometric method to identify users with a pen-top camera is to recognise their hand. Previous work has used the geometry and features of the back of the hand to identify users in collaborative tabletop scenarios [40, 44]. Since the pen-holding hand is very close to the camera, we hypothesise users can be easily and reliably identified using machine learning, given enough training data. Alternatively, if training on hand data is impossible or too cumbersome, the user's face can be used as the biometric identifier instead. For that, the pen camera can simply be pointed at the user's face. Since facial recognition is a more established identification method, existing databases can be used to recognise users, without the need for ad hoc training with hand image data.

Finally, another customisation that PenSight can support is detecting the pen on which the camera is attached. Many pens can be distinguished from their top end only, so analysing and

Figure 5. Tool identification using the central area of the camera image.

comparing the central region of the camera image in which the pen end appears can enable coarse identification (Figure 5). Pens that look too similar can be differentiated or personalised by adding recognisable markers.

While in this work, we only look at digital pens, the camera mount can also conceivably be fixed to regular pens and even to other pen-like instruments as well. Again, different settings can be loaded depending on the capabilities or security properties of the connected instrument. For instance, attached to a light stylus, only quick pen hand postures might be activated, whereas if coupled with a brush, different holding grips used by artists might become available [1].

DETECTION PIPELINE

Hand pose estimation involves recognising the location of hands, fingers and their joints in RGB or depth images using computer vision [28]. In recent years, techniques based on deep learning have enabled robust real-time hand keypoint detection from individual RGB frames, with frameworks like OpenPose [9] including code, datasets and pre-trained neural network models made publicly available [11]. Our camera lens and capturing angle differ markedly from those used to train public models, however, and our tests with these frameworks did not give satisfactory results, especially for the pen hand. We therefore develop our own machine learning pipeline for our specific needs.

Data Gathering

To be able to robustly detect postures from different users, a large amount of training data is required. This data should ideally come from a diversity of people with different hand sizes, skin tones, clothes, and be captured in various environments with different backgrounds and lighting conditions. The goal of this work is not to produce a general dataset, but merely to prove the feasibility of the PenSight concept and its interactions. To keep data gathering to a reasonable level for this purpose, we limit ourselves to collecting data in our lab using one tablet and with 15 people.

Data Capture Environment

We record our posture data with the tablet placed on a desk and when gripped by participants. We do so in two different settings: one with a white desk and the other with a table and enclosure covered by a green sheet. This green backdrop can then be artificially removed using chroma keying and replaced with random images so that the neural network learns to ignore the background. We use the white desk for 12 participants and the green environment for 5, with therefore two participants using both setups. We use an iPad Pro for the tablet (size 10.5", weight 469g) and an Apple Pencil for the pen (length 175.7 mm, diameter 8.9 mm, weight 20.7g).

To record data, participants are asked to move the pen on the tablet while forming postures successively. When performing postures with the other hand (fist, chop, vertical hand etc.), the pen is held normally. In addition to creating data for those other hand postures, this also increases the amount of data for normal poses of the pen-holding hand. We use that data later to train our models for user identification. Normal postures further need to include transitory phases between resting poses on the table and mode-activation postures requiring more hand movement, such as index pen sawing. Since that posture should only become active when the index finger touches the pen barrel, we need to record data up to that point: from the hand resting next to the tablet to the moment just before the finger contacts the pen. It is crucial for a neural network to be exposed to many of these negative examples in order to mitigate misrecognitions.

No specific tasks are given to participants when recording data. Participants form and hold postures while drawing randomly on the tablet. They are requested to move their hands as much as possible in order to cover different positions, orientations, and tilting angles for the pen. Unfortunately, the camera does not capture frames showing the full circle covered by the fish-eye lens as the top and bottom parts are partially cropped out (see Figure 1b). This, and the three visible support blades, result in the camera view being angle-dependent, which makes it difficult to programmatically rotate images for data augmentation. We therefore ask participants to physically rotate the camera to cover different rotation angles.

Data

After capturing the data, to make sure we obtain clean datasets for training, we manually inspect and remove all images where postures are not visible or correctly formed. After cleaning, there were 338,891 images with amounts ranging from 15,000 to 23,600 for each of the 17 posture classes and roughly 45,000 images for normal handwriting poses. About 20% of the data was acquired with the green backdrop and thus subject to artificial background replacement. Images for index pen sawing and pointing actions are manually labelled with the x-y coordinates of the fingertip. For index pen sawing, we further compute the distance between the fingertip and the pen (the centre of the image) to be used for regression.

We create seven datasets for different hands and their interactions: five for posture classification, *Pen Hand*, *Other Hand*, *Tablet Grip*, *Left Grip Mode*, *Top Grip Mode*; one for regression, *Index-pen Distance*; and one for keypoint detection, *Fingertip Location*. Postures included in each set are listed in Table 1. We arbitrarily include the tap on wrist and index pen sawing postures, which involve both hands, in the *Pen Hand* and *Other Hand* sets respectively. Note that images for index pen sawing are used for both classification and regression, as it is first required to know when that mode is active (classification in *Other Hand* set), before the continuous value can be inferred from the fingertip coordinates (regression in *Index-pen Distance*).

Neural network models are trained for each of the above datasets resulting in multiple recognisers that can be activated and deactivated in applications when needed. We describe

when the recognisers are enabled and how they are used in our demonstration applications further below.

We create an eighth dataset, *User ID Hand*, for hand-based user identification consisting of all other hand postures, which were recorded when the dominant hand was holding the pen normally. This makes it easier to conduct fairer and easier accuracy assessments by splitting training and validation sets according to postures sets, but of the same type (e.g. "normal" postures used for validation, the rest for training).

Neural Network Models

We use neural network models based on a ResNet-50 architecture [20], a type of convolutional neural network (CNN) that has proven successful for classification and regression tasks on image data. The base layers of the models are pre-trained on ImageNet [14]. To these base layers we add a single fully-connected layer after the last pooling layer, whose dimensions correspond to the number of desired output classes or values. This is equal to the number of postures included in the dataset, except for *Tablet Grip* where we merge the left and top postures together (i.e. grip with and without raised index), since the purpose of that set is to determine where the user is holding the tablet, irrespective of the index position.

For the classification models, we use softmax and cross entropy as the loss function and match rate as the accuracy metric. For the regression models, mean squared error is used for the loss and coefficient of determination for accuracy.

We use the Adam [27] optimiser to train all models. The learning rate coefficients for the base layers and the last layer are set to 5×10^{-5} and 1×10^{-3} respectively. Posture classification models are trained for 10 epochs and regression models for 50 epochs.

We perform 5-fold cross-validation to evaluate each trained model. For the validation of the seven posture dataset models, we leave out three participants chosen pseudo-randomly such that one belongs to the green background condition, and two to the regular desk group. Two pilot testers, who provided a large amount of data for both conditions are always included in the training set. For the *User ID Hand* model, we validate on one of the four included posture sets, and train on the remaining three.

Results

Results are shown in Table 1. Rates are all above 79% with the main dataset models *Pen Hand* and *Other Hand* showing accuracies above 90%. This is theoretically sufficient for prototype testing, although these results do not reflect badly formed and occluded postures. We provide estimates of the impact of these factors later when evaluating the postures within applications.

The results for user identification using hand images is close to 100%, which is encouraging, but is admittedly based on a large amount of data and few users. In a real deployment, robust performance would need to be achieved with only a few training samples.

Dataset	Included Postures	Accuracy
Pen Hand	normal, ext. pinkie, ext. index, ext. mid. ring pinkie, curled up index, tap on wrist, pen tucked thumb in, pen tucked thumb out	93.6%
Other Hand	normal, raised index, vertical hand, fist, index pen sawing	90.8%
Tablet Grip	top grip, top grip raised index, left grip, left grip raised index	96.1%
Left Grip Mode	left grip, left grip raised index	79.6%
Top Grip Mode	top grip, top grip raised index	87.2%
Index-pen Distance (regression)	index pen sawing	0.90 (R^2)
Fingertip Location (keypoint)	pointing with pen hand	0.84 (R^2)
User ID Hand	normal, raised index, vertical hand, fist	99.5%

Table 1. Posture dataset and model recognition accuracy (error rate for classification, coefficient of determination (R^2) for regression and keypoint detection).

For our demonstration applications, we train models for each posture dataset using all data (combining training and testing partitions used above), since with deep learning, more data generally translates to more robust models.

DEMONSTRATION APPLICATIONS

We create two tablet applications to demonstrate the interactions enabled by the PenSight concept, a sketching application and a map application. We use these applications to evaluate the usability of our techniques when sitting with the tablet placed on a desk and sitting while holding the tablet with the other hand. Posture-to-action assignments for each application are made with one of these two settings in mind: the sketching application for the desk condition and the map application when holding the tablet. This allocation is somewhat arbitrary and mainly serves the purpose of providing two different testing contexts to experience PenSight interactions.

Since current tablets are not sufficiently powerful for a posture recognition pipeline based on multiple ResNet-50 networks, we opt for a client-server architecture. The server, a Windows PC with a Geforce GTX 1080 GPU, runs the various posture recognisers in Python. The client is an iOS Swift application running on an iPad Pro. The server and the client are connected via WebSockets. When running the required recognisers for each application in parallel, the frame rate averages 12fps, which is sufficient for testing purposes.

Sketching Application

The first application is a sketching app. The default mode is inking, when both hands are in their normal pen-holding and resting poses. In this mode the pen inputs freeform strokes on the canvas. For other modes and actions, we utilise our recognisers with the *Pen Hand*, *Other Hand*, *Index-pen Distance*, and *Fingertip Location* models described above. Active modes are displayed in a status bar at the bottom of the screen, so that the user can confirm that postures have been correctly detected.

Pen Hand Model for Instant Actions

The pen-holding hand is used mainly to trigger instant actions as it is fatiguing to maintain non-conventional pen gripping

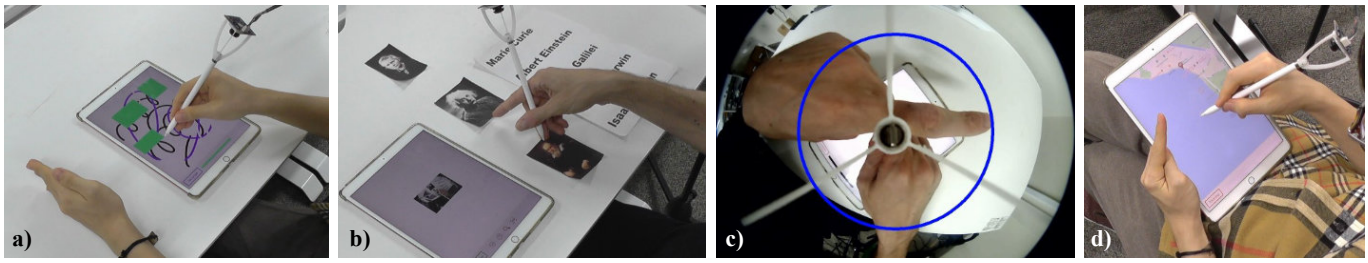

Figure 6. Example applications and interactions: a) sketching application with other-hand posture setting a mode; b) Pointing at off-tablet content to perform searches; c) pen "sawing" action with fingertip distance (blue circle) mapped to a continuous parameter; d) map application with raised finger of the tablet-gripping hand setting a mode. See the accompanying video for full demonstrations.

postures for quasimodes [33]. Among the basic functionality of our application, we support undo and redo. Those are frequent and symmetric operations so we assign them to extended index finger and pinkie respectively. The user can thus perform multiple undo/redo operations by repeatedly lifting the required finger. A menu with a colour palette and other actions can be invoked in place (it appears where the pen last touched the screen) with *curled up index*. This is a very frequent operation, and *curled up index* seems most suitable since it requires comparatively little finger movement. However, pilot tests revealed that it was not always easy to perform or detect that posture, so we add a backup possibility to call the menu with *extended middle ring and pinkie*. This is perhaps more costly in terms of finger movement, but seems to be more reliable for some users. The application also supports a clear-all function, which is less frequent but consequential if mistakenly triggered, so we map it to the *tap on wrist* action, which requires more movement from the other hand.

The canvas can be scrolled using touch with the pen tucked. When the thumb is kept in, scrolling in any direction is possible. Extending the thumb with the *pen tucked thumb out* posture locks scrolling to vertical or horizontal axis-aligned directions. These postures need to be maintained to keep the quasimode active, but because the pen is tucked, they are likely less taxing compared to a finger kept extended while writing. This design might not be suitable for applications with zooming or scaling capabilities, as the thumb is often used in combination with the index finger for the common pinch-spread gesture. We use these postures to test the feasibility of thumb adduction and abduction for mode-switching while the pen is tucked.

Other Hand Model for Quasimodes

The other hand is mainly used to set quasimodes, similar to pen and touch applications [23, 31, 7]. Erasing mode can be activated with *raised index*, a low-cost posture for a frequent operation. When that mode is active, the pen deletes instead of inking. Further quasimode operations include the insertion of rectangles and circles. We map those to *vertical hand* and *fist* respectively, given their shapes roughly match and therefore mental association is likely facilitated (Figure 6a).

Finally, we map *index pen sawing*, our posture for continuous parameter control, to stroke width. While arguably also costly in terms of hand movement, we believe it is intuitive and easy to perform. To change the width of the stroke the user first brings the index finger of their other hand against the pen (detected by the *other Hand* recogniser), then moves their finger forward to increase the distance between the fingertip and the pen, or pulls it towards them to decrease it. The length is determined by the *index-pen distance* model and mapped to a stroke width value. The status bar shows a preview of a stroke with the currently selected thickness (Figure 6c).

Fingertip Location to Point at Off-Tablet Content

The last model used in the sketching application is *Fingertip Location* to detect where the index finger of the pen-holding hand is pointing. We use this to support the capture of physical off-tablet content such as documents and photographs that may be lying on the desk. Among the functionality that the application supports is pick-and-drop [42] by copy-pasting the image of the content, and image and text search from selected content (Figure 6b). For the latter feature, it is necessary to determine what region to crop from the captured image. In this proof-of-concept application, we use artifacts with clear boundaries such as photographs and single lines of large text. Rudimentary computer vision techniques based on Canny edge detection and contour-finding can then be used to extract the element that is the closest to the pointing fingertip. This approach does not work for text, so we use the EAST detector [62] to obtain the bounding box of detected text elements in images. These text boxes can then be converted to machine-readable text using the Google OCR API. The captured image or extract text is used in a Google web search with results shown in a browser.

To use the capture tool, the user first activates it in the menu and then points the index finger of their pen hand underneath the artifact they wish to capture. With our current hardware prototype, it is necessary to carefully position the camera so that the element to capture is fully visible in the image. In particular, it should not be partially occluded by a support blade of the mount, otherwise content extraction fails. A preview screen on the tablet shows the camera view to facilitate adjustments.

Map Application

Our second application features interactions with a map when holding the tablet with the other hand. Casual navigation of digital maps on tablets is usually performed with touch input, but a pen can be useful for precise tapping, tracing, and annotating. In our map application, the default pen mode is panning, where dragging the pen causes the map to scroll in any direction. Other modes and actions are enabled by recognisers based on *Tablet Grip*, *Left Grip Mode*, *Top Grip Mode*, and a subset of the postures of the *Pen Hand* model.

Mode-Switching with Tablet-Gripping Hand

Since panning is the default mode, zooming support is required. We assign the *raised index posture* of the tablet-gripping hand to the zooming mode (Figure 6d). When the index finger of the other hand is raised, the pen can be dragged up and down the screen to zoom in and out. Since extending the index finger is quick, users can rapidly change between panning and zooming. Any posture recognition errors also have limited impact.

To demonstrate a typical pen task, the application also supports annotations. In this scenario, the default mode is inking, with the *raised index posture* mode used for erasing. Panning and zooming can then be performed using traditional touch gestures with the pen tucked. For technical reasons, we implemented inking as a switched mode and left panning as the default. While this design is arguably less logical, the main purpose is to informally test pen interaction in regular and switched modes.

Pen Hand Model for Triggering Actions

We reuse some of the postures of the *Pen Hand* set to support typical map operations such as changing the terrain type and performing route searches. Switching between terrain types is achieved by *extended index*. To search for routes, the user first taps two or more locations on the map. Forming the *extended middle ring pinkie* posture then triggers the route calculation between the locations in the order they were selected.

QUALITATIVE EVALUATION

We conduct a qualitative evaluation to assess the usability of PenSight techniques within our two applications.

Participants

We recruited 16 participants (12 male, 4 female) of mean age 30 years old (SD 6.1). Seven of these participants also completed the data gathering sessions earlier. Six participants reported using a tablet on a weekly or daily basis, four once or twice every couple of months, and the rest rarely or never. One participant was left-handed and since our models were trained for right-handed people, we flipped the frames streamed from the camera to support that user.

Natural Pen Postures

Before beginning the main part of the session, we asked the participant to show how they naturally hold a pen, and where they place their other hand while writing or sketching. We also asked them to demonstrate how they would tuck the pen for temporary touch input with the pen hand, and how they usually hold a tablet.

We found 11 participants used the dynamic tripod, three lateral tripod, and two dynamic quadrupod as their natural pen grip (see [33] for a description of these grip types). Regarding how participants naturally placed their other hand, two participants put it on their cheek or lap, and all others placed it on the table: either flat (9 people), with lightly curled fingers (4 people), or forming a fist (1 person). Pen tucking strategies varied between pinching with the index (8 participants, our supported posture), tucking with the ring finger (1 person), and palm grips (3 people). Two people used their normal writing pose and grazed the tablet with the pinkie or the ring finger to perform touch operations. Two used a combination of different tucking strategies. As for the normal tablet grip, the left grip was the most common (11 people), then top left or right corner (4 people), and bottom (1).

We instructed participants to use the "normal postures" assumed by the models during the rest of the study. For example, to avoid conflicts with *Other Hand* postures (especially *fist*), we instructed participants to adopt the flat hand resting pose. We recorded comments regarding potential discomforts when forming these unfamiliar poses.

Main Task and Protocol

We performed the study in the same conditions in which the data was captured, with people either sitting at a desk with the tablet placed on it or holding the device while sitting back. In each session, we demonstrated each group of techniques, starting with the postures for the sketching application. After showing each set of interactions, the participant was given the opportunity to freely use the applications and PenSight techniques in a self-directed way. They were asked to actively provide feedback at any time during and after the experiments.

A session lasted approximately 50 minutes. Participants were given a choice of snacks as a thank you.

Results

Overall, participants enjoyed the *Other Hand* postures the most as they proved the easiest to perform. They particularly liked the postures for the geometric shapes (*vertical Hand* and *fist*) as the mapping was intuitive. The pen tucking postures were also generally comfortable, even though 4 participants who naturally extend the thumb to stabilise the pen felt they lost some of that stability when moving the thumb in.

The most disliked posture was raising the index finger with the other hand while holding the tablet. All but two participants said it reduced gripping stability. Using the raised index with the top grip was better, but people who usually hold the tablet from the side are not used to that grip and so were not entirely comfortable with it.

From the *Pen Hand* set, the best posture was *extended index* with 11 participants finding it easy to execute. Preferences for curled index were split, with four people stating that it was the most difficult pen hand posture to perform (4 people), but five people finding it efficient. The *extended mid ring pinkie* was preferred by 7 people to invoke the menu. Opinions were equally divided for extending the pinkie, with 4 people finding the posture comfortable and 4 others finding it difficult, especially when the pen is close to the screen and the pinkie cannot unfold without hitting it. Both *tap on wrist* and *index pen sawing* were deemed natural and practical (5 people for both), despite requiring more hand and arm movement. For the pen sawing action, 4 participants said it was difficult to aim for a precise value. There is indeed some limitations as to the precision that can be achieved with this technique.

One of the other main concerns was that the pen felt heavy (8 participants) and that it may have affected how easily they

can perform some of the *Pen Hand* postures. Using one of the designs of Figure 3 would hopefully remedy this issue.

Pen Hand versus Other Hand

Participants were asked if given the choice to execute instant actions with postures from one of the hands only, which hand they would prefer. 9 favoured the other hand, since it is unconstrained when forming postures, while 3 people would select postures formed by the pen-holding hand. They stated they are used to their other hand not participating in any input actions when operating a touchscreen. This dichotomy is also reflected in the preferred interactions to point at external content. 9 people declared preferring holding the pen with one hand while pointing with the other, whereas 2 people felt the dominant hand should be the only one actively engaged in manipulation tasks. When the other hand is gripping the tablet, however, participants all agreed that *Pen Hand* postures are more convenient.

Reverting to Natural Resting Pose

Participants who do not adopt the flat resting hand on the surface as their normal resting pose (and were forcibly required to do so) tended to revert to their natural pose after a while or when returning from postures like *tap on wrist* or *index pen saw*. While these behaviours might gradually disappear through habit, postures could also be adapted to user constraints and preferences.

Detection Accuracy

From our observations, *extended index*, *fist*, and *vertical hand* were the most robustly detected postures. Conversely, *raised index* when gripping the tablet, *extended pinkie*, and *curled up index* suffered from recognition issues. For a large part, this was due to occlusions or because differences between other postures were too small. For some postures, misrecognitions can be mitigated by allowing slight variations. For instance, raising both the index and the middle finger was still detected as a raised index. Furthermore, these modified postures are also sometimes more comfortable to execute, so this is an added benefit.

Discussion

With some caveats, posture-based interaction can be a powerful addition to classic user interfaces. PenSight techniques seem strongest for *Other Hand* postures, which, contrary to touch or grip-based actions, can be performed around the device. *Pen Hand* postures are also practical for users who do not want to interact using both hands. The *extended index* is the most comfortable of our tested postures. It can be assigned to a common shortcut such as menu invocation or undo, depending on the application needs. Other finger extensions are also possible, but reduced visibility becomes an issue for the pinkie. Interaction techniques requiring both hands demand more movement but they are perceived as intuitive and so should not be discarded. Postures for continuous parameter control can be used to a limited extent using only an RGB camera. As for grip-based postures, while detecting where the user holds the tablet is feasible (it does not even need to occur at each frame, since people do not frequently switch grip locations), extending a finger of the gripping hand to trigger a mode is not recommended. This means that in such contexts, posture interaction is mostly limited to the pen-holding hand.

As for interactions with the environment, possibilities are also constrained by what the camera can see, considering the occlusion of the pen hand. Pointing at content in documents can be achieved with fingers instead of the pen tip, preferably by dividing the roles between the two hands: The pen hand positions the camera while the other hand selects the content to be captured. Further interactions can be considered to precisely indicate which portion of an artifact should be used for capture. For text, this could be running the finger on or under the term or phrase. For images, tapping the corners of the region could extract the desired area.

As for detection accuracy, hand pose estimation has made tremendous progress thanks to advanced deep learning techniques. It is now possible to infer hand and finger keypoints using a single RGB frame. To achieve this level of accuracy, however, considerable amounts of data are required and it is not clear how much of the existing public datasets can be successfully leveraged for the rather uncommon viewpoint of a pen-top fisheye camera. But data acquisition and labelling techniques are also improving, with methods that can automatically train or pre-train neural networks using synthetically generated data [18, 37]. As camera and sensor technology improves, and machine learning techniques evolve, more precise and accurate, yet cost-effective solutions, will likely emerge.

CONCLUSION

We presented PenSight, a concept to enhance pen interaction for tablets by attaching a camera to the top of the pen. We built prototypes using a fisheye camera and 3D-printed mounts to explore several examples of techniques enabled by this paradigm. These include posture-based interaction using both hands, individually or in tandem, interacting with physical documents in the surrounding environment and identifying users and the attached pen. We examined postures and off-tablet interaction in more depth with two demonstration applications. The results of our qualitative evaluation demonstrate potential for postures formed by the other hand when not holding the tablet and for some poses of the pen-holding hand. There are doubtless many other possible techniques that can be realised. Overall, we believe PenSight provides a simple holistic mobile sensing solution for capturing and interpreting interactions of the two hands as well as the surrounding environment.

REFERENCES

[1] 6 Key Ways To Hold A Watercolor Brush. `https://watercolorpainting.com/brush-exercise/`. Accessed: 2019-09-17.

[2] Anoto Livescribe. `https://www.anoto.com/solutions/livescribe/`. Accessed: 2019-09-01.

[3] Ilhan Aslan, Björn Bittner, Florian Müller, and Elisabeth André. 2018. Exploring the User Experience of Proxemic Hand and Pen Input Above and Aside a Drawing Screen. In *Proceedings of the 17th*

International Conference on Mobile and Ubiquitous Multimedia. ACM, 183–192.

[4] Ilhan Aslan, Ida Buchwald, Philipp Koytek, and Elisabeth André. 2016. Pen + Mid-Air: An Exploration of Mid-Air Gestures to Complement Pen Input on Tablets. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, 1:1—-1:10. DOI: http://dx.doi.org/10.1145/2971485.2971511

[5] Ilhan Aslan, Tabea Schmidt, Jens Woehrle, Lukas Vogel, and Elisabeth André. 2018. Pen + Mid-Air Gestures: Eliciting Contextual Gestures. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction (ICMI '18)*. ACM, New York, NY, USA, 135–144. DOI: http://dx.doi.org/10.1145/3242969.3242979

[6] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. 2008. An Exploration of Pen Rolling for Pen-based Interaction. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 191–200. DOI: http://dx.doi.org/10.1145/1449715.1449745

[7] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. Combining and Measuring the Benefits of Bimanual Pen and Direct-touch Interaction on Horizontal Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 154–161. DOI: http://dx.doi.org/10.1145/1385569.1385595

[8] Drini Cami, Fabrice Matulic, Richard G Calland, Brian Vogel, and Daniel Vogel. 2018. Unimanual Pen+Touch Input Using Variations of Precision Grip Postures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 825–837. DOI: http://dx.doi.org/10.1145/3242587.3242652

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*.

[10] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. DOI: http://dx.doi.org/10.1145/2807442.2807450

[11] Xinghao Chen. Awesome Hand Pose Estimation. https://github.com/xinghaochen/awesome-hand-pose-estimation. Accessed: 2019-09-01.

[12] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+Touch: Interweaving Touch & In-air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 519–525. DOI: http://dx.doi.org/10.1145/2642918.2647392

[13] Artem Dementyev and Joseph A Paradiso. 2014. WristFlex: Low-power Gesture Input with Wrist-worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 161–166. DOI: http://dx.doi.org/10.1145/2642918.2647396

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database.. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE Computer Society, 248–255. DOI: http://dx.doi.org/10.1109/CVPR.2009.5206848

[15] Nicholas Fellion, Thomas Pietrzak, and Audrey Girouard. 2017. FlexStylus: Leveraging Bend Input for Pen Interaction. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 375–385. DOI: http://dx.doi.org/10.1145/3126594.3126597

[16] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-operated Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 861–870. DOI: http://dx.doi.org/10.1145/1124772.1124898

[17] Yves Guiard. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior* 19, 4 (1987), 486–517.

[18] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2315–2324.

[19] Khalad Hasan, Xing-Dong Yang, Andrea Bunt, and Pourang Irani. 2012. A-coord Input: Coordinating Auxiliary Input Streams for Augmenting Contextual Pen-based Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 805–814. DOI: http://dx.doi.org/10.1145/2207676.2208519

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[21] Ken Hinckley, Xiang 'Anthony' Chen, and Hrvoje Benko. 2013. Motion and Context Sensing Techniques for Pen Computing. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 71–78. http://dl.acm.org/citation.cfm?id=2532129.2532143

[22] Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang 'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. 2014. Sensing Techniques for Tablet+Stylus Interaction. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 605–614. DOI: http://dx.doi.org/10.1145/2642918.2647379

[23] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + touch = new tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 27–36. DOI: http://dx.doi.org/10.1145/1866029.1866036

[24] Sungjae Hwang, Andrea Bianchi, Myungwook Ahn, and Kwangyun Wohn. 2013. MagPen: Magnetically Driven Pen Interactions on and Around Conventional Smartphones. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 412–415. DOI: http://dx.doi.org/10.1145/2493190.2493194

[25] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture Sensing Using On-Body Acoustic Interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 276:1—-276:13. DOI: http://dx.doi.org/10.1145/3290605.3300506

[26] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/2380116.2380139

[27] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Rui Li, Zhenyu Liu, and Jianrong Tan. 2019. A survey on 3D hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition* 93 (2019), 251–272.

[29] Michael Linderman, Mikhail A Lebedev, and Joseph S Erlichman. 2009. Recognition of handwriting from electromyography. *PLoS One* 4, 8 (2009), e6791.

[30] Shenwei Liu and François Guimbretière. 2012. FlexAura: A Flexible Near-surface Range Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 327–330. DOI: http://dx.doi.org/10.1145/2380116.2380158

[31] Fabrice Matulic and Moira Norrie. 2013. Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. ACM, St Andrews, UK, 41–50.

[32] Fabrice Matulic and Moira C. Norrie. 2012. Supporting Active Reading on Pen and Touch-operated Tabletops. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 612–619. DOI: http://dx.doi.org/10.1145/2254556.2254669

[33] Fabrice Matulic, Brian Vogel, Naoki Kimura, and Daniel Vogel. 2019. Eliciting Pen-Holding Postures for General Input with Suitability for EMG Armband Detection. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19)*. ACM, New York, NY, USA, 89–100. DOI: http://dx.doi.org/10.1145/3343055.3359720

[34] Fabrice Matulic, Daniel Vogel, and Raimund Dachselt. 2017. Hand Contact Shape Recognition for Posture-Based Tabletop Widgets and Interaction. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 3–11. DOI: http://dx.doi.org/10.1145/3132272.3134126

[35] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017a. SensIR: Detecting Hand Gestures with a Wearable Bracelet Using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 593–597. DOI: http://dx.doi.org/10.1145/3126594.3126604

[36] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017b. EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1923–1934. DOI: http://dx.doi.org/10.1145/3025453.3025807

[37] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

[38] S. K. Nayar. 1997. Catadioptric omnidirectional camera. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 482–488. DOI:http://dx.doi.org/10.1109/CVPR.1997.609369

[39] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. 2017. Thumb + Pen Interaction on Tablets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3254–3266. DOI: http://dx.doi.org/10.1145/3025453.3025567

[40] Raf Ramakers, Davy Vanacken, Kris Luyten, Karin Coninx, and Johannes Schöning. 2012. Carpus: A Non-intrusive User Identification Technique for Interactive Surfaces. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 35–44. DOI:http://dx.doi.org/10.1145/2380116.2380123

[41] Jef Raskin. 2000. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

[42] Jun Rekimoto. 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. DOI: http://dx.doi.org/10.1145/263407.263505

[43] T Scott Saponas, Desney S Tan, Dan Morris, and Ravin Balakrishnan. 2008. Demonstrating the Feasibility of Using Forearm Electromyography for Muscle-computer Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 515–524. DOI: http://dx.doi.org/10.1145/1357054.1357138

[44] Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. 2010. HandsDown: Hand-contour-based User Identification for Interactive Surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*. ACM, New York, NY, USA, 432–441. DOI: http://dx.doi.org/10.1145/1868914.1868964

[45] A. Seniuk and D. Blostein. 2009. Pen Acoustic Emissions for Text and Gesture Recognition. In *2009 10th International Conference on Document Analysis and Recognition*. 872–876. DOI: http://dx.doi.org/10.1109/ICDAR.2009.251

[46] Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and Gestures on a Multi-touch Pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1323–1332. DOI: http://dx.doi.org/10.1145/1978942.1979138

[47] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. DOI: http://dx.doi.org/10.1145/2642918.2647373

[48] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. 2009. Interaction Technique for a Pen-Based Interface Using Finger Motions. In *Human-Computer Interaction. Novel Interaction Methods and Techniques*, Julie A. Jacko (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 503–512.

[49] Marc Teyssier, Gilles Bailly, and Éric Lecolinet. 2017. VersaPen: An Adaptable, Modular and Multimodal I/O Pen. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2155–2163. DOI: http://dx.doi.org/10.1145/3027063.3053159

[50] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. 2008. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1371–1380. DOI: http://dx.doi.org/10.1145/1357054.1357269

[51] Andrew M. Webb, Hannah Fowler, Andruid Kerne, Galen Newman, Jun-Hyun Kim, and Wendy E. Mackay. 2019. Interstices: Sustained Spatial Relationships Between Hands and Surfaces Reveal Anticipated Action. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 588, 12 pages. DOI: http://dx.doi.org/10.1145/3290605.3300818

[52] Hongyi Wen, Julian Ramos Rojas, and Anind K Dey. 2016. Serendipity: Finger Gesture Recognition Using an Off-the-Shelf Smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3847–3851. DOI: http://dx.doi.org/10.1145/2858036.2858466

[53] Mike Wu and Ravin Balakrishnan. 2003. Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-user Tabletop Displays. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 193–202. DOI: http://dx.doi.org/10.1145/964696.964718

[54] Chao Xu, Parth H Pathak, and Prasant Mohapatra. 2015. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 9–14. DOI:http://dx.doi.org/10.1145/2699343.2699350

[55] Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. 2013. Surround-see: Enabling Peripheral Vision on Smartphones During Active Use. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 291–300. DOI: http://dx.doi.org/10.1145/2501988.2502049

[56] Zhongliang Yang and Yumiao Chen. 2016. Surface EMG-based sketching recognition using two analysis windows and gene expression programming. *Frontiers in neuroscience* 10 (2016), 445.

[57] Dongwook Yoon, Ken Hinckley, Hrvoje Benko, François Guimbretière, Pourang Irani, Michel Pahud, and Marcel Gavriliu. 2015. Sensing Tablet Grasp + Micro-mobility for Active Reading. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 477–487. DOI: http://dx.doi.org/10.1145/2807442.2807510

[58] Chun Yu, Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, and Yuanchun Shi. 2019. HandSee: Enabling Full Hand Interaction on Smartphone with Front Camera-based Stereo Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 705, 13 pages. DOI: http://dx.doi.org/10.1145/3290605.3300935

[59] X Zhang, X Chen, Y Li, V Lantz, K Wang, and J Yang. 2011. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41, 6 (nov 2011), 1064–1076. DOI: http://dx.doi.org/10.1109/TSMCA.2011.2116004

[60] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 167–173. DOI: http://dx.doi.org/10.1145/2807442.2807480

[61] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, and Ken Hinckley. 2019. Sensing Posture-Aware Pen+Touch Interaction on Tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 55:1—-55:14. DOI: http://dx.doi.org/10.1145/3290605.3300285

[62] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 5551–5560.