Perfect Accuracy with Human-in-the-Loop Object Detection

Rorry Brenner(✉), Jay Priyadarshi, and Laurent Itti

Computer Science and Neuroscience,
University of Southern California, Los Angeles, USA
{rorry.brenner,jpriyada,itti}@usc.edu

Abstract. Modern state-of-the-art computer vision systems still perform imperfectly in many benchmark object recognition tasks. This hinders their application to real-time tasks where even a low but non-zero probability of error in analyzing every frame from a camera quickly accumulates to unacceptable performance for end users. Here we consider a visual aid to guide blind or visually-impaired persons in finding items in grocery stores using a head-mounted camera. The system uses a human-in-the-decision-loop approach to instruct the user how to turn or move when an object is detected with low confidence, to improve the object's view captured by the camera, until computer vision confidence is higher than the highest mistaken confidence observed during algorithm training. In experiments with 42 blindfolded participants reaching for 25 different objects randomly arranged on shelves 15 times, our system was able to achieve 100 % accuracy, with all participants selecting the goal object in all trials.

Keywords: Scene understanding · Quality of life technologies · Sensory substitution · Mobile and wearable systems · Applications for the visually impaired · Egocentric and first-person vision · Computer vision · Object detection

1 Introduction and Background

People who are blind have more difficulty navigating the world than those with sight, even in places they have been before [8,23]. This is a condition that affects 39 million people worldwide [32]. Much progress has been achieved in developing electronic travel aids to assist them as technology has advanced. One method is to convert images to soundscapes which some subjects can learn to interpret well enough to differentiate places, and to identify and locate some objects [27]. Others include localization in an environment using stereo cameras, accelerometers, and even wifi access points [6,13]. Advances have also been made to traditional aids such as canes, by developing electronic replacements using, e.g., sonar to increase their warning range or grant the same feedback but without a physical cane [20,31], and replacing guide dogs with robots [16]. Among these devices

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part II, LNCS 9914, pp. 360–374, 2016.
DOI: 10.1007/978-3-319-48881-3_25

many utilize computer vision to help with navigation, text reading, and object recognition. [1,18–20,29].

Many advances have been made in computer vision, yet, even state of the art algorithms have not yet been able to achieve perfect accuracy on standard datasets [7,12,28]. Our algorithm's success is founded in the areas of dynamic thresholding and active vision [2]. Active vision is the process of changing views to better identify what is being looked at. This can be through changing the pose of the camera or choosing a region of interest with a larger field of view and then attempting identification within that region using a zoomed-in image [3, 5,11,30]. Dynamic thresholding is any recognition system which has a decision threshold more complicated than a single number. For example, some methods include different thresholds for parts vs. whole object detection [9], adaptive local thresholding [14,33], and connectivity based thresholding [22].

Despite the discussed advances in assistance for the blind, shopping can still be a nearly impossible task. Many boxed and canned items have identical shapes, which means without one of these aids, or normal vision, help from a person with vision is required for selecting the correct item [17]. Even successful devices such as OrCam [19] require the user to point at the desired object to be identified. This is great for people with poor vision, but not helpful for the fully blind. To address this, systems have been proposed that read barcodes [21] or identify items on the shelves using computer vision algorithms [18,29]. On the one hand, barcode scanners never make mistakes, although they can be tedious to use when looking for a specific item in a large grocery store (as shown in our own results, see below). On the other hand, a serious problem with using a computer vision system for this application is that if they make too many mistakes, users will likely stop using them [10,24]. An acceptable system cannot ever tell the user to select an item they do not want.

2 Motivation

In a typical object detection computer vision system each input image requires the system to determine a confidence for how likely it is that any items trained for are currently in that image. If the confidence is high enough it will tell the user it has found the item. However, no matter where the confidence threshold is set, for most objects and algorithms there will be some range of values where the system will make mistakes [18], either false alarms or misses. If the threshold is set too high the system can decide it has not found the item when it was present (miss), and if the threshold is set too low and the system can decide it has found the item when the item was not present (false alarm). This problem happens with almost every system with a confidence threshold for detection because there often are some images without a particular item where the confidence may be higher than for some images with the item.

To show this point using the set of 25 objects used in our experiments below (Fig. 1), a dataset of pictures was collected in our simulated grocery store setting. A camera was placed in a fixed position and objects were arranged in front of

Fig. 1. Template images for objects in database by row, left to right. 1, Cereal: PEB, CP, HNC, LUC, MGC. 2, Snacks: SR, HBN, OCP, PS, NB. 3, Pasta: HH, KRA, PR, MAC, VEL. 4, Tea: SM, LIP, FTS, FR, STA. 5, Candy: NRD, HT, GNP, MD, JM.

it with their centers two feet away from the camera. Objects were then rotated vertically and horizontally at 15° degree intervals for each picture from negative 45° to positive 45° offset giving a total of 1225 images. Images in which the object pose (homography, discussed in the following section) could not be recovered by our algorithm were not included because the system would not be able to guide the user from those images. Removing these images left a total of 1112 usable images or 44.5 images on average per item. Figure 2-A shows receiver operating characteristic (ROC) curves for recognizing each of the objects in the dataset individually and Fig. 2-B shows one ROC curve over all objects. Confidences were calculated using the SURF [4] algorithm. Some objects had less of a problem than others, with a smaller portion of overlap between the highest confidence without the object and the lowest confidence with the object. Only 2 objects had no overlap at all. This means just these 2 objects of 25, with the images collected, would yield no mistakes with a fixed threshold. The ROC curves

for some of the other items are quite good as well; however, even an error rate of only 1 %, might cause an error every 25 s in our system that runs in real time at approximately 4 frames per second. In the discussion section we will detail why every mistake is a large issue for the user. The only way to solve this problem is to not have a yes/no threshold, and instead allow the system to output that it is unsure within this range of values where there will be uncertainty.

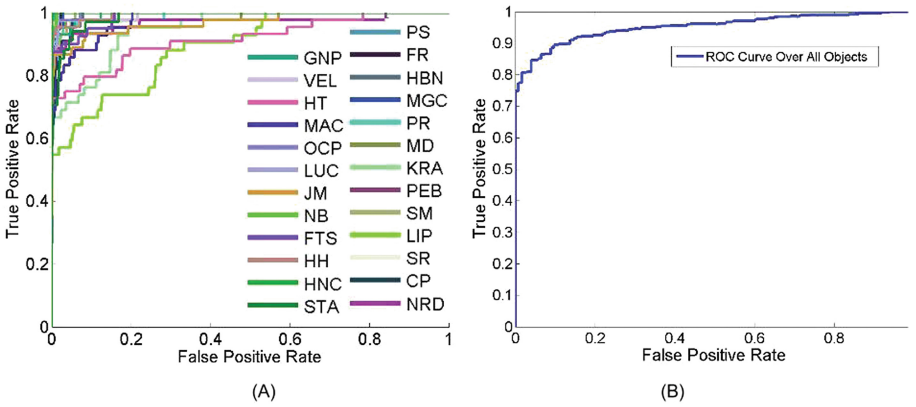

Fig. 2. (A) ROC curves of confidence values over all images and all objects collected. PEB and HNC are the only ones where all confidences for images of other objects are lower than all confidences for images of themselves. (B) ROC curve for correctness with a single fixed threshold over all objects being tested on.

3 Proposed System

The proposed system consists of a camera mounted on a pair of glasses, which captures images in real time. Users can provide instructions as to which object they want to reach for next (in experiments, that was controlled by the experimenter). Camera images are then analyzed as the user moves through the environment until at least some weak evidence for the presence of the object is determined by the vision algorithm. If there is evidence that the object may be present, but the system is uncertain (as further detailed below), the user is not yet told that the object has been found. Instead the user is instructed to turn, move, strafe, or crouch in a way that will decrease the difference in object pose between the current camera view and the system's template image for that object. Template images for the objects are front and centered. As the viewpoint changes and provides increasingly more front and centered views of the object, confidence of the vision algorithm is expected to increase. When confidence exceeds the threshold necessary to ensure no mistake, the object is declared found. The user may still be further guided so that the object becomes centered in the camera's field of view. At this point the user is informed that the

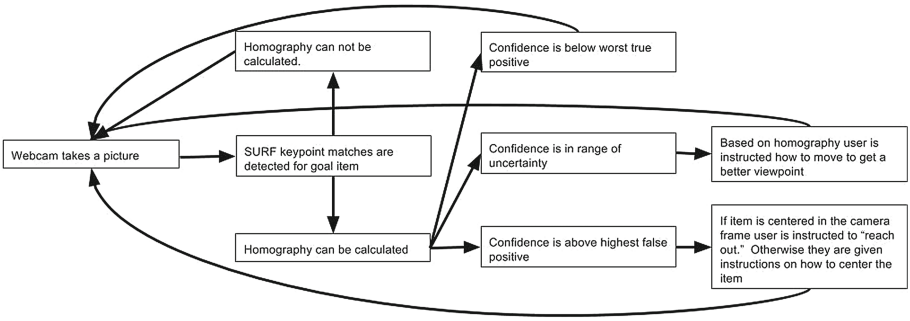

Fig. 3. Flow diagram of choices algorithm makes.

object is straight in front of their face and that they should reach out to grasp it. A simple flow diagram of this process is shown in Fig. 3.

Thus, our approach uses the cognitive abilities of the users (to understand instructions) and their mobility (to execute the suggested moves) to improve the quality of the view of an object as captured by the head mounted camera. Because our main application is for blind users, the system never needs to rely on any human visual ability.

Training is performed to find confidence thresholds for the top and bottom of the uncertainty range for each item. This range is defined as the values between the lowest true positive threshold, and the highest false positive threshold. The lowest true positive is the smallest confidence score ever given to an image that an item is present where the item being trained for was actually in the image. The highest false positive is the largest confidence ever given that an item was in a training image when it was not present. Training and testing incorporate the use of homography matrixes. A homography matrix is a representation of where the camera is relative to a set of points in space that all lie on the same plane. Homographies are calculated based on the relative positions of a set of points in relation to each other in a template image compared to their relative positions in a camera image. For example, if the points are all proportionally closer, the homography would show the camera is further away from the object than where it was when the template image was taken. Another example can be seen in Fig. 4. In our case, the template points are from the goal object for which the system is currently training. Because the points must be on the same plane, in the current instantiation of the system objects being found must be in boxes, as opposed to cans or other objects without a flat front surface. To calculate a homography a keypoint matching algorithm is required. These algorithms calculate feature descriptors in images and finds matches between similar descriptors in other images. Matches will include a match confidence as well as the pixel positions in both images, as needed for the homography calculation. We chose SURF [4], as opposed to others such as SFIT [15], because of it's speed. In the end system, homography matrices are the method used to give instructions. To train the lowest true positive threshold, objects are displayed to the camera and rotated

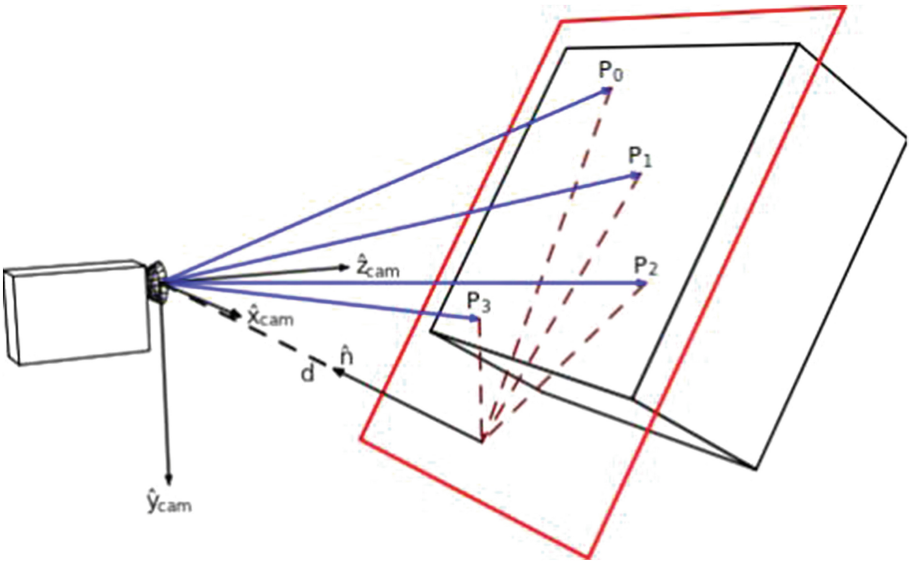

Fig. 4. Visual representation of Homography calculation [26]. In this case the points will be proportionally closer in the y-axis, while in the x-axis they will closer at the top and further apart at the bottom. The calculated homography would describe a position where the camera is looking up at the object from below.

in all directions and moved closer and farther away. The lowest true positive value is determined to be the lowest confidence value seen where a homography is still able to be calculated. If a homography cannot be calculated these confidences are not used because we would not be able to direct the user from those images. The highest false positive value is trained at the same time. While training the lowest true positive for one object, confidences are recorded for every other object in the database. The strongest confidence ever seen for each object, while actually looking at other objects, sets the thresholds for the highest false positives. To be safe, we additionally add 15 % of the range between thresholds to this value as a buffer. An example of scores relative to these thresholds is shown in Fig. 5.

During a run, if the confidence is within the uncertainty range the system outputs that it doesn't know the answer. However, it uses the information it has to arrive at a better decision later. If the confidence is between these thresholds, and a homography can be calculated, the system will know where the camera is relative to the points on the object used in the homography calculation. It can then pass on this information to the component that moves the camera. In our application that component is the human user. Using audio feedback our system tells the user how to move in order to guide the camera to a better viewing angle. If homographies continue to be calculated, eventually an ideal, front and centered, viewpoint can be achieved. Images from this camera angle generate the most similar keypoints to the template's keypoints, giving the highest confidences. If the confidence of an image

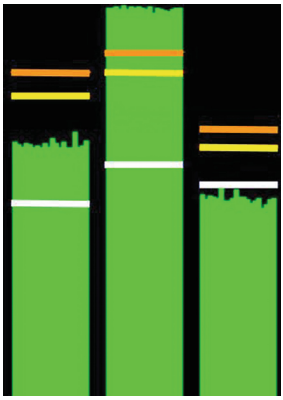

Fig. 5. Confidence threshold display for three items. Bars represent confidence values over last 20 frames. No units are shown because display confidences for each are relative percentage between max and min ever seen by the system when searching for each item. Middle threshold is the highest false positive. Bottom threshold is the lowest true positive. Top threshold is the extra buffer, 15 % above of the range above the highest false positive. Left confidence is in the range of uncertainty, if this was the item being searched for directions would be given. Middle confidence is above the max false positive value meaning this item is actually in the camera frame. Right confidence is below the lowest true positive so this item is certain to not have enough keypoint matches in the image to recover a homography.

surpasses the highest false positive value for the goal object correctness is certain. If, with an ideal viewpoint, the item is still not above this threshold the user knows to move on. This will happen when items have enough keypoints in common that a homography for the goal item is still able to be calculated from keypoints found on the alternate item. Most frequently this is seen between objects which share brand logos or other portions of similar visuals.

The physical system consists of three components. The first is the headset, created by attaching a webcam to a pair of glasses. The camera is attached directly in the middle to best capture images replicating where a person would be looking. The next is a pair of headphones to allow the user to hear the audio feedback. The last is the computer which performs SURF template matching, checks confidences, and gives instructions. We have used a GigaByte Brix which is able to be placed in a backpack and powered with a battery while the user is performing the task. These components are all controlled via SSH by a Samsung tablet.

4 Experiment Setup

For real world applications a computer vision system must be flawless, or close to flawless in identifying what it is being used for. A system saying it has found what it is looking for when it has not could range from catastrophic to just

inconvenient, but in any case it would not be widely used with more than a minute allowance for mistakes. The goal of our system is to show that this method of thresholding can achieve perfect accuracy. Blind grocery shopping tests this goal. The visually impaired user is able to use human decision making and movement for all parts of the task other than the actual vision. Grocery shopping is also a task where a system telling the user to purchase the wrong item even once would be considered a serious mistake.

Fig. 6. Experiment setup. Shown is a user confirming a selected item in the simulated grocery store.

4.1 Environment and Instructions

Our experiment took place in a simulated grocery store aisle using blindfolded participants as shown in Fig. 6. Subjects were 42 students. We arranged two bookshelves next to each other where our grocery store items could be placed. Each bookshelf has three shelves and we allowed items to be placed in three locations per shelf making 18 total locations items could be placed. During any given run five items would be out on the shelves at a time. These items came from one of five categories; cereal, snacks, pasta, tea, and candy. The five items arranged together during each trial would be from the same category. Users all performed 1 trial from each category with the same locations, and 2 trials from each category with randomized positions which were unique, for a total of 15 trials per subject. To begin each trial, the user would stand against the back wall of the room facing the items. At this point the system would be

turned on with the goal item selected. The user was instructed to move slowly around the room, while facing the shelves, until an initial movement command was given by the system. This would occur when SURF matches were made in arrangements where homographies could be calculated and the confidence was above the lower worst true positive threshold. Once an instruction was received the user was to follow the instructions which would guide them to be centered in front of the item. Instructions included "Left," "Right," "Up," "Down," "Left Up," "Left Down," "Right Up," "Right Down," "Strafe Left," "Strafe Right," "Strafe Up," "Strafe Down," "Step Forward," "Step Back," and "Reach Out." Examples of images which would elicit direction commands can be seen in Fig. 7. Direction commands were to move in those directions, strafe commands were to move in that direction but rotate the opposite direction, and Reach Out was the command which was only given when the object was directly centered and the confidence was above the worst false positive confidence plus buffer threshold.

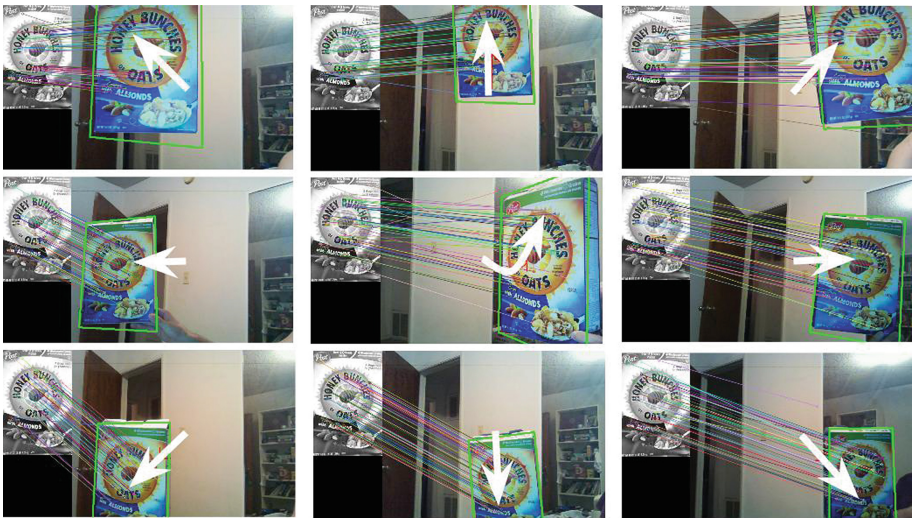

Fig. 7. Instructions correspond to camera's position based on homography calculation. User is guided to make camera point directly at center of object. Center image shows a strafe command where user would be instructed to rotate in addition to move.

When the "Reach Out" command was given users were to reach out, from the camera, and pick up the item in front of them. Once this item was grasped they turned 90 °, to be sure no other items from the shelves were in the background, and confirm the item by receiving a second "Reach Out" command. This would be done by holding the item up to the camera and moving the item based on the audio feedback, rather than moving themselves as was done with the item on the shelf. Sometimes users would be guided towards incorrect items when two items had similar enough features that confidence would be high when looking

at the wrong item and points matched in such a way that homographies could still be calculated. However, when centered on the incorrect item the worst false positive threshold would not be surpassed, and hence no "Reach Out" command would be issued. It would be up to the participant to decide to move on to other locations in the occasions where they had centered onto an object but were not being instructed to reach out.

4.2 Training

Each participant was first briefly trained on how to use the system. Training started with all 25 items out on the shelves. This would be more difficult than during non-training trials, where the items would be less crowded. Participants ran the experiment three times without a blindfold, and then three times with a blindfold to get a feel for the system. At that time the participant continued training until they successfully performed three trials in a row without making a mistake. Mistakes are defined in two ways. One was if they picked up the wrong item. Users knew not to pick anything up until they received the "Reach Out" command, but actually reaching out towards the location directly in front of the camera's center of field proved to not be an inherently easy function to perform. Some users reached slightly to the left or right, or even too high or too low to a different shelf. The second predetermined mistake to avoid was "losing" the item once tracking had begun. When the system was initially turned on, instructions typically were not received as the item to be searched for was either not in the camera frame, or far away and therefore too small in the image to get enough keypoint matches to calculate a homography. This was the "no instruction" condition. In this case the user was to scan the shelves without instructions until a first instruction was given. At this point the user followed instructions which would guide the object to the center of the camera frame. If the user moved in such a way that the object was lost from the camera frame and they were relapsed to the "no instruction" condition that would be considered a failure during training. During the actual experiment, trials would not be aborted whenever the subject "lost" an item, and users had to recover from it on their own. Likewise, if the user picked up an item but could not confirm it and decided they had reached incorrectly, the item would be returned to the shelf and the trial would continue. Failures during the experiment could hence occur if users both picked up and confirmed the wrong item, or if they gave up on a given trial (which never happened).

4.3 Control Experiment

Our control experiment was performed in a similar manner using a barcode scanner. This was chosen, rather than another computer vision system, because we were confident we could achieve perfect accuracy and wanted to test against a second option which would have perfect accuracy [21]. The barcode scanner used was an Amazon Dash. Experimental setup for these trials was kept as parallel as possible to a grocery store setup. The same two bookshelves were used, again

with 18 possible locations. As in modern grocery stores the barcodes were placed directly on the front face of the shelf. In these trials one barcode was selected at random as the goal. Users scanned every barcode in any order they chose until the correct one was scanned. No mistake conditions were defined for these trials. Training simply consisted of giving the user the scanner and a blindfold and they were allowed to practice indefinitely until they felt confident.

5 Experimental Results

Experiments were run as described on 42 participants. For all trials with all participants the correct item was always correctly obtained by the participant. Barcode scanning trials also were always successful. In each trial three time points were collected. The first was the time at which the first instruction was received. This cutoff was included because in some trials participants would take the majority of their total time moving blindly before receiving any instructions. With the barcode scanner, subjects would start trials with the scanner already held to the first barcode. The first scan would regularly take less than a second, so we wanted to have a cutoff for the first piece of feedback for our system. The second time recorded was the time of the first "Reach Out" command. At this point the system was 100 % sure the user has found the item they are looking for and it was directly in front of them. The final time recorded was the additional time needed for the user to actually pick up and confirm the item, a final step not taken during the barcode scanning trials.

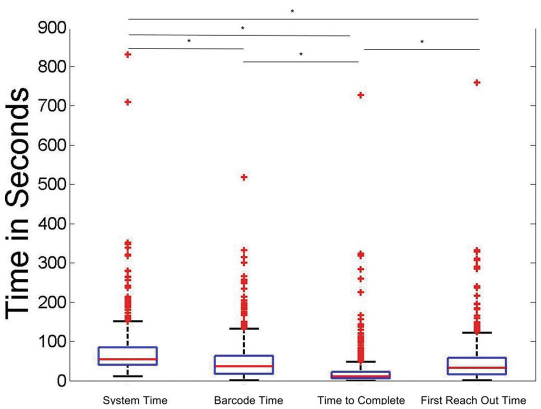

Fig. 8. Boxplots for total time taken for runs of the system and the barcode scanner, "time to complete" times for the system, and first "Reach Out" times for the system. Wilcoxon Rank Sum Tests were run on each pair to test if they could have come from continuous distributions with equal medians. All but Barcode Time vs First Reach Out Time had significant p-values: System Time vs Barcode Time: 1.5e-24, System Time vs Time to Complete: 6.2e-118, Barcode Time vs Time to Complete: 2.4e-46, System Time vs First Reach Out Time: 1.2e-34, First Reach Out Time vs Time to Complete: 7.4e-46, and Barcode Time vs First Reach Out Time: 0.18.

Mean total time for our system was 73.1 s per trial. Mean Barcode scanner time was 49.4 s. Using total time, this would mean the barcode scanner was distinctly faster. However, mean for first instruction time with our system was 23.7 s and for first "Reach Out" command was 46.5 s. This gives a mean "time to complete" with our system, time between first instruction and first "Reach Out" command, a mean time of only 18.1 s. We believe this is the time that should be compared, as further discussed in the next section. These results are shown in Fig. 8.

Surveyed participants were asked to assign a value 1–10 to their preference of systems with 10 being completely preferred our system, and 1 being completely preferred the barcode scanner. Mean score was 7.8 with only 2 participants reporting that they preferred the barcode scanner. Many reported their preference came from our system being able to provide more continuous feedback than a barcode scanner as guidance to the goal object. Of course, there could be some response bias of the subjects wanting to be "friendly" participants.

6 Discussion

The strongest algorithm from the most recent ImageNet Challenge [25] was developed by MSRA [12]. They achieved an accuracy rate of 62.07 % (as reported by [25]) over all object categories in the dataset, with a range of 95.93 % for the most accurate category and only 19.41 % for the weakest. This is still an outstanding result with the complexity of the ImageNet dataset, and impressive work with deep residual neural networks to achieve it. However, this rate of accuracy would be far too low for any real world applications where mistakes are costly. In situations such as assistance to blind grocery shoppers it is essential to not make mistakes. In earlier instantiations of our algorithm "Wrong Item" was also an instruction. It was given when an object was centered but the worst false positive threshold was not surpassed. The intention was to inform users they had centered on an item with similar enough keypoints to calculate homographies for the goal item, even though it was not the goal item itself. However, in the cases when this happened when they were actually looking at the goal item, only because one frame didn't calculate good keypoints, users would typically move away immediately. This choice sometimes added minutes more to their time before coming back to the correct item. This is why we decided to instead give no instruction when the item was centered but the threshold was not surpassed, and rely on the participant to decide on their own when they had centered on an incorrect item. As seen in the ROC curves earlier, with a fixed threshold a SURF based algorithm could perform with reasonable error rates on all of the items in our dataset. However, when even a single bad instruction from a single frame can increase your time significantly, and the algorithm is running at many frames per second, perfect accuracy is necessary for an algorithm to be optimal. Our experiment has shown that using this human-in-the-loop system 100 % accuracy is, in fact, possible with a computer vision based system in a real life application. Using a human's mobility and decision making allows the

algorithm to not have a fixed threshold and instead postpone decisions when uncertain. Without forcing answers from uncertain conditions the algorithm is able to never make mistakes.

Time for the barcode scanner was stopped when the user scanned the correct barcode. These trials did not require the user to pick up an actual item or confirm it. Removing the time to pick up and confirm with our system makes the two more equivalent. The time before first command is also not parallel for the barcode scanner. In barcode scanning trials the subjects were allowed to start with the scanner already held up to the first barcode of their choice. This often meant the first piece of feedback would be immediate. In trials for our system the time taken before the first command was received was regularly a large majority of the total time taken. A major cause of this was the choice of webcam for our original system. With a low definition webcam, the smaller items would sometimes require users to have to get within a couple feet from the item before they took up a large enough portion of the image to detect any keypoints. This meant the subject might have to blindly scan all 18 positions before getting any feedback whatsoever. Sometimes they would even have to do this more than once if they did not scan correctly the first time. With an HD webcam the user should be able to scan all 18 positions on both bookshelves at once from the starting position at the back wall. This would eliminate all time taken before first command.

As evidence for this, for the larger items in the cereal category this was already the case. With such large items initial instructions were often heard immediately. Considering only this category, mean total time was 57.1 s. However, for cereals first instruction time had a mean of 9.1 s compared to 27.5 for the other categories. With a mean time of 34.6 s to pick up and confirm an item after receiving the first "Reach Out" command this gave cereals a mean "time to complete" time of only 13.5 s and a mean time from start to the first "Reach Out" command of 22.5 s. Either of these times are more comparable to the barcode scanner times, since barcode trials did not require confirmation and started feedback immediately, and both are faster.

Compared to a barcode scanner the total times for our system were slower. However, when only considering "time to complete," the time needed for the subject to center the correct item in the camera frame after receiving their first instruction, our system was faster. Also considering only time to first "Reach Out" command, ignoring time taken to grasp and confirm the item not necessary in barcode scanner trials, times did not show significant difference. Importantly, surveyed participants reported they preferred the constant guided feedback of our system against the yes/no feedback the barcode scanner could provide, even in our reduced store with only two shelves. We hence conclude that this study has successfully demonstrated a user-in-the-loop machine vision algorithm that made no mistakes and could be an interesting basis for a new generation of visual aids.

Acknowledgment. This work was supported by the National Science Foundation (grant numbers CCF-1317433 and CNS-1545089), and the Office of Naval Research (N00014-13-1-0563). The authors affirm that the views expressed herein are solely their

own, and do not represent the views of the United States government or any agency thereof.

References

1. Adebiyi, A., Mante, N., Zhang, C., Sahin, F.E., Medioni, G.G., Tanguay, A.R., Weiland, J.D.: Evaluation of feedback mechanisms for wearable visual aids. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. IEEE (2013)
2. Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. Int. J. Comput. Vis. **1**(4), 333–356 (1988)
3. Bagdanov, A.D., Del Bimbo, A., Nunziati, W.: Improving evidential quality of surveillance imagery through active face tracking. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 1200–1203. IEEE (2006)
4. Bay, H., Tuytelaars, T., Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). doi:10.1007/11744023_32
5. Bjorkman, M., Eklundh, J.O.: Vision in the real world: finding, attending and recognizing objects. Int. J. Imaging Syst. Technol. **16**(5), 189–208 (2006)
6. Bowen III, C.L., Buennemeyer, T.K., Burbey, I., Joshi, V.: Using wireless networks to assist navigation for individuals with disabilities. In: California State University, Northridge Center on Disabilities' 21st Annual International Technology and Persons with Disabilities Conference (2006)
7. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
8. Dramas, F., Thorpe, S.J., Jouffrais, C.: Artificial vision for the blind: a bio-inspired algorithm for objects and obstacles detection. Int. J. Image Graph. **10**(04), 531–544 (2010)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2241–2248. IEEE (2010)
10. Fok, D., Polgar, J.M., Shaw, L., Jutai, J.W.: Low vision assistive technology device usage and importance in daily occupations. Work **39**(1), 37–48 (2011)
11. Gratal, X., Romero, J., Bohg, J., Kragic, D.: Visual servoing on unknown objects. Mechatronics **22**(4), 423–435 (2012)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
13. Hub, A., Hartter, T., Ertl, T.: Interactive tracking of movable objects for the blind on the basis of environment models and perception-oriented object recognition methods. In: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 111–118. ACM (2006)
14. Jiang, X., Mojon, D.: Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. IEEE Trans. Pattern Anal. Mach. Intell. **25**(1), 131–137 (2003)
15. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, p. II-506. IEEE (2004)

16. Kulykukin, V., Gharpure, C., DeGraw, N.: Human-robot interaction in a robotic guide for the visually impaired. In: AAAI Spring Symposium, pp. 158–164 (2004)
17. Kulyukin, V., Gharpure, C., Nicholson, J.: Robocart: toward robot-assisted navigation of grocery stores by the visually impaired. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2845–2850. IEEE (2005)
18. Merler, M., Galleguillos, C., Belongie, S.: Recognizing groceries in situ using in vitro training data. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
19. Na'aman, E., Shashua, A., Wexler, Y.: User wearable visual assistance system, 23 August 2012, uS Patent Ap. 13/397,919
20. Nanayakkara, S., Shilkrot, R., Maes, P.: Eyering: a finger-worn assistant. In: CHI 2012 Extended Abstracts on Human Factors in Computing Systems, pp. 1961–1966. ACM (2012)
21. Nicholson, J., Kulyukin, V., Coster, D.: Shoptalk: independent blind shopping through verbal route directions and barcode scans. Open Rehabil. J. **2**(1), 11–23 (2009)
22. O'Gorman, L.: Binarization and multithresholding of document images using connectivity. CVGIP. Graph. Models Image Process. **56**(6), 494–506 (1994)
23. Passini, R., Proulx, G.: Wayfinding without vision an experiment with congenitally totally blind people. Environ. Beh. **20**(2), 227–252 (1988)
24. Phillips, B., Zhao, H.: Predictors of assistive technology abandonment. Assistive Technol. **5**(1), 36–45 (1993)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
26. Scientist 47: Homography-transl. In: Wikimedia.org (2008)
27. Striem-Amit, E., Guendelman, M., Amedi, A.: 'Visual' acuity of the congenitally blind using visual-to-auditory sensory substitution. PloS One **7**(3), e33136 (2012)
28. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
29. Thakoor, K.A., Marat, S., Nasiatka, P.J., McIntosh, B.P., Sahin, F.E., Tanguay, A.R., Weiland, J.D., Itti, L.: Attention biased speeded up robust features (ab-surf): a neurally-inspired object recognition algorithm for a wearable aid for the visually-impaired. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. IEEE (2013)
30. Ude, A., Gaskett, C., Cheng, G.: Foveated vision systems with two cameras per eye. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 3457–3462. IEEE (2006)
31. Ultracane: Ultracane: Putting the world at your fingertips (2016). http://www.ultracane.com/about_the_ultracane
32. WHO: World health organization fact sheet. WHO N°282 (2014)
33. Zhao, X., Ong, S.: Adaptive local thresholding with fuzzy-validity-guided spatial partitioning. In: Proceedings of the Fourteenth International Conference on Pattern Recognition, vol. 2, pp. 988–990. IEEE (1998)