2021
The Journal on Technology and Persons with Disabilities

Real-Time Sign Detection for Accessible Indoor Navigation

Seyed Ali Cheraghi, Giovanni Fusco and James M. Coughlan
The Smith-Kettlewell Eye Research Institute
ali.cheraghi@ski.org, giofusco@ski.org, coughlan@ski.org

Abstract

Indoor navigation is a major challenge for people with visual impairments, who often lack access to visual cues such as informational signs, landmarks and structural features that people with normal vision rely on for wayfinding. We describe a new approach to recognizing and analyzing informational signs, such as Exit and restroom signs, in a building. This approach will be incorporated in iNavigate, a smartphone app we are developing, that provides accessible indoor navigation assistance. The app combines a digital map of the environment with computer vision and inertial sensing to estimate the user's location on the map in real time. Our new approach can recognize and analyze any sign from a small number of training images, and multiple types of signs can be processed simultaneously in each video frame. Moreover, in addition to estimating the distance to each detected sign, we can also estimate the approximate sign orientation (indicating if the sign is viewed head-on or obliquely), which improves the localization performance in challenging conditions. We evaluate the performance of our approach on four sign types distributed among multiple floors of an office building.

Keywords

Blindness, Low Vision, Visually Impaired, Accessibility, Navigation, Wayfinding

Introduction

Indoor navigation is a major challenge for people with visual impairments, who often lack access to visual cues such as informational signs, landmarks and structural features that people with normal vision rely on for wayfinding. The most widespread localization approach is GPS, which enables a variety of wayfinding tools such as Google Maps and BlindSquare, but it is only accurate outdoors. Dead reckoning approaches such as step counting using inertial navigation (Flores & Manduchi, 2018) can estimate relative movements indoors or outdoors without any physical infrastructure, but this tracking estimate drifts over time unless it is augmented by absolute location estimates. There are a range of indoor localization approaches, including Bluetooth beacons (Ahmetovic et al., 2016), Wi-Fi triangulation (Heater, 2017) and RFIDs (Ganz et al., 2010). However, all of these approaches incur the cost of installing and maintaining physical infrastructure.

Computer vision is a promising localization approach, but most past work in this area has either required special hardware (Hu et al., 2014) or detailed 3D models of the environment (Gleason et al., 2016) that are time-consuming to generate and make the approach vulnerable to superficial environmental changes (e.g., new carpeting or moved shelves). The iOS app Clew (Yoon et al., 2019) uses visual-inertial odometry (VIO), a function built into modern smartphones combining computer vision and inertial sensing (Kelly & Sukhatme, 2011), to perform dead reckoning. This approach has the advantage of requiring no model of the environment. However, while dead reckoning allows a blind user to retrace their steps from a destination they have already reached back to their starting point, on its own it doesn't provide guidance to a new destination and does not provide absolute localization.

We developed iNavigate, an accessible localization iPhone app (Fusco & Coughlan, 2020), which combines a 2D map, computer vision and VIO to estimate and track the user's location in an indoor environment. While this approach requires the user to either hold the smartphone or wear it (e.g., on a lanyard) with the camera facing forward while walking, the user doesn't need to aim the camera towards specific signs, which would be challenging for people with low or no vision. We demonstrated the feasibility of our approach with five blind travelers navigating an indoor space, with localization accuracy of roughly 1 meter.

Recently we added verbal turn-by-turn directions to iNavigate (Fusco et al., 2020), thereby creating an accessible wayfinding app that guides the user in real time towards a desired destination. We build on this work by using a more powerful recognition algorithm, YOLOv5, which is able to simultaneously recognize multiple types of signs, with only 8 training images per sign type (instead of the hundreds used before). This will make it easier to create a model for each building, which includes not only the map but also the ability to recognize selected signs inside the building. Moreover, in addition to estimating the distance to each detected sign, we can also estimate the approximate sign orientation (i.e., viewed head-on or obliquely), which can be used to improve the localization performance in challenging conditions.

Discussion

Overview of Sign Detection for Indoor Navigation

The ability to recognize informational signs provides information about the user's location on the map that complements other information sources. For instance, if a specific sign is recognized then the user must be in an area where the sign is visible. If the sign has known physical height and width, we can infer the user's distance from the sign, as well as the

approximate orientation of the sign (i.e., viewed head-on or obliquely). This means we can estimate the user's rough location relative to the sign.

Information inferred from sign detections is combined with other information acquired by the iNavigate app, enabling the user's location to be estimated within a meter or better accuracy. This additional information includes visual-inertial odometry (VIO), which fuses computer vision with inertial sensing to estimate relative movements (i.e., dead reckoning), and the locations of walls and other barriers on the map that constrain the estimated trajectory. Note that the fusion of multiple sources of data allows the app to disambiguate which one of multiple identical signs is currently in view.

While iNavigate already uses Exit sign detection as a source of localization information, the new approach we are pursuing provides more localization information. This additional localization information includes the recognition of multiple sign types (see Fig. 1) and estimation of the approximate sign orientation, in addition to the distance to each detected sign. Moreover, whereas the sign recognition algorithms we used in the past required hundreds of sample training images, we demonstrate good recognition results with only eight training images for each type of sign. The need for minimal training data will make it easier to deploy our app in a variety of new buildings, each with its own signs.

Fig. 1. The Four Types of Signs Recognized in this Paper.

Top row: Exit on left and restroom on right. Bottom row: COVID-19 mask on left and fire alarm box on right. The fire alarm box is actually a 3D object but in this paper we treat it as a sign since it is rectangular and nearly flat.

Sign Recognition Algorithm

Previously (Fusco et al., 2020) we used a deep learning model called U-Net (Ronneberger et al., 2015) to detect and segment Exit signs, enabling the distance to be estimated to each sign, with encouraging results. An advantage of U-Net is that it not only recognizes signs but gives precise pixel-by-pixel segmentations. These segmentations allow the contours of the sign to be delineated, which is useful for estimating quantities such as sign distance and orientation. Unfortunately, we have found that U-Net recognition suffers from increased false positive and false negative detections, and inaccurate segmentations, when it is trained on multiple sign types. A separate U-Net could be trained on each sign type, but this approach would be too slow for real-time use for more than just a few sign types.

By contrast, the recently released YOLOv5 object recognizer (Nelson & Solawetz, 2020) is powerful enough to simultaneously recognize many types of signs and runs in real time on a smartphone (several seconds per frame or faster). Our experiments with YOLOv5 show that it is well suited to recognizing a variety of sign types, using only a small number (eight are used in this work) of training images for each type of sign. Note that each training image includes both the target sign of interest, cropped to demarcate a positive example of the sign, and the visual context around the sign (including other objects in the scene), which is used to provide negative examples of imagery to be distinguished from the sign itself. A limitation of YOLOv5 is that it returns an "xy axis-aligned" bounding box (Fig. 2b) around the sign, i.e., a rectangle with sides parallel to the x- and y- axes of the image, instead of a precise pixel-by-pixel segmentation. Fortunately, we have found that the bounding box fits tightly around the rectangular sign, especially since we use the estimated camera roll to undo any camera rotations (Fig. 2a,b) that would make the sign borders appear far from horizontal or vertical.

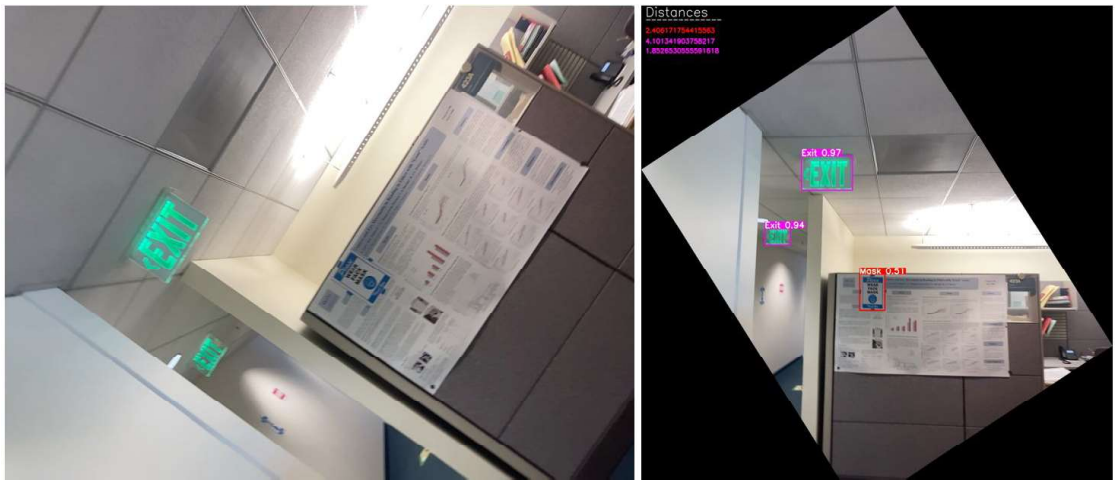

Fig. 2. Sample Image and Sign Detections.

Left to right: (a) Original image taken by iPhone SE shows that the camera is highly "rolled," i.e., horizontal lines in the scene appear far from horizontal in the image. (b) Using the roll angle estimated by the iPhone, the image is unrolled so that horizonal lines appear

horizontal. The black triangular regions near the borders correspond to unknown pixels in the unrolled image. YOLOv5 detections are drawn as bounding boxes around one mask sign and two Exit signs. Note that the bounding boxes estimated by YOLOv5 are aligned to the x- and y- axes and fit tightly around the actual signs.

In this work we consider four types of signs to be recognized. The first three are true signs and the fourth is a 3D object that is similar to a sign: the Exit sign, restroom sign, COVID-19 mask sign and a fire alarm box, respectively (Fig. 1). The fire alarm box is a 3D object that is shaped roughly like a small rectangular sign protruding from the wall; we chose this as a "sign" type both because it is an important feature in our building and also to explore how well YOLOv5 works on non-flat objects. We will explore using 3D objects such as water coolers, vending machines, and hand sanitizer stations as visual landmarks in more detail in future work. In the future we will also explore the trade-offs of adding more sign types to YOLOv5, including the trade-off between the rarity of a sign and the amount of localization information it provides.

Distance Estimation

We apply the distance estimation approach we described in (Fusco et al., 2020) to the output of the YOLOv5 algorithm. This approach uses the apparent height of the sign in the image, compared with the sign's known physical height, to estimate the distance using laws of perspective. It relies on three key assumptions (Fusco et al., 2020):

1. The sign is flat and rectangular, with known physical height (e.g., in cm).
2. It is mounted so that the sign lies in a vertical plane, with the borders of the sign horizontal or vertical with respect to gravity.
3. The camera pitch (angle that the camera line of sight makes with respect to the horizontal plane) and roll (the angle the camera is rotated about its line of sight, with

0° and 90° corresponding to portrait and landscape orientations, respectively) are known.

Fortunately, these assumptions are satisfied for our application. The signs we consider are rectangular, with a standard size, and they are mounted in a way that satisfies assumption 2. (The fire alarm box isn't a true sign, but it is shaped roughly like one.) Moreover, the camera pitch and roll are estimated in real time on modern smartphones using the built-in inertial measurement unit (IMU). Finally, note that the distance estimated by our algorithm is the *straight-line distance along the floor* formed by projecting the 3D camera and sign locations down onto the floor.

Orientation Estimation

A new feature we are exploring is to estimate the sign's orientation to the user by measuring how foreshortened the sign appears in the image. The rough orientation angle is determined by comparing the aspect ratio of the sign in the image with the sign's physical aspect ratio. While approximate, this estimate, when combined with the distance estimate, allows us to estimate the user's rough location relative to the sign. This approach makes the same three key assumptions described in the previous sub-section, Distance Estimation, augmented by knowledge of the physical width of the sign.

More specifically, assuming that the distance to the sign is significantly greater than the height difference between the camera and the sign, the dominant factor that determines the apparent aspect ratio of the sign is the amount of horizontal foreshortening. No foreshortening occurs if the sign is viewed head-on (i.e., the orientation angle θ is 0°). By contrast, significant foreshortening of the apparent width relative to the apparent height (resulting in the bounding box of the sign having a taller, skinnier aspect ratio than the sign's physical aspect ratio) occurs

for greater orientation angles (with $\theta \geq 60°$ corresponding to highly oblique viewpoints). We can roughly estimate θ by noting that the amount of foreshortening changes the physical aspect ratio by a factor of $\cos \theta$; this factor can be estimated by comparing the observed aspect ratio of a bounding box detection with the sign's physical aspect ratio. However, since $\cos \theta = \cos (-\theta)$, we can estimate θ only up to an unknown plus or minus sign. This two-fold ambiguity corresponds to the inability to distinguish a sign slanted to the left from a sign slanted an equal angle to the right. (This ambiguity could be resolved for sufficiently close-up signs by observing which of the two vertical sides of the sign appears shorter. However, this approach is unreliable unless the sign is nearby, and we leave this for future work.)

Performance Evaluation on Image Datasets

We acquired three image datasets of the top three floors of the main Smith-Kettlewell building. Because of restrictions due to the COVID-19 pandemic, we were unable to invite visually impaired volunteers to take images. Instead, the experimenter (one of the co-authors) took pictures using an iPhone SE running a data logging app that recorded about 3-5 images per second, along with the pitch, yaw and roll estimated for each image.

The first image dataset was acquired to evaluate the effectiveness of the YOLOv5 object recognition algorithm in terms of *precision* (the proportion of detections that correspond to an actual sign of the corresponding type in the image) and *recall* (the proportion of signs visible in the image that are correctly detected); see (Davis & Goadrich, 2006) for definitions of these measures. For this purpose, the experimenter walked around all three floors of the building while holding the iPhone pointing in the forward direction (in portrait orientation), roughly simulating how a blind person might hold the iPhone while using iNavigate as we observed in our earlier work. The experimental results are shown in Table 1, based on 810 images acquired in total. We

note that the recall values are significantly lower than 1.0, but the precision values are close to 1.0. This is appropriate for our planned integration with iNavigate, in which a sign needs only to be correctly recognized in a few frames for it to provide useful localization information; iNavigate can recover from occasional false negative detections.

Table 1. Recall and precision for each class of sign recognition.

Table includes numbers of true positives (TP), false positives (FP) and false negatives (FN), recall and precision. Recall is defined as TP/(TP + FN) and precision as TP/(TP + FP).

Sign type	**TP**	**FP**	**FN**	**Recall**	**Precision**
Exit	458	14	71	0.87	0.97
Restroom	72	0	79	0.48	1.0
Mask	197	5	184	0.52	0.98
Fire alarm	58	2	29	0.67	0.97

The second image dataset, totaling over 8000 images, was acquired to evaluate the distance estimation algorithm. The need for ground truth (actual) distances meant that the experimenter acquired images while standing at rest in multiple locations throughout the top three floors of the building, for which the distances to nearby signs were measured by tape measure. To challenge the distance estimation algorithm, the iPhone was held at multiple angles (e.g., portrait or landscape orientation, upside down, or any roll angle in between), and the hand was sometimes moved to induce the kind of motion blur that often arises in the use of iNavigate in real-world conditions. Given this hand motion, we estimate that the ground truth distances were known to roughly 20 cm accuracy.

To evaluate the performance of our distance estimation algorithm, we estimated the percent distance estimation error, defined as $E=|e-a|/a$ (expressed as a percentage), where e = estimated distance and a = actual distance. The median value of E is reported in Table 2, where it is broken down by sign type and by distance bins (i.e., signs whose actual distance is under a

certain threshold are included in the first bin, etc.). These statistics only include distance estimates for signs that are detected, and we note that distance estimates can be distorted if the sign is cut off in the image (which makes it look smaller than it should), or if the bounding box is inaccurately estimated. Because of a bug in the camera logging app, a small number of images had to be discarded because the corresponding roll and pitch values were incorrectly logged. Overall, we find that the median value of *E* is typically under 10% for most signs, with higher values for Exit signs and nearby signs (we are exploring possible explanations for these trends).

Table 2. Median distance estimate error broken down by sign type and actual distance to sign.

Sign type	**Dist < 3 m**	**3 m ≤ Dist < 5 m**	**5 m ≤ Dist < 10 m**	**Dist ≥ 10 m**
Exit	14.3%, 887	12.7%, 586	10.8%, 1129	13.5%, 161
Restroom	5.8%, 340	4.1%, 363	3.9%, 346	N/A
Mask	11.3%, 996	3.7%, 437	3.9%, 453	2.4%, 13
Fire alarm	7.8%, 1303	9.1%, 443	N/A	N/A

Each cell of the table indicates the median percent distance estimation error (see text for details) and the total number of sign detections included in the cell. N/A indicates that no sign detections are available for a cell.

The third image dataset was acquired for a preliminary evaluation of our orientation estimate algorithm. The dataset is small, consisting of just 51 images, with a mask sign and fire alarm box visible in each; the two signs shared the same 2D position (when projected to the floor). The experimenter stood in place at five locations, each a few meters away from the two signs, and aimed the iPhone while holding it in portrait orientation to capture both signs in each photo. The ground truth orientation for each image was either head-on ($\theta_g = 0°$), oblique ($\theta_g = \pm 45°$), or very oblique ($\theta_g = \pm 63°$). The sign orientation estimates are unable to determine whether the orientation angle is positive or negative, so we evaluate the estimation error as follows: $| |\theta_g| -$

θ|, where θ is the (unsigned, assumed non-negative) orientation estimate. Fig. 3 shows histograms of the estimation error, broken down by sign type and ground truth orientation. The error is fairly low for the mask sign but poor for the fire alarm (except in the head-on case when the orientation was correctly estimated). The mask sign is not only larger than the fire alarm, but more important, it is almost perfectly flat, both of which imply a more accurate prediction of the apparent aspect ratio and thus the orientation. We will measure the orientation accuracy for other sign types in the future, at a range of viewing distances, and will explore possible ways to make the orientation estimate usable for non-flat signs.

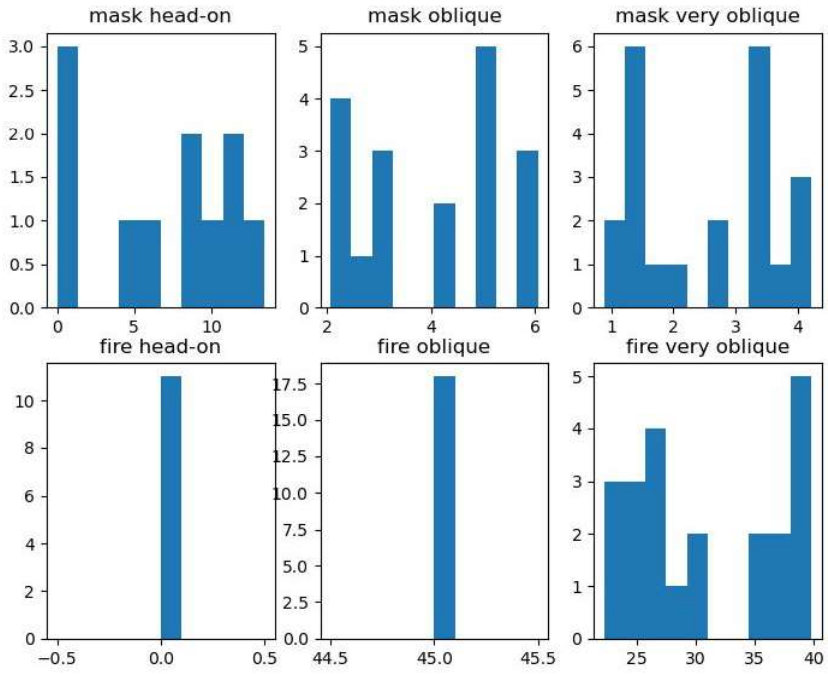

Fig. 3. Orientation Estimation Error Histograms for the Mask Sign and Fire Alarm Box.

Each histogram shows counts on the y-axis as a function of absolute orientation estimation error in degrees on the x-axis. The title of each histogram indicates the sign type (mask on top row and fire alarm on bottom row) and the ground truth orientation of the sign relative to the camera: head-on ($\theta_g = 0°$), oblique ($\theta_g = \pm 45°$) and very oblique ($\theta_g = \pm 63°$). The error is fairly

low for the mask sign but poor for the fire alarm (except in the head-on case when the orientation was correctly estimated).

Conclusions

We have demonstrated a new approach to sign detection that is useful for indoor navigation. Our approach allows real-time detection of multiple sign types along with distance and sign orientation estimates that provide useful information about the user's location. Experimental results demonstrate the feasibility of the approach. Our past work with an early version of our wayfinding app, iNavigate, established its usability by blind users. In the future we will integrate our new approach in the app, and we will perform ongoing tests with visually impaired participants as soon as current pandemic restrictions lift.

Acknowledgments

This work was supported by NIH grant 1R01EY029033 and NIDILRR grant 90RE5024-01-00. SAC was supported by Smith-Kettlewell's CV Starr Fellowship.

Works Cited

Ahmetovic, Dragan, et al. “NavCog: turn-by-turn smartphone navigation assistant for people with visual impairments or blindness.” *Web for All Conference*. 2016.

Flores, German, and Roberto Manduchi. (2018, April). Easy return: an app for indoor backtracking assistance. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-12).

Fusco, Giovanni, and James M. Coughlan. “Indoor Localization for Visually Impaired Travelers Using Computer Vision on a Smartphone.” *Web for All Conference: Automation for Accessibility*. 2020.

Fusco, Giovanni, et al. “An Indoor Navigation App using Computer Vision and Sign Recognition.” *Conference on Computers Helping People with Special Needs*. 2020.

Ganz, Aura, et al. “INSIGHT: RFID and Bluetooth enabled automated space for the blind and visually impaired.” *IEEE Engineering in Medicine and Biology*. 2010.

Gleason, Cole, et al. “VizMap: Accessible visual information through crowdsourced map reconstruction.” *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 2016.

Heater, B. “Apple maps gets indoor mapping for more than 30 airports.” TechCrunch.com. 14 December, 2017, https://techcrunch.com/2017/12/14/apple-maps-gets-indoor-mapping-for-more-than-30-airports/ Accessed 17 Nov 2020.

Hu, Feng, Zhigang Zhu, and Jianting Zhang. “Mobile Panoramic Vision for Assisting the Blind via Indexing and Localization.” Workshop on Assistive Computer Vision and Robotics. 2014.

Davis, Jesse, and Mark Goadrich. "The relationship between Precision-Recall and ROC curves." *Proceedings of the 23rd International conference on Machine learning*. 2006.

Kelly, Jonathan, and Gaurav S. Sukhatme. "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration." *The International Journal of Robotics Research* 30.1 (2011): 56-79.

Nelson, J. & Solawetz, J. "YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS." Roboflow.com. https://blog.roboflow.com/yolov5-is-here Accessed 17 Nov 2020.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Yoon, Chris, et al. "Leveraging Augmented Reality to Create Apps for People with Visual Disabilities: A Case Study in Indoor Navigation." *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 2019.