Towards Enabling Blind People to Independently Write on Printed Forms

Shirin Feiz*
Stony Brook University
sfeizdisfani@cs.stonybrook.edu

Syed Masum Billah*
Stony Brook University
sbillah@cs.stonybrook.edu

Vikas Ashok
Stony Brook University
vganjiguntea@cs.stonybrook.edu

Roy Shilkrot
Stony Brook University
roys@cs.stonybrook.edu

IV Ramakrishnan
Stony Brook University
ram@cs.stonybrook.edu

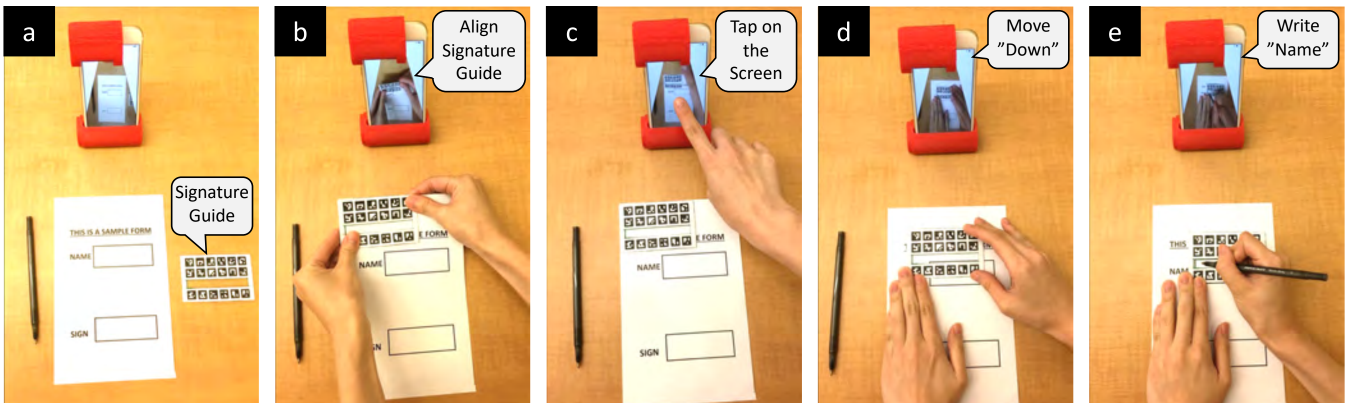

Figure 1: An illustration of how a blind user fills out a paper form with WiYG: The user (a) attaches 3D printed apparatus to the phone and places it in front of the paper; (b) aligns a signature guide to the top-left corner of the paper for calibration; (c) taps the screen to start receiving voice instructions for moving the signature guide to the first form-field; (d) follows the instructions and moves the signature guide; and (e) writes the information requested by the application, and then taps the screen to start receiving instructions for the next form-field. This process continues until the paper form is completely filled.

ABSTRACT

Filling out printed forms (e.g., checks) independently is currently impossible for blind people, since they cannot pinpoint the locations of the form fields, and quite often, they cannot even figure out what fields (e.g., name) are present in the form. Hence, they always depend on sighted people to write on their behalf, and help them affix their signatures. Extant assistive technologies have exclusively focused on reading, with no support for writing. In this paper, we introduce WiYG, a Write-it-Yourself guide that directs a blind user to the different form fields, so that she can independently fill out these fields without seeking assistance from a sighted person. Specifically, WiYG uses a pocket-sized custom 3D printed smartphone attachment, and well-established computer vision algorithms to dynamically generate audio instructions that guide the user to the different form fields. A user study with 13 blind participants showed that with WiYG, users could correctly fill out the form fields at the right locations with an accuracy as high as 89.5%.

*Equal Contribution

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300530

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; *Accessibility technologies*; *Accessibility systems and tools*; • **Hardware** → Emerging interfaces.

KEYWORDS

Blind; 3D Print, Paper, Form; Writing, Write-it-Yourself, WIY.

ACM Reference Format:
Shirin Feiz, Syed Masum Billah, Vikas Ashok, Roy Shilkrot, and IV Ramakrishnan. 2019. Towards Enabling Blind People to Independently Write on Printed Forms. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300530

1 INTRODUCTION

Although the Web has become the dominant medium for conducting digital transactions, printed materials such as paper forms, bank checks, contracts and credit-card receipts continue to abound in our daily lives. Hence, one still continues to do paper-centered transactions such as write out checks, fill out hospital forms and tax returns, sign receipts and contracts, and so on.

Any kind of paper-centered transaction is straightforward for sighted people. On the other hand, working with non-digital, standard printed materials has always been a challenge for blind people, especially writing. For instance, filling out forms on printed documents independently is currently impossible for blind people, since they cannot pinpoint where the form fields are located in the document, and quite often, they cannot even figure out what fields (e.g., name, address, etc.) are present on the form. Hence, they always depend on sighted people to write on their behalf as well as help them to affix their signatures at the right spots on the documents. Even for the latter, blind people resort to signature guides, a small rectangular card with a cutout space within, as an aid to write or sign in a straight line.

It is worthwhile mentioning that a number of assistive technology solutions that enable blind people to read printed materials have been developed. These technologies primarily employ OCR via cameras embedded in smartphones (KNFB Reader [2], Seeing AI [4]), Head-Mounted Displays [6], and custom-designed hardware (FingerReader [32]). All these technologies focus exclusively on reading printed content, and provide little-to-no support for writing. Besides, a bigger problem is that some of them, especially ones based on special purpose hardware designed to be worn on fingers [32], can interfere with writing.

Developing assistive technology aids to help blind people write on printed paper independently by themselves, is an important open-ended and technically challenging problem.

In this paper, we introduce *Write-it-Yourself Guide (WiYG)* for exploring the feasibility of using smartphones to *automatically guide blind people* to the different form fields in the printed document, and let them fill out these fields all by themselves without seeking any third-party assistance from sighted people.

WiYG uses a pocket-sized custom 3D printed smartphone attachment comprising two parts: *base* and *reflector*. The *base* serves as a phone stand that helps keep the phone upright; the *reflector* gets attached to the top of the phone and redirects the phone-camera's focus to the paper document that is placed in front of the phone (see Figure 2). WiYG uses this live camera feed to track the user's signature-guide movements over the form. Since visual markers yield more accurate tracking of objects, an *aruco pattern* [18] is wrapped around the signature guide to aid in tracking it. WiYG leverages state-of-the-art image processing algorithms to estimate the location of the user's signature guide on the document, and dynamically generates audio commands to instruct the user move the signature guide to the different form fields one-by-one.

When the user is ready to write, he places the signature guide at the top-left corner of the paper to facilitate camera calibration. Upon completion of the calibration, WiYG starts providing navigational instructions in audio (e.g., *move left*, *you are close*) for moving the signature guide to align with the first form field. Once the alignment is achieved, the user is notified and she fills out the form field. After writing, the user simply taps on the phone screen, and WiYG provides navigational guidance for the next form field. In this way, WiYG helps users complete the form themselves by guiding them to each form field one after another.

We assume that annotations specifying the locations of the field boundaries are given. However, automation of the annotation process for identifying form fields and their location boundaries can be done using Convolution Neural Networks, a gold standard for image recognition tasks (e.g., see [19, 30, 34, 37]).

WiYG can be used on the go, since the custom 3D printed smart-phone attachments can be easily carried in a pocket, purse, or a bag. Also, WiYG does not interfere with writing, since it does not require any custom hardware to be worn on fingers. Furthermore, to promote robustness, WiYG incorporates state-of-the-art image processing techniques for handling issues that may crop up during tracking, namely occlusion, accidental form displacements, shadows, partial views, and blurriness.

We summarize our contributions as follows:

- Pocket-size custom 3-D printed phone attachments for capturing live camera feed of the paper forms, as well as user actions.
- A vision-based guidance system to assist the user align their signature guide with various annotated form fields with a view to filling them out by themselves.
- Results from a user study that explored how 13 blind participants used WiYG to fill out different printed forms with varying number of form-fields.

2 RELATED WORK

Reading Aids

The use of braille, one of the oldest tactile-based read/write systems for blind users, has been steadily declining [28], hastened by advances in digital technologies, particularly speech, computer vision and OCR. A number of standalone reading applications, leveraging these advances, have emerged — examples include the pioneering Kurzweil Scanner [3] for desktops; KNFB Reader [2], Seeing AI [4], and Text Detective [9] for mobile phones; and FingerReader [32], OrCam [6], and HandSight [33] for mobile and wearable devices. OCR, that underpin all of these standalone reading applications have several limitations—they are unusable in poor lighting conditions; they require careful camera framing so that a target object is completely visible and centered within the camera's field of view [24, 26, 36]; they do not support complex documents and spatial data [25]; and cannot determine which blocks of text to read, and in what order [12]. Reading applications on wearables such as [6, 32, 33] address some of these limitations. But they have other problems, namely, they require specialized hardware and cameras, and the ones designed to be worn on fingers interfere with writing.

3D Printing

Assistive technology often needs to be customized for individuals with disabilities. Hurst and Tobias [23] explored motivations for creating do-it-yourself assistive technology. The advent of 3D printing has opened up enormous potential for rapid prototyping and customization of assistive technology solutions [15]. Buehler et al. [16] examine how organizations that serve people with disabilities use 3D printing tools. Hook et al. [22] also explore the role of 3D printing as a means to do-it-yourself assistive technologies targeted towards children with disabilities. Our work leverages 3D printing for a write-it-yourself assistive aid for blind people.

A number of other crowd-based systems make visual information accessible to blind people [13]. For example, "Be My Eyes" lets blind users seek assistance from sighted persons via a video call [1]. Similarly, the Aira glasses [8] let blind people seek assistance from sighted people by sending a live video feed from their glasses. The need to rely on sighted users does not arise in WiYG in so far as writing on printed materials is concerned.

Audio-Haptic Aids

Audio-haptic cues for way-finding and for exploration of maps have been used in numerous projects [10, 27, 29, 31]. These works use some combination of synthesized speech, sonification, and haptic patterns for effective discrimination [14, 21]. The drawback with these audio-haptic approaches is that they require extraneous hardware including several specialized haptic motors, ranging from 4 to 8, and mounted camera sensors. Pairing a smartphone with an off-the-shelf smartwatch that comes with built-in audio-haptic feedback is an attractive approach to solving the Write-it-Yourself problem without the need for such extraneous hardware, and thus has the potential to become an unobtrusive, viable mainstream writing aid for blind individuals. In our own work [11], we conducted a wizard-of-oz study with blind participants to gather requirements for the design of a smartphone-smartwatch based write-it-yourself aid for blind people. This paper is built on that work by providing the technology that automatically guides blind people in filling out paper forms.

3 WIYG SYSTEM

As shown in Figure 1, WiYG system consists of a signature guide, a pocket-size custom 3D printed attachment designed for a regular smartphone, and a smartphone application for guiding users' signature guide to the designated form fields. Users can write in these form fields with a regular pen.

Signature Guide

The signature guide, a rectangular card similar in size to that of a typical credit card, with dimensions $85.60mm \times 53.98mm$ (3.375×2.125 inches), and a rectangular opening of size $75mm \times 10mm$ (2.95×0.39 inches) cutout within, as shown in all of the sub-figures in Figure 1, is generally used by blind people to write in a straight line. WiYG attaches a paper print with special markers called Aruco board [18] on top of the signature guide to facilitate accurate and fast detection of the signature guide as the user moves it towards different form fields. Specifically, WiYG uses a 4×6 Aruco board, where each marker is drawn from a dictionary of 50 markers, and each marker comprises 4×4 bits [18]. However, all 24 markers cannot be used by WiYG on the signature guide due to the cutout space within it; only 18 markers are used. Using these many aruco markers makes the tracking robust to occlusion caused by the user's hand covering some portions of the signature guide while moving.

3D Printed Attachment

WiYG relies on a smartphone holder to function, which we specially designed and fabricated for this purpose. We designed the WiYG holder to prop the device at an angle that will allow capturing a full sheet of paper (A4, US Letter), while still allowing to interact with the touchscreen, i.e. the screen is facing forward. Personal fabrication techniques were used: 3D printing (FDM, PLA) and laser cutting (of mirror sheet acrylic). There were several considerations for the 3D design, to enable:

- Capturing a full sheet of paper.

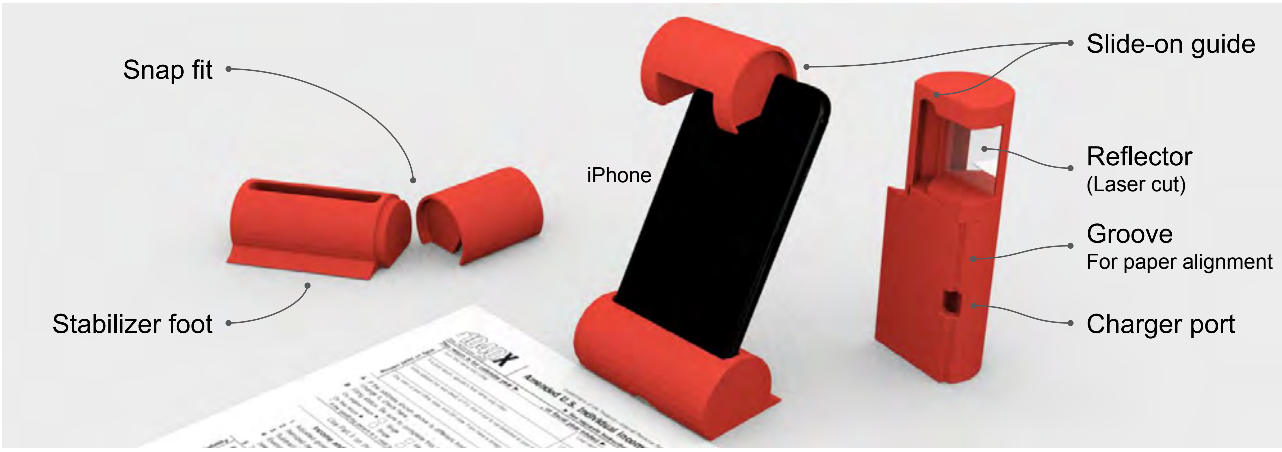

Figure 2: A 3D rendering of the phone holder base and top cover reflector, and their main features: snap fit connection, slide-on guide, stabilizer foot, and paper alignment groove.

- Interacting with the touchscreen.
- Aligning the paper to the holder.
- Easy yet robust and repeatable assembly on the phone.
- Stow-away, "pocketable" design.

These considerations informed the design of our 3D printed attachment. See an annotated illustration of the holder in Figure 2 (showing it propping up an iPhone as well as the main parts).

Design Parameters and Constraints. To achieve the goals, we iteratively experimented with 3D designs for the holder until we found the right configuration of iPhone device elevation, angle and reflector angle that allows the front-facing camera to capture a full sheet of paper. The iPhone's center of mass with the reflector housing was exceeding the holder base footprint and would topple over backwards, so we installed a "foot" to extend the base.

We aimed to design the holder so that the camera sees the top of paper where it meets the base. The final angle of the iPhone was determined to be 61° from horizontal, which is $90° - 1/2 \cdot 58°$ - the VFOV (vertical field of view) of the iPhone 7. The camera focal point has a virtual focal point at roughly 145mm from the ground, considering the reflector, which we denote O in Figure 3. The reflector offsets another 3.5°, and quick trigonometry reveals the distance from O's projection on the ground to the beginning of the capture frustum is: $\tan(29° - 3.5°) \cdot 145mm =$ **69mm**. This allows the iPhone's front camera to see the beginning of the paper and much beyond the end of the paper, however there the pixels/mm resolution drops rapidly. See Figure 3 for an illustration of these parameters.

We used a first-surface acrylic mirror sheet to reduce reflection aberrations. The sheet was cut preciously using a laser cutter to fit exactly in the top cover using parameters from the 3D design. Using a laser cutter also reduced the noise from machining.

The two-part phone holder can snap together and form a kind of barrel, so that the users can easily carry it in their pocket or purse. Other assistive features of the holder are the slide-on guide, which helps put the top cover in the precise position by sliding, and the paper alignment groove can be seen in Figure 2.

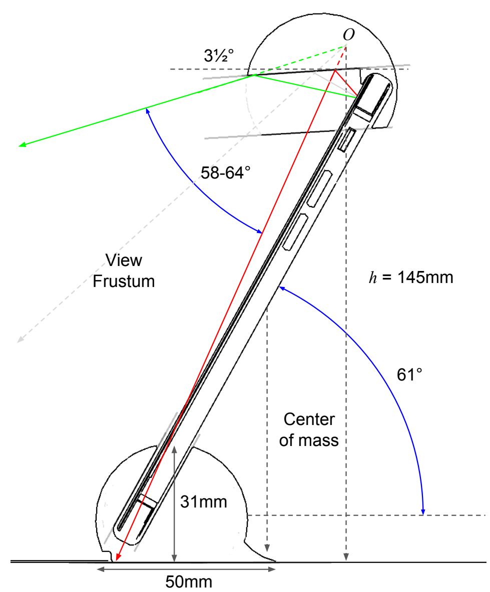

Figure 3: A cross section trigonometric analysis of the phone holder base and top cover. The virtual focal point O illustrates the extent of the front camera's frustum.

The Rationale Behind Two-Part Phone Holder Design. Only using a holder without a reflector attachment requires a steeper inclination of the camera. For example, in Figure 3, the angle needs to be ~49° instead of 61°, just to keep the paper centered in the camera image. Also, the holder becomes bulkier (~65mm instead of 50mm) to support a steeper inclination. Moreover, to cover the same field of view attainable with the mirror, the document has to be placed at least 126mm farther away from the holder, which makes it difficult to reach the apparatus as users have to interact with the touchscreen of the phone.

However, one drawback of this design is that the *width* of an A4 paper is not completely covered in the camera view — the two top corners of the paper cannot be seen. However, even if the *whole* signature guide is not visible, its aruco board still has plenty of visible markers in the field of view.

The Rationale Behind Not Using Phone's Back Camera. Similarly, the back-camera was not used because (i) we found that it was cumbersome to tap on a back-facing touchscreen; and (ii) we measured that the resolution of the front camera did not compromise tracking accuracy.

WiYG Smartphone Application

The WiYG application that runs on the smartphone has 2 main goals: (i) continuously detect the location of signature guide on the paper form (in the real world), assuming the top-left corner of the paper form as the origin $\langle 0, 0 \rangle$; and (ii) dynamically generate navigational instructions on how to move the signature guide to different form-fields in a form.

Detecting the Signature Guide. WiYG application analyzes the live video feed from the smartphone's front camera. We use a frame rate of 15 frames per second with a camera resolution of 750x1500. The application first determines the pixel location of the signature guide in each of the images of the input video stream, and then transforms this pixel location to the actual real-world location of the signature guide on the paper form, assuming the top-left corner of the form as the origin $\langle 0, 0 \rangle$. Note that we denote the location of the guide (both in the image and real-world) by a quadruple $\langle x_t, y_t, x_b, y_b \rangle$, where $\langle x_t, y_t \rangle$ is the top-left corner and $\langle x_b, y_b \rangle$ is the bottom-right corner of the cutout space within the signature guide. The detailed description of each steps in detection appear next.

Determining Pixel Location of Signature Guide in Input Image. Recall that WiYG uses Aruco board markers on the signature guide. The WiYG application detects these markers in each input image using the algorithm described in [18], and implemented in the OpenCV library [5, 7]. This algorithm returns the pixel location of each visible marker along with its unique marker code. The application then extrapolates this information to compute the pixel location of the signature guide in the image.

Transforming Pixel Location to Real World Location. Since the transformation between pixel location and real-world location is between two 2D planes (i.e., the image and the paper form), it should be of type homography [20]. Therefore, WiYG maps every pixel p in the input image to a point P on the actual paper form using a homography transformation matrix H_p as shown in equation 1.

$$p = H_p P, \forall p \in \text{image}, P \in \text{paper form} \quad (1)$$

Computing the Homography Transformation Matrix H_p. To estimate H_p, at least 4 pairs of matching points between the input image and the actual paper form are required, with additional matches leading to better estimation [20]. Therefore, to obtain these matching points, the application first requests the user to align the signature guide to the top-left corner of the paper form (see Figure 1 (b)), before any writing task begins. Aligning the signature guide to the top-left corner enables the application to match the physical locations of the aruco markers on the signature guide to the pixel locations of the same markers in the corresponding input image. The application then uses these matched points to estimate H_p by running an optimization technique [20] available in OpenCV. The average Re-projection error for H_p estimation was 1.29mm.

Handling Accidental Paper-Form Displacements. The user may accidentally move the paper form, and therefore H_p needs to be updated to account for this displacement. For this purpose, WiYG first employs the *Kanade-Lucas-Tomasi (KLT)* feature tracking algorithm [35] to track the paper movements. Specifically, the KLT algorithm compares two consecutive image frames and outputs a set containing pairs of matching points between the two consecutive image frames. Therefore, if the paper form has been displaced, its new position can be computed from the KLT output. To improve the accuracy of the KLT algorithm, WiYG also uses (a) *Random Sample Consensus (RANSAC)* [17] outlier detection to filter out noise such as shadows; and (b) a predefined color threshold to discard irrelevant matches corresponding to user's hands in the KLT output. Specifically, a threshold of 180 on a scale of 1 to 255 was applied on the hue channel. Using the matched pairs in the KLT output, WiYG estimates a homography transformation matrix H_i between the two consecutive 2D image frames. The application then updates H_p by simply multiplying its current value with H_i as shown in equation 2.

$$H_p = H_p H_i \quad (2)$$

The updated H_p is then used to detect the real-world location of the signature guide as explained earlier.

Dynamically Generating Navigational Instructions. After determining the location of the signature guide in real-world, WiYG application dynamically generates instructions for the user to move the signature guide to the various form-fields one-by-one. As mentioned earlier, in this work, we assume that the real-world locations of the various form fields are predefined. Automatic identification and extraction of these field locations based on either crowdsourcing or computer vision methods is a topic for future work.

WiYG uses the following speech instructions:

- Move ⟨*direction*⟩; direction ∈ {Up, Down, Left, Right}
- Keep moving ⟨*direction*⟩
- You are close
- You are very close
- Stop! Write ⟨*label*⟩; *label* ∈ {Name, Signature, etc.}
- Signature Guide not visible.

Note that WiYG guides the user only along horizontal (Left, Right) and vertical (Up, Down) directions. Therefore, given the current location $\langle x_c, y_c \rangle$ of the signature guide and the target location $\langle x_t, y_t \rangle$ of a form field, WiYG calculates a rectilinear path for guidance consisting of two components: one along the horizontal direction, and the other along the vertical direction. Since two such paths are possible (vertical direction first and then horizontal, or vice versa), WiYG chooses the vertical direction first if $|y_t - y_c| > |x_t - x_c|$, otherwise the horizontal direction. Choosing a direction in this way eliminates the risk of issuing instructions in staircase pattern. Also, WiYG dynamically readjusts the path if the user unintentionally deviates from the chosen path or overshoots the target.

If the distance to be traversed in a direction is longer than a predefined threshold, WiYG repeats the previous instruction (albeit with a change in verbiage to sound more natural) every 5 seconds to keep the user informed. When the distance between the signature guide and the target falls under some thresholds, "you are close", "you are very close" instructions are given to the users to ensure that they exercise cautious hand movements and thereby not overshoot the intended *target* location; in case the user misses the target, WiYG recalculates the path to get them back to the *target* location, and guides them along this path as described before.

Finally, when the users reach the target, WiYG announces the label associated with the target field (e.g., name, address, date, signature). Once the users finish writing in a field, they simply tap of the touchscreen to start receiving navigational instructions for the next field. If the application cannot capture Aruco markers on signature guide while guiding, "Signature Guide is not visible" notification is given to the users prompting them to move their hands to clear the view between the camera and the signature guide.

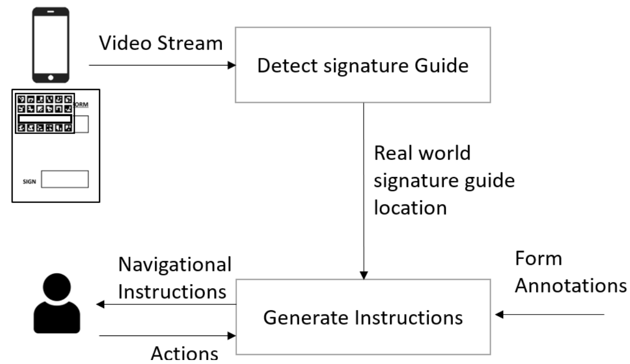

Figure 4: The work-flow diagram of WiYG System.

Putting It All Together. The workflow of WiYG system is shown in Figure 4. The smartphone application analyzes the video stream capturing the paper form and the signature guide to continuously detect the position of the signature guide, as the user moves it on the paper. To do this, it first detects the pixel location of the signature guide in each input image in the stream, and then uses a homographic transformation H_p to map this pixel location to a point on the actual paper form, assuming the top-left corner of the paper to be the origin. Using the real-world coordinates of the signature guide together with the that of the form-fields obtained from apriori human-annotations, the application continuously generates navigational instructions to guide the user to the fields one-by-one for writing.

4 EVALUATION

We conducted an IRB-approved user study to evaluate WiYG. Thirteen (13) participants (5 males, 8 females) were recruited,

ID	Age/Sex	Phone	Use SG?	Reading Technology
P1	51/F	Android	Yes	None
P2	30/F	iPhone	Yes	SeeingAI
P3	45/F	iPhone	Yes	SeeingAI, TapTapSee
P4	33/F	iPhone	Yes	SeeingAI
P5	44/M	Android	Yes	None
P6	35/F	iPhone	Yes	SeeingAI, TapTapSee
P7	65/F	iPhone	Yes	SeeingAI
P8	36/M	iPhone	Yes	SeeingAI
P9	38/M	iPhone	Yes	SeeingAI
P10	26/F	Android	Yes	None
P11	36/M	iPhone	Yes	SeeingAI, TapTapSee
P12	48/M	iPhone	No	None
P13	35/F	iPhone	Yes	SeeingAI

Table 1: Participant demographics. SG stands for Signature Guide.

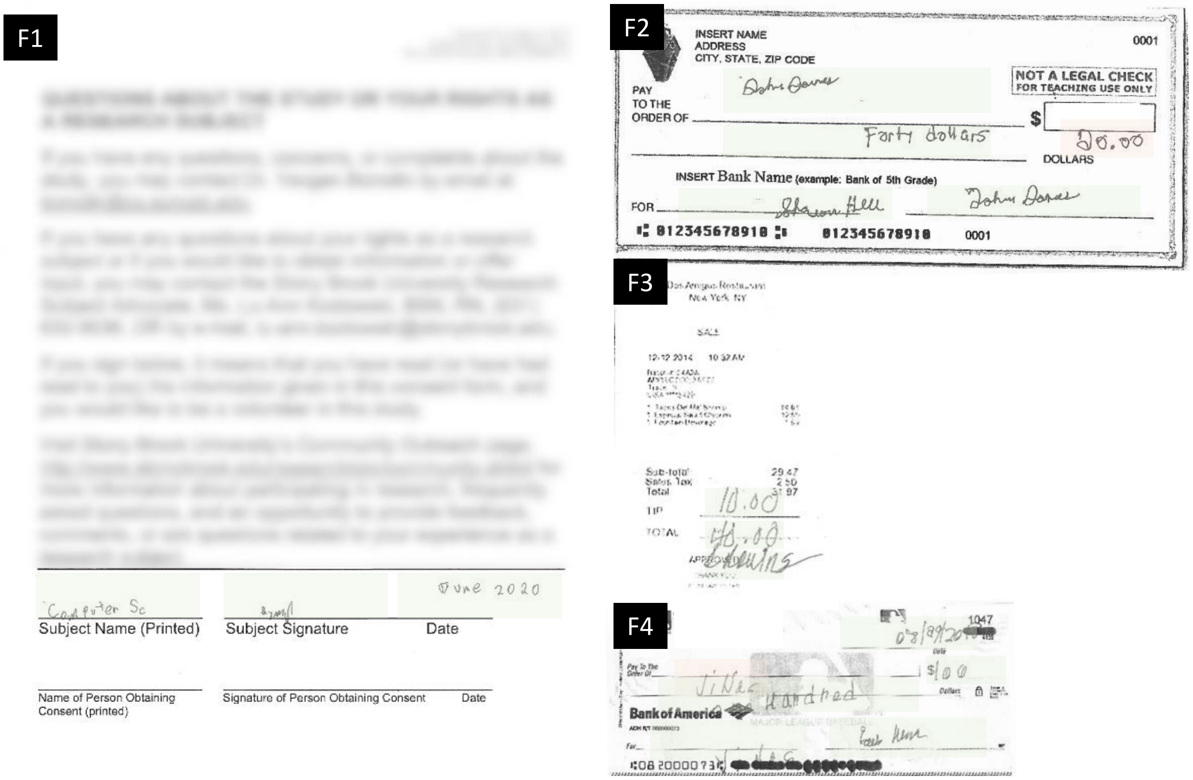

Figure 5: Forms used in the study - (F1) consent form, (F2) letter-sized bank cheque, (F3) restaurant receipt, and (F4) standard cheque - that were filled by different participants. Form-fields highlighted in green-shades were marked as correctly filled by human annotators, whereas the red-shaded ones were not.

with an average age of 40.2 (Median=36, SD=9.9, Range=26–65). All participants were completely blind. All of them knew how to write on paper, and they did not have any motor impairments that affected their interaction with WiYG. Table 1 presents the participant demographics. All participants owned smartphones (10 iPhone users and 3 Android users). We compensated each participant with a hourly rate of $25.

Apparatus

The experiment was conducted with the 3D printed attachment fixed to an iPhone 7 running the WiYG application. The signature guide used was black in color, and had Aruco markers pasted on its surface. For writing, a ball point pen was used by the participants.

Design

We conducted a repeated measures within-subject experiment. We designed real-world paper-form filling tasks that people routinely do in their everyday lives. Specifically, the participants were asked to fill out the following forms (see Figure 5) using WiYG:

- **F1**: A sample letter-size consent form. This form had 3 fields, namely, *Name*, *Signature*, and *Date*.
- **F2**: A large-sized sample bank cheque. This form had 7 fields namely, *Payee*, *Amount*, *Amount in words*, *Date*, *Memo*, and *Signature*.
- **F3**: A sample restaurant receipt. This form had 3 fields, namely, *Tip*, *Total*, and *Signature*.
- **F4**: A standard-sized sample bank cheque. This form had 7 fields namely, *Payee*, *Amount*, *Amount in words*, *Date*, *Memo*, and *Signature*.

The forms were chosen with different sizes, and different form-field arrangements. For instance, the fields in the consent form were horizontally placed next to each other towards the end of the form, and farther away from the smartphone camera. The fields in the cheques, on the other hand, were scattered throughout the forms with different dimensions; in the receipt, the fields were vertically arranged one below the other towards the bottom; however, they were still closer to the camera since the receipt was small in size.

To minimize the learning effect, we counterbalanced the ordering of tasks (i.e., forms). The quantitative metrics collected during a session included (i) task completion times; and (ii) accuracy of form filling, i.e., whether the user wrote the required information within a field-boundary in the form.

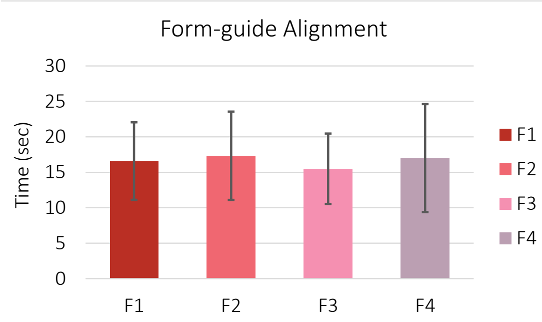

Figure 6: Time spent in aligning the signature guide to the top-left corner of the forms F1, F2, F3, and F4. Error-bars show ±1 SD.

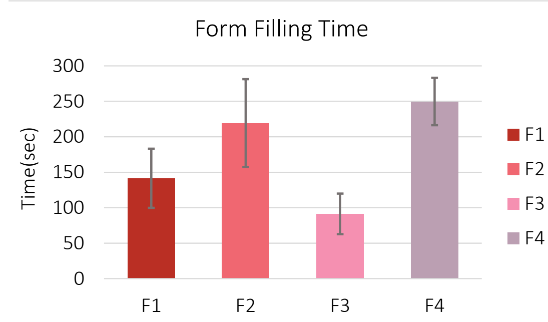

Figure 7: Time spent in moving the signature guide to different form-fields, as well as filling them out in forms F1, F2, F3, and F4. Error-bars show ±1 SD.

Each participant was allotted 10 minutes to complete each task. Every session was video recorded and lasted for 2 hours.

Procedure

At the beginning, the participants were given ~20 minutes to familiarize themselves with the 3D printed attachment, as well as with the WiYG interaction protocol. Specifically, they were asked to attach the two parts of our 3D-printed apparatus to the study iPhone, and to interact with WiYG application to fill out a few practice forms. Next, the participants started to conduct the study tasks by following the instructions given by the WiYG application. This included aligning the signature guide to the top-left corner of the form for calibration, and subsequently moving the signature guide according to the instructions until the first form-field was reached. After writing the required information in that field, they needed to tap on the iPhone screen to start receiving instructions on how to move the signature guide to reach to the next form-field. This process continued until all of the fields were filled; at that point, the application announced to the users that they had completed the task. All conversations during the study were in English. The experimenter also took notes during the session.

Result: Task Completion Time

The task completion time consisted of two components: (i) time spent in aligning the signature guide to the top-left corner of the form for initial calibration; and (ii) time spent in moving the signature guide to different form-fields, as well as writing the required information in those fields.

Time Spent in Aligning Signature Guide. On average, the participants spent 16.58s (SD=5.46s) for F1, 17.33s (SD=6.22s) for F2, 15.50s (SD=4.96s) for F3, and 17.00s (SD= 7.61s) for F4 in aligning the signature guide to the top-left corner of the form (see Figure 6). A one-way Anova test showed that there was no significant difference among these times ($F = 0.20, p = .89$). This result suggested that time spent in aligning the signature guide was invariant of form sizes.

Time Spent in Form Filling. Figure 7 shows the average times the participants spent in moving the signature guide to form-fields, as well as writing the required information in those fields for each study form. As shown in the figure, the participants spent 141.5s (SD= 41.69s) on F1, 219.25s (SD= 62.00S) on F2, 91.25 (SD= 28.62s) on F3, and 249.75 (SD= 33.42s) on F4. A one-way Anova test showed a significant difference in the form-filling times among the 4 forms ($F = 33.36, p \approx 0$). This was not surprising because the participants took more time on cheques (F2 and F4), where they had to fill out more fields as opposed to just 3 fields on consent (F1) and receipt (F3) forms.

Interestingly, even though the consent (F1) and receipt (F3) forms both had equal number of form-fields (i.e., 3), there was a significant difference between their completion times (based on Tukey's post-hoc HSD test ($p = .033$)). This difference in form-filling times was caused by the difference in form sizes. We observed that each user moved the signature guide at a steady pace. Factoring this out, it was easy to see that moving the signature guide over a larger form (e.g., F1) could take a longer time than over a smaller one (e.g., F3).

We also noticed that the participants relied on the edges of the forms to ensure that the relative alignment of the signature guide with respect to the edges was maintained while the signature guide was in motion. However, maintaining this relative alignment introduced a slight overhead to the overall form filling times.

Result: Accuracy

We measured the accuracy of WiYG both (i) objectively, using apriori annotations, and (ii) subjectively, using human evaluators.

Objective Measure: Accuracy Based on Apriori Form-Field Annotations. In this approach, we measured the accuracy as a

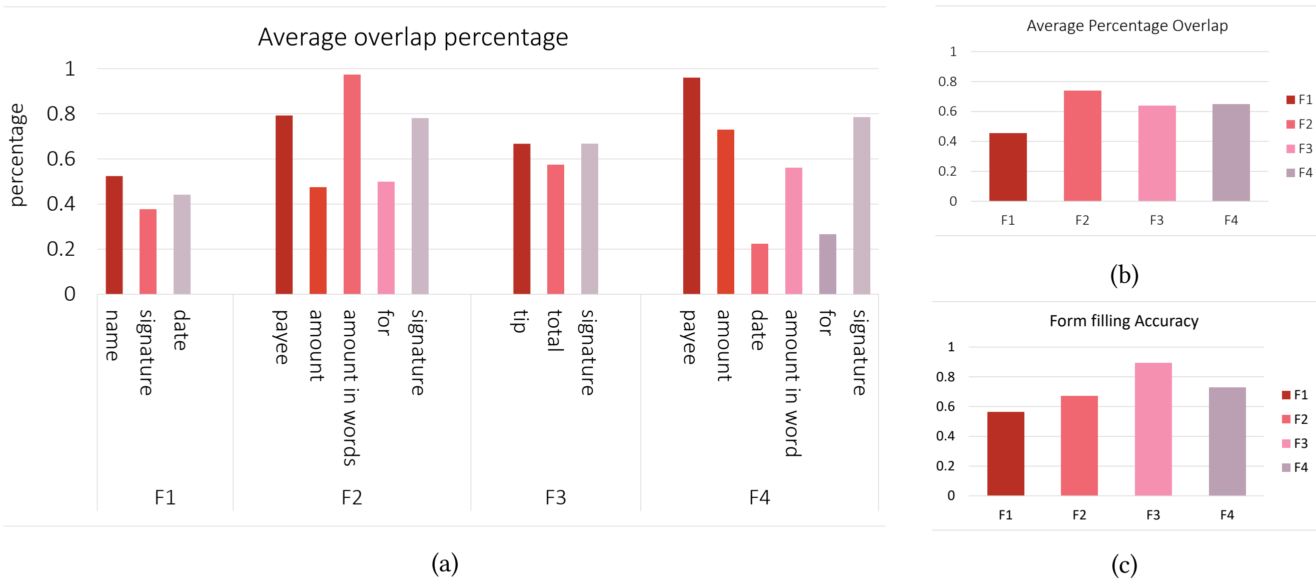

Figure 8: Percentage of overlaps for (a) each field, and (b) all fields in each form (F1, F2, F3, and F4); (c) Percentage of accuracy of all form-fields in each form as assessed by human evaluators.

percentage of overlap between the annotated rectangular region of a given form-field (apriori) and the rectangle enclosing the participant's written text in that field. Figure 8.(a) shows the average percentage of overlap for each form-field in four study forms, whereas Figure 8.(b) shows the average of percentage overlap of all form-fields in each form.

The differences in percentage of overlap among the 4 forms were found to be statistically significant (Kruskal Wallis test, $H = 17.81, p = .00048$). Pairwise comparisons using two-tailed Mann-Whitney U Test showed that the differences between F1 and F2 ($U = 426.5, p = .0001$), and between F1 and F3 ($U = 332.0, p = .00038$) were statistically significant.

Analysis of the collected data indicated that inaccuracies with F1 (consent form) were caused by two factors: form size and field locations. Specifically, since F1 was large in size and its fields were near the bottom, there was relatively a big distance between the camera and the fields when compared to other forms. Since the inaccuracies in image-to-world homography transformation estimates add up for points that are further away from the camera, instructions provided for F1 were less accurate.

Subjective Measure: Accuracy Based on Human Assessment. In this approach, we asked three human evaluators to assess whether the form-fields were correctly filled by the participants. The final verdict was made via majority voting. The inter-annotator agreement was high (Fleiss' $\kappa = 0.744$). Figure 8.(c) shows the average percentage of accuracy of all form-fields in each form based on human discretion.

The accuracy in subjective evaluation suggests that even though the average percentage of overlap was not high, a written form-field could still be considered as acceptable to human. For example, out of 56 form-fields that the participants filled, 25 (44.64%) of them had an overlap of less than 10% in objective measure, yet those were acceptable to human evaluators. Furthermore, note that WiYG achieved an accuracy as high as 89.5% for Form F3. Surprisingly, the subjective accuracy for Form F2 was lower than the objective accuracy. A closer inspection revealed that this stemmed from the prominent boundary enclosing Form F2's "amount" field which caused the human evaluators to annotate that field as incorrectly filled whenever the written content crossed this boundary.

Factors Affecting the Accuracy. We found that there were 3 major factors affecting the accuracy of WiYG: (i) slight perturbations during calibration at the beginning of the tasks, i.e., when the participants could not properly align the signature guide to the top-left corner of the study forms. This misalignment during calibration caused the image-to-world homographic transformation H_p to be slightly inaccurate, and therefore the guidance given by WiYG deviated from the actual target; (ii) sudden movements of the paper form that could not be captured between two consecutive input frames in camera. The homographic matrix was therefore not properly updated, thereby resulting in inaccurate instructions; and (iii) accidental movements of signature guide while writing, also contributed to errors.

Subjective Feedback

In our exit interview, all participants stated that they would use WiYG for filling out paper forms in this everyday life (mean score = 9.22, on a scale of 1 to 10 with 10 being most favorable response). Eight participants stated that WiYG would let them finish form-filling tasks faster, as they do not need to wait any external assistance. Also, P4 and P8 wanted to personalize WiYG by customizing instruction frequency and speech rate. They also wanted WiYG to detect faster movement of signature guide. Only P5 participant stated that he found it challenging to use WiYG for filling small paper forms where the fields were very close to each other.

5 DISCUSSION

Computing Homography Geometrically. It was possible to compute the homography H_p between the image and the paper sheet since the intrinsic parameters of a camera could be known, as well as the camera height, inclination, and the position of the paper. However, computing H_p in that way was not reliable for our setup because (i) we allowed some millimeter-range tolerance in 3D apparatus so that blind users can easily slide the phone into the holder base and the top cover. A tight fit could have made it difficult to set up the apparatus besides increasing the risk of scratches on the phone; and (ii) we also observed that depending on the wear and tear, the intrinsic camera parameters vary across different phones of the same model.

WiYG vs Crowdsourcing-based Solution. In contrast to WiYG, crowdsourcing-based solutions (e.g., Aira glasses [8]) rely on sighted crowd-workers, as well as on their availability. This may cause privacy leaks, as the crowd-worker can see partially filled sensitive personal information (e.g., SSN), whereas no such obvious privacy leaks arise with WiYG.

Subjective and Objective Accuracy Measures. Although both subjective and objective metrics were used to analyze the accuracy in this paper, we envision that for a fully automated system, only objective measures, based on an overlap threshold (e.g., 50% overlap) would be more suitable.

Robust Calibration. We noticed that calibration had an immense impact on accuracy of WiYG. Towards this, we will explore alternative calibration protocols that are more robust. One potential method is to obtain more matching points while estimating the homography as the quality of estimation is directly proportional to the number of matching points. We can obtain additional matching points by requesting the user to align the signature guide to all 4 corners of the paper form during the initial calibration.

Voice as an Alternative Input Modality. The current design of WiYG requires the user to tap on the phone every time they finish writing, in order to get the navigational instructions for the next form field. To tap on the phone, the users have to either put down their pen or remove their hand from the signature guide, and also lean forward; repeatedly doing this can get frustrating, especially if the form has many fields. 3 participants P4, P9, and P11 raised this concern after doing the study tasks. Voice input can potentially serve as a more convenient alternative to tapping on the phone. The users can simply say "Next", "Done", instead of tapping on the phone screen each time. However, the accuracy of voice-command interpretation is always subject to the noise in the surrounding environment.

Tactile Feedback as an Alternative to Speech. For people with smartwatches, and those who prefer quieter interaction, tactile feedback can serve as a potential alternative for the voice instructions. By assigning different haptic stimulations to different movement directions (Up, Down, Left, Right), instructions can be transmitted to the user in a tactile format via the smartwatch. In fact, we explored this idea in a Wizard-of-Oz study [11], where we found that with little practice and customization, the participants could accurately interpret the tactile instructions.

Annotation of Form Fields. In this work, we focused only on the HCI aspect of providing navigational guidance for blind users to help them move their signature guide to the different fields in the paper form. Automatically extracting the locations of these fields however, is orthogonal to the work presented in this paper. Given the recent advancements in image processing, especially in extracting regions of interest using Convolution Neural Networks (e.g., see [30, 34]), automatically identifying and extracting the labels and locations of form-fields is realizable in practice.

Alternative Design of Signature Guide. The signature guide we used was ideal for writing in a 10-20 characters long form-field (e.g., name, signature). However, many real-world forms contain check-boxes (e.g., gender), or multi-line (e.g., address) or multi-segment (e.g., phone number) fields. We will explore alternative designs for the guide, such as a guide having a small square cut-out as well as multiple cut-outs with different rectangular dimensions.

6 CONCLUSION

Write-it-Yourself aids have the potential to become a transformative assistive technology that can empower blind people to work with printed materials independently. It can possibly break up many barriers in their daily lives and open up many educational and employment opportunities. Towards that, this paper took a first step. Given a printed form, WiYG serves as the “seeing eyes” of a blind user by guiding her to all the form fields where she is supposed to write. The tangible pieces that make up this technology are a smartphone and a pocket-size attachment. Blind users can carry these pieces with them in their pockets and thus vastly expand the reach of their independence. In the future we envision a vastly expanded WiYG, in terms of functionality, to become the “go to” technology for blind people for all their writing needs on printed materials.

Acknowledgements. We thank anonymous reviewers for their insightful feedback that helped shape this paper. Hae-Na Lee contributed to the teaser figure, and Abhijeet Gulhane and Sushma Indluru helped in Form annotation. This research was supported in part by NSF: 1806076, NEI/NIH: R01EY02662, and NIDILRR: 90IF0117-01-00.

REFERENCES

[1] 2018. Be My Eyes: Bringing sight to blind and low vision people. https://www.bemyeyes.com/
[2] 2018. KNFB Reader. https://knfbreader.com
[3] 2018. KURZWEIL 1000 FOR WINDOWS. https://www.kurzweiledu.com/products/kurzweil-1000-v14-windows.html
[4] 2018. Microsoft AI. https://www.microsoft.com/en-us/seeing-ai/
[5] 2018. OpenCV: Open Source Computer Vision Library. https://github.com/opencv/opencv
[6] 2018. Orcam. https://www.orcam.com/en/myeye/
[7] 2018. Repository for OpenCV's Extra Modules. https://github.com/opencv/opencv_contrib
[8] Aira. 2018. Aira. https://aira.io
[9] AppleVis. 2018. Text Detective. https://www.applevis.com/apps/ios/utilities/text-detective
[10] Sandra Bardot, Anke Brock, Marcos Serrano, and Christophe Jouffrais. 2014. Quick-glance and In-depth Exploration of a Tabletop Map for Visually Impaired People. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine (IHM '14)*. ACM, New York, NY, USA, 165–170. https://doi.org/10.1145/2670444.2670465
[11] Syed Masum Billah, Vikas Ashok, and IV Ramakrishnan. 2018. Write-it-Yourself with the Aid of Smartwatches: A Wizard-of-Oz Experiment with Blind People. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 427–431. https://doi.org/10.1145/3172944.3173005
[12] Yevgen Borodin, Jeffrey P. Bigham, Glenn Dausch, and I. V. Ramakrishnan. 2010. More Than Meets the Eye: A Survey of Screen-reader Browsing Strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10)*. ACM, New York, NY, USA, Article 13, 10 pages. https://doi.org/10.1145/1805986.1806005
[13] Erin Brady, Meredith Ringel Morris, Yu Zhong, Samuel White, and Jeffrey P. Bigham. 2013. Visual Challenges in the Everyday Lives of Blind People. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2117–2126. https://doi.org/10.1145/2470654.2481291
[14] Stephen Brewster and Lorna M. Brown. 2004. Tactons: Structured Tactile Messages for Non-visual Information Display. In *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28 (AUIC '04)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 15–23. http://dl.acm.org/citation.cfm?id=976310.976313
[15] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J. Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K. Kane. 2015. Sharing is Caring: Assistive Technology Designs on Thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 525–534. https://doi.org/10.1145/2702123.2702525
[16] Erin Buehler, Shaun K. Kane, and Amy Hurst. 2014. ABC and 3D: Opportunities and Obstacles to 3D Printing in Special Education Environments. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '14)*. ACM, New York, NY, USA, 107–114. https://doi.org/10.1145/2661334.2661365
[17] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
[18] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.
[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.
[20] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.
[21] Jonggi Hong, Lee Stearns, Jon Froehlich, David Ross, and Leah Findlater. 2016. Evaluating Angular Accuracy of Wrist-based Haptic Directional Guidance for Hand Movement. In *Proceedings of the 42Nd Graphics Interface Conference (GI '16)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 195–200. https://doi.org/10.20380/GI2016.25
[22] Jonathan Hook, Sanne Verbaan, Abigail Durrant, Patrick Olivier, and Peter Wright. 2014. A Study of the Challenges Related to DIY Assistive Technology in the Context of Children with Disabilities. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14)*. ACM, New York, NY, USA, 597–606. https://doi.org/10.1145/2598510.2598530
[23] Amy Hurst and Jasmine Tobias. 2011. Empowering Individuals with Do-it-yourself Assistive Technology. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '11)*. ACM, New York, NY, USA, 11–18. https://doi.org/10.1145/2049536.2049541
[24] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham. 2011. Supporting Blind Photography. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '11)*. ACM, New York, NY, USA, 203–210. https://doi.org/10.1145/2049536.2049573
[25] Shaun K. Kane, Brian Frey, and Jacob O. Wobbrock. 2013. Access Lens: A Gesture-based Screen Reader for Real-world Documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 347–350. https://doi.org/10.1145/2470654.2470704
[26] Roberto Manduchi and James M. Coughlan. 2014. The Last Meter: Blind Visual Guidance to a Target. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3113–3122. https://doi.org/10.1145/2556288.2557328
[27] Roberto Manduchi and James M. Coughlan. 2014. The Last Meter: Blind Visual Guidance to a Target. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3113–3122. https://doi.org/10.1145/2556288.2557328
[28] NFB. 2016. Statistical Facts about Blindness in the United States. http://www.nfb.org/
[29] Martin Pielot, Benjamin Poppinga, Wilko Heuten, and Susanne Boll. 2011. A Tactile Compass for Eyes-free Pedestrian Navigation. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II (INTERACT'11)*. Springer-Verlag, Berlin, Heidelberg, 640–656. http://dl.acm.org/citation.cfm?id=2042118.2042179
[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
[31] Sonja Rümelin, Enrico Rukzio, and Robert Hardy. 2011. NaviRadar: A Novel Tactile Information Display for Pedestrian Navigation. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 293–302. https://doi.org/10.1145/2047196.2047234

[32] Roy Shilkrot, Jochen Huber, Wong Meng Ee, Pattie Maes, and Suranga Chandima Nanayakkara. 2015. FingerReader: A Wearable Device to Explore Printed Text on the Go. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2363–2372. https://doi.org/10.1145/2702123.2702421

[33] Lee Stearns, Ruofei Du, Uran Oh, Catherine Jou, Leah Findlater, David A. Ross, and Jon E. Froehlich. 2016. Evaluating Haptic and Auditory Directional Guidance to Assist Blind People in Reading Printed Text Using Finger-Mounted Cameras. *ACM Trans. Access. Comput.* 9, 1, Article 1 (Oct. 2016), 38 pages. https://doi.org/10.1145/2914793

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[35] Carlo Tomasi and Takeo Kanade. 1991. Detection and tracking of point features. (1991).

[36] Marynel Vázquez and Aaron Steinfeld. 2014. An assisted photography framework to help visually impaired users properly aim a camera. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 5 (2014), 25.

[37] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.