Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio

Rui (Forest) Jiang
Earth Science, Stanford
forestj@stanford.edu

Qian Lin
Applied Physics, Stanford
linqian@stanford.edu

Shuhui Qu
Civil and Environmental Engineering, Stanford
shuhuiq@stanford.edu

Abstract

This project tries to transform the visual world into the audio world with the potential to inform blind people objects as well as their spatial locations. Objects detected from the scene are represented by their names and converted to speech. Their spatial locations are encoded into the 2-channel audio with the help of 3D binaural sound simulation.

Our system composes of several modules. Video is captured with a portable camera device (Ricoh Theta S, Microsoft Kinect, or GoPro) on the client side, and is streamed to the server for real-time image recognition with existing object detection models (YOLO). The 3D location of the objects is estimated from the location and the size of the bounding boxes from the detection algorithm. Then, a 3D sound generation application based on Unity game engine renders the binaural sound with locations encoded. The sound is transmitted to the user with wireless earphones. Sound is play at an interval of few seconds, or when the recognized object differs from previous one, whichever earliest.

The prototype device is tested in a situation simulating a blind people being exposed to a new environment. With the help of the device, the user successfully found a chair that is 3-5 meters away, walk towards it and sit on it. Issues about current prototype have been identified as: detection failure when objects are too close or too far, and overload of information when the system tries to notify users too many objects.

The project demonstration can be found in YouTube https://youtu.be/s_-gAVTJl18.

1. Introduction

Millions of people live in this world with incapacities of understanding the environment due to visual impairment. Although they can develop alternative approaches to deal with daily routines, they also suffers from certain navigation difficulties as well as social awkwardness. For example, it is very difficult for them to find a particular room in an unfamiliar environment. And blind and visually impaired people find it difficult to know whether a person is talking to them or someone else during a conversation.

Computer vision technologies, especially the deep convolutional neural network, have been rapidly developed in recent years. It is promising to use the state-of-art computer vision techniques to help people with vision loss.

In this project, we want to explore the possibility of using the hearing sense to understand visual objects. The sense of sight and hearing sense share a striking similarity: both visual object and audio sound can be spatially localized. It is not often realized by many people that we are capable at identifying the spatial location of a sound source just by hearing it with two ears.

In our project, we build a real-time object detection and position estimation pipeline, with the goal of informing the user about surrounding object and their spatial position using binaural sound. Section 2 discuss the relate works on sensory substitution, assistive products using computer vision for blind people, and the exploration of 3D sound. Section 3 introduces different components of our prototype. The testing and result discussions are in Section 4. Then the report concludes with Section 5.

2. Related Work

There exists multiple tools to use computer vision technologies to assist blind people.

The mobile app TapTapSee uses computer vision and crowdsourcing to describe a picture captured by blind users in about 10 seconds. The Blindsight offers a mobile app Text Detective featuring optical character recognition (OCR) technology to detect and read text from pictures captured from the camera. Facebook is developing image captioning technology to help blind users engaging in conversations with other users about pictures. Baidu recently released a demo video of a DuLight project. No further details of the product is available at the moment. However, the product video suggests concepts of describing scenes and recognizing people, money bills, merchandises, and crosswalk signal. However, these products were not focusing on enabling general visual sense for blind people and did not use the spatial sound techniques to further enhance the user experience.

Some works exist in the general scope of sensory substitution. Daniel Kish, who are totally blind, developed accurate echolocation ability using “mouth clicks” for navigation tasks including biking and hiking independently . Colorblind artist Neil Harbisson developed a device to transform color information into sound frequencies. An extreme attempt of converting visual sense to sound is introduced by the vOICe technology [6]. The vOICe system scans each camera snapshot from left to right, while associating height with pitch and brightness with loudness. However, all these attempts on sensory substitution are reported with very difficult learning process. In contrast, we utilize visual recognition algorithms which lead to more direct ways of understanding objects from a visual scene.

The use of 3D sound technology for providing useful information and assisting blind people has also been investigated by researchers. [7] introduced a system that uses spatial audio to facilitate discovery of points of interest in large, unfamiliar indoor environments (e.g. shopping mall). [8] tries to integrate 3D sound into GPS-based outdoor navigation product. However, no visual recognition has been used in those works. The use of object detection techniques can open up new possibilities in assisting indoor navigation for blind and visually impaired people.

3. Methods

3.1. Object detection algorithm

To successfully detect surrounding objects, we investigate several existing detection systems that could classify objects and evaluate it at various locations in an image. Deformable Parts Model (DPM) [10] uses root filters that slides detection windows over the entire image. R-CNN [11] uses region proposal methods to generate possible bounding boxes in an image. Then, it applies various ConvNets to classify each box. The results are then post-processed and output finer boxes. The slow test-time, complex training pipeline and the large storage does not fit into our application. Fast R-CNN [12] max-pools proposed regions and combines the computation of ConvNet for each proposal of an image and outputs features of all regions at once. Based on Fast R-CNN, Faster R-CNN [13] inserts a region proposal network after the last layer of ConvNet. Both methods speed up the computational time and improve the accuracy. The pipelines of these methods are still relatively complex and hard to optimize. Considering the requirement of real-time objective detection, in this project, we use You Only Look Once (YOLO) model [9]. YOLO could efficiently provides relatively good objective detection with extremely fast speed.

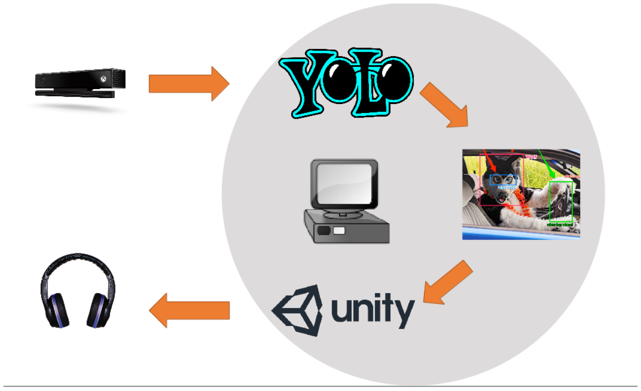

Figure 1. Data flow pipeline of our system.

3.1.1 YOLO Model

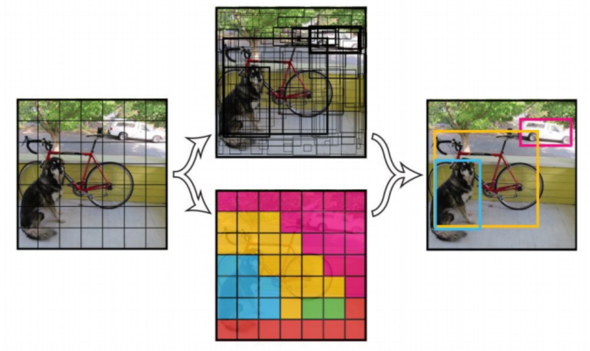

Figure 2. The YOLO Model.

Instead of using region proposal method, YOLO model divides an image into S×S grid. Each grid cell predicts B bounding boxes, and boxes’ confidence scores for the prediction and detect if a class falls in the boxes. The confidence is defined as $Pr(object) \times IOU^{truth}_{pred}$, which represents the confidence of a class in the box and accuracy of the box coordinates. Thus, each box has five parameters to predict: x, y, w, h and confidence. Each grid cell also predicts $Pr(Class_i|Object)$. Thus the confidence for each box is $Pr(Class_i|Object) \times Pr(object) \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth}$. The overall variables to be predicted can be represented as a $S \times S \times (B \times 5 + C)$ tensor.

3.1.2 YOLO Model ConvNet

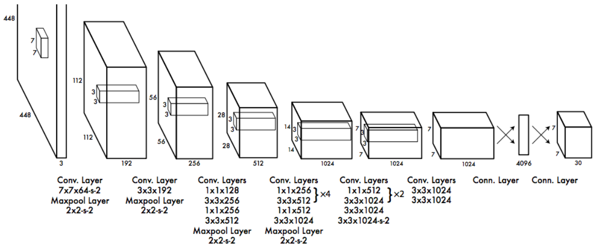

Figure 3. Convolutional neural network of the YOLO Model.

The ConvNet architecture is shown in Figure 3. The network has 24 convolutional layers with 2 fully connected layers. The ConvNet is to extract features from input images and the fully connected layers are to predict the probability of the boxes coordinates and confidence score. The accuracies of the predictions also depend on the architecture of the network. The loss function of the final output depends on the x, y, w, h, prediction of classes and overall probabilities. In our project, we use pretrained YOLO weight to detect objects.

3.2. Depth estimation

After detecting the type of objects in a video frame, the next step is to obtain the depth or distance of the detected object from the user. We make two separate attempts to this problem.

In our initial attempt, we use a Microsoft Kinect as the video camera device in our pipeline. Kinect has the benefit of capturing real-time depth map together with the RGB image. After detecting the object in the RGB image and its corresponding bounding box, we can simply use the average depth of the bounding box area as the distance. The disadvantage is that Kinect is very bulky as a personal carry-on camera, and that it's difficult to work in a wireless mode, thus limiting the traveling range of the user. There exist other more portable depth cameras, for example the Zed camera with relies on depth estimation from stereo vision. However, such camera generally requires high computation resource (GPU) to estimate depth from stereo image.

To overcome the difficulty of integration of depth camera into our project pipeline, we revisit the user need for depth information. First of all, human are good at inferring direction from binaural sound, and the relative distance, namely object A is closer than object B or object is moving closer and closer between frames. However, absolute distance is difficult to deduce from binaural sound. This means our image processing algorithm needs to provide the accurate directional information and the relative distance, but not the exact depth.

Thus we resort to estimate the direction and relative depth from an RGB image. We choose to use GoPro Hero 3 since it's a very light-weight carry on camera with large field of view, high frame rate and wireless compatibility. Giving the field of view of the camera, and the bounding box of the object, the direction can be estimated from the central pixel location of the bounding box. For the estimated depth, we assume a "default" height for any particular class, for example human is assumed to be around 5.5 feet, and chairs are assume to be 2.5 feet. We hard code this for each of the 20 classes in our classifier. Then from the height of the bounding box and the default height of the object we can estimate the depth.

3.3. Data streaming

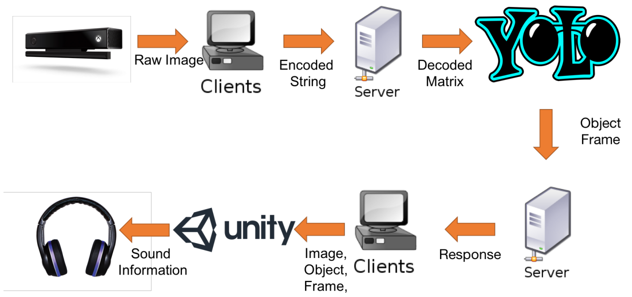

Figure 4. Initial Data flow pipeline.

Our project is based on a platform that is capable of processing real-time image. Thus, it is required to have a powerful GPU that could give feedback in no time. Considering the computational cost and performance, we initially use rye machine provided by Stanford as our prototype's server machine. A pipeline is developed that enables us to communicate quickly. As Figure 4 shows, a program in local machine extracts raw image from a camera (e.g. Kinect), encodes it into a string and sends through a client to a server running on the Stanford Rye machine. The server decodes it and use trained object detection engine to return detected items. The server then sends that information back to the client, which triggers the Unity-based stereo generator to play the 3D sound. During the implementation, we find that the communication between our personal computer and Rye machine takes a few milliseconds to transfer each video frame. Also the performance of the Rye machine is not stable due to the mass occupancy of GPU.

Based on our initial platform, we switch to local platform that is more efficient. The architecture is shown in Figure 5. In this platform, the evnironment is captured by a portable camera and transfers through HD video link directly to the

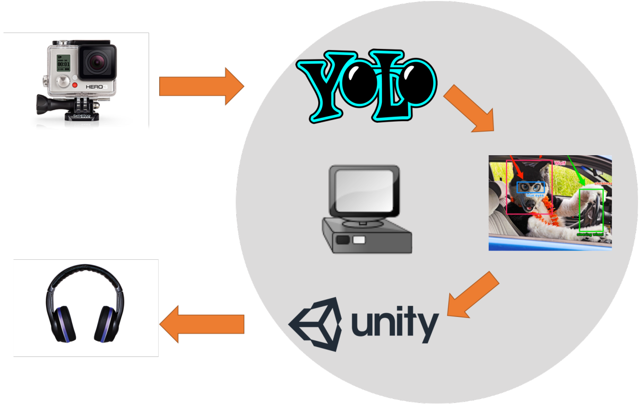

Figure 5. Current Data flow pipeline of our system.

YOLO model running on a local server machine with high performance GPU. The server detects objects, sends information directly to the unity sound generator and plays the binaural sound.

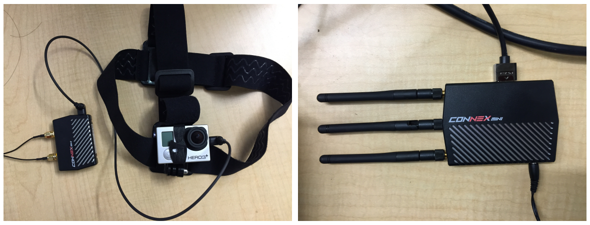

Figure 6. Camera device and image streaming device.

In particular, the environment picture is captured by a portable GoPro Hero3 at 30 frames $1080p$ resolution. The video is live streamed through the HD video link to the computer server as shown in Figure 6. The HD video link could transfer a high resolution image within 2 miles in 1 millisecond. The object detection engine YOLO then predicts objects in the stream. The YOLO algorithm could process a single image frame at a speed of 4-60 frames/second depending on the image size we send to the engine. The outputs are sent to unity sound generator and the generated sounds are played through wireless earbuds shown in Figure 9. During the implementation, the platform is capable of processing all captured live stream at a minimum speed of 30 frames per second at 1080p resolution.

Figure 7. Unity program for generating 3D sound and device to transmit the audio signal to the user.

3.4. Result filtering

YOLO outputs the top classes and their probability for each frame. We take any probability above 20% as a confident detection result.

To present the results to the user in a reasonable manner, our algorithm also has to decide whether to speak out a detected object and at what time. Obviously it's undesirable to keep speaking out the same object to the user even if the detection result is correct. It's also undesirable if two object names are spoken overlapping or very closely that the user won't be able to distinguish.

To solve the first problem, we assume a cool-down-time of five seconds for each class. For example, if a person is detected in the first frame and is spoken out, the program will not speak out "person" again until after five seconds. This is only an sub-optimal solution since it does not deal with multiple objects of the same class. Ideally, if there are two persons in the frame, the user should be informed about the two person, but he does not need to be informed about the same person continuously. One possible improvement, which we are still working on, is to track the object using overlapping bounding box between frames. To solve the second problem, we plan to enforce a delay of half a second between any spoken classes.

Among all 20 classes of the existing YOLO model, we choose the following classes to inform the user: "bottle", "chair", "diningtable", "person", "pottedplant", "sofa", "tv-monitor".

3.5. 3D sound generation

We use a plug-in for Unity 3D game engine called 3DCeption to simulate the 3D sound. We developed a Unity-based game program "3D Sound Generator" using either a file watcher or TCP socket to receive the information about the correct sound clips to be played as well as their spatial coordinates. Then, 3DCeption renders the binaural sound effect with the help of the Head-Related Transfer Function (HRTF) to simulate the reflection of the sound on human body (head, ear, etc.) and obstacles (such as wall and floor).

As most of the sighted people may be not aware of the sound localization capability, the reader is recommended to experience the 3D binaural sound effect demonstrations (e.g. 3Dception Realtime 3D Audio Demo and Virtual Barber Shop) on YouTube.

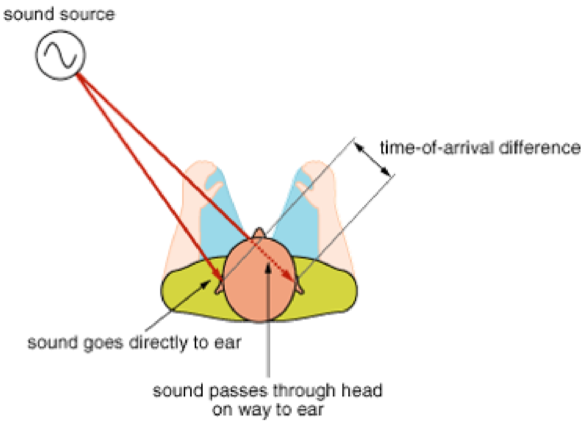
sound source
time-of-arrival difference
sound goes directly to ear
sound passes through head
on way to ear

Figure 8. Principle of 3D sound.

Figure 9. Unity program for generating 3D sound and device to transmit the audio signal to the user.

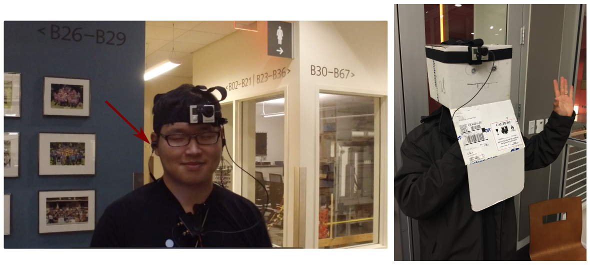

Figure 10. Testing setup.

4. Testing and Discussion

4.1. Testing

We first try our system on ourselves with eyes covered by a paper box, as shown in Figure 10. The objective is to simulate the situation of a blind person just exposed with a new environment. Wearing the device, the user correctly identified objects such as persons, and chairs at the indoor range.

Figure 11 shows a situation where an user entered a small room, with common objects like chair, table, and monitor, and intended to find a bottle. The user was notified by the system the type and spatial location of the objects, and successfully located the bottle and grabed it.

Figure 12 shows an example where a "blind" person entered a lobby and successfully identified a chair, walked and sat on it. With existing device prototype, the user on average takes about 15 seconds to identify and walk to the chair.

In both situations, the user immediately identified the spatial locations of objects just by hearing the 3D sound notification and reported a sense of augmented reality, and a feeling of "that person is right there."

We also let a blind person try the device. Showed positive feedback along with some suggestions of improvement (see discussion below.)

4.2. Discussion

The prototype we build successfully recognizes visual objects and presents the detection information as 3D sound, giving the user a sense of "augmented reality". However, the prototype suffers from the following limitations.

First, it is common for user to focus on certain object from afar and navigate to a location close to the object. In this task, the user need a consistent instruction of the target object from approximately 10 m away to only 20 cm away. That impose a very high requirement to the object detection model. To our experience, YOLO can correctly detect objects, such as chair, within a range about 2-5 m away. Objects that are outside this range are either unrecognized or misclassified. One approach to solve this issue is to incorporate training images with greater scale ranges (e.g., include chair picture captured from 20 cm away and 10 m away). However, it may be difficult for object detection models to classify the object from a picture of extreme scale (too close or too far). Another approach to solve this is to use object tracking algorithm to track the object (e.g. a chair) once the user have identified as the target. These two approaches are worth exploring in the future work.

The second issue reported by the blind user is the blocking of ambient sound by using earbuds. However, this can be solved by using bone conduction earphones, which leave ears open for hearing surrounding sounds.

The third issue reported by the blind user is "information overload" when the system is trying to notify user of multiple objects at the same time. This can be solved by delayed notifications. For example, the system can sequentially notify the user of the object from left to right. However this solution requires the user stands still while playing the 3D sounds. Moreover, blind people usually do not want to know every objects in his "eyesight", but instead want to know objects that are pertinent to their immediate need. For example, they may want to find a particular room in a building, or find food and drinks during a conference. In this regard, the system should have three modes: exploration mode where users are notified with every detected objects, search mode where the system only notify users of the object they are looking for, and navigation mode where only the target object and obstacle objects are notified to users in real time.

In sum, extensive work is required to analyze users' need if one would like to stem from this prototype to a really helpful assistive product.

5. Conclusion and Future Work

In this project, we investigate the need from blind and visually impaired people. Base on the impetus of the CNN, we develop a blind visualization system that helps blind people better explore the surrounding environment. A portable and real time solution is provided in the project. We present a platform that utilizes portable cameras, fast HD video link and powerful server to generate 3D sounds. By using YOLO algorithm and advanced wireless transmitter, the solution could perform accurate real time objective detection with live stream at a speed of 30 frames, 1080P resolution. A prototype for sensory substitution (vision to hearing) is established in the project. Through this project, we hope to demonstrate the possibility of using computer vision techniques as a type of assistive technology.

The project demonstration can be found in YouTube, see https://youtu.be/s_-gAVTJl18.

References

[1] Joseph Redmon and Anelia Angelova, Real-Time Grasp Detection Using Convolutional Neural Networks (ICRA), 2015.

[2] A. Quattoni, and A.Torralba. Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[3] Saurabh Gupta, Ross Girshick, Pablo Arbelaez and Jitendra Malik, Learning Rich Features from RGB-D Images for Object Detection and Segmentation (ECCV), 2014.

[4] Tadas Naltrusaitis, Peter Robison, and Louis-Philippe Morency, 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking (CVPR), 2012.

[5] Andrej Karpathy and Fei-Fei Li, Deep Visual-Semantic Alignments for Generating Image Descriptions (CVPR), 2015.

[6] David Brown, Tom Macpherson, and Jamie Ward, Seeing with sound? exploring different characteristics of a visual-to-auditory sensory substitution device. *Perception*, 40(9):1120–1135, 2011.

[7] Liam Betsworth, Nitendra Rajput, Saurabh Srivastava, and Matt Jones. Audvert: Using spatial audio to gain a sense of place. In *Human-Computer Interaction–INTERACT 2013*, pages 455–462. Springer, 2013.

[8] Jizhong Xiao, Kevin Ramdath, Manor Iosilevish, Dharmdeo Sigh, and Anastasis Tsakas. A low cost outdoor assistive navigation system for blind people. In *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*, pages 828–833. IEEE, 2013.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

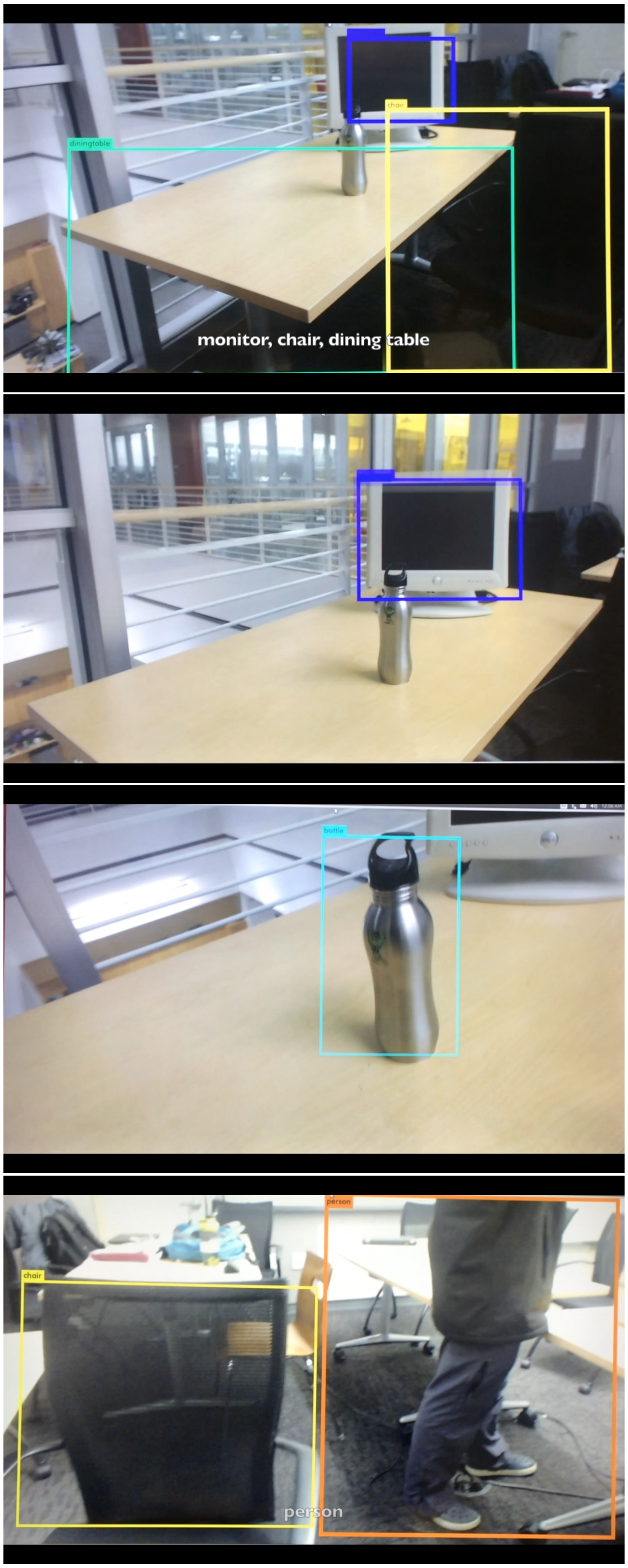

Figure 11. In this test scenario, the user is entering a small room, with common place objects like chair, table, monitor. He is trying to grab the water bottle on the table.

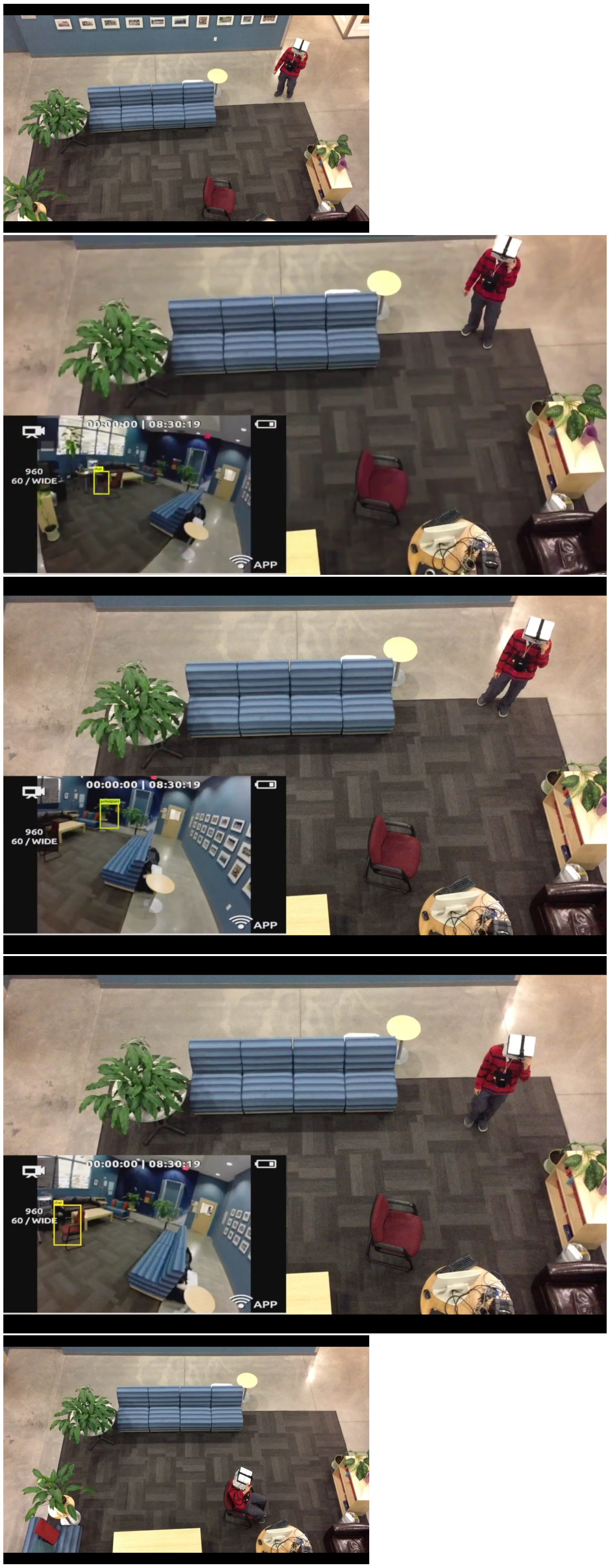

Figure 12. In this test scenario, the user is entering an unfamiliar open space and try to find a place to side down. He also wish to get informed of objects around him and their relative position.