AUGMENTED ANNOTATIONS: INDOOR DATASET GENERATION WITH AUGMENTED REALITY

Vedant Saran*, James Lin*, Avideh Zakhor

University of California, Berkeley (vedantsaran, james97lin, avz)@berkeley.edu

KEY WORDS: augmented reality, machine learning, real time 3D annotation, data augmentation, 3D labeling, 3D object detection, mapping, visualization

ABSTRACT:

The proliferation of machine learning applied to 3D computer vision tasks such as object detection has heightened the need for large, high-quality datasets of labeled 3D scans for training and testing purposes. Current methods of producing these datasets require first scanning the environment, then transferring the resulting point cloud or mesh to a separate tool for it to be annotated with semantic information, both of which are time consuming processes. In this paper, we introduce *Augmented Annotations*, a novel approach to bounding box data annotation that solves the scanning and annotation processes of an environment in parallel. Leveraging knowledge of the user's position in 3D space during scanning, we use augmented reality (AR) to place persistent digital annotations directly on top of indoor real world objects. We test our system with seven human subjects, and demonstrate that this approach can produce annotated 3D data faster than the state-of-the-art. Additionally, we show that Augmented Annotations can also be adapted to automatically produce 2D labeled image data from many viewpoints, a much needed augmentation technique for 2D object detection and recognition. Finally, we release our work to the public as an open-source iPad application designed for efficient 3D data collection.

1. INTRODUCTION

Access to human-labeled data is a necessary component in training supervised models on computer vision tasks. As the performance and robustness of these models increase, so do their demands for greater amounts of training data. It's imperative that methods for capturing and producing this data continue to evolve and improve alongside the algorithms that use them, lest researchers run into the problem of having innovative ideas but not enough data to evaluate them properly.

In the 2D domain, image data is commonplace. Smartphones with high quality cameras are ubiquitous in many parts of the world, and as a result huge numbers of pictures and videos are taken every day. Accompanying this are social media platforms such as Instagram [1] or image hosting sites such as Imgur [2], which make it easy to aggregate these images and take advantage of their existing metadata. As an example, Hays and Efros took a million pre-tagged pictures from Flickr to build a scene completion solution for arbitrary images (Hays and Efros, 2007) - no manual data collection or annotation was necessary.

In the 3D domain, however, properly annotated data remains relatively scarce. 3D information is physically more difficult and expensive to capture. Depending on the format of representation, the sparse nature of most 3D environments causes the data to take up significantly more storage and can be harder to process. Finally, there is little incentive for consumers or industry to capture and annotate such data; data repositories such as Thingiverse[3] and GrabCAD[4] exist, but are much less popular than their 2D counterparts, and focus more on 3D modeled scenes rather than of captures of real-life 3D environments. As a result, many publications rely on datasets that were generated for the sole purpose of research. In this paper, we address one of the bottlenecks limiting the availability of 3D data: the time consuming process of 3D data annotation for the purposes of training and testing.

Current state-of-the-art methods for producing 3D datasets adopt a two-step approach. First, the environment is scanned, often through some sort of tripod-mounted or handheld depth camera system. Next, the scans are uploaded to a server and accessed through a program or web app designed for data annotation. Users, using a mouse and keyboard, manually draw bounding boxes and apply text labels to objects in the scans through this interface. This is inefficient, as it requires two detailed passes over the same environment, once for scanning and once for annotating, whereby both steps of the process can be laborious and time consuming. It would be ideal to develop a procedure that eliminated this redundancy.

We point out two key insights that guide our solution to this problem. The first is that real-time Simultaneous Localization and Mapping (SLAM) algorithms have become accurate enough to play a role in generating ground-truth data. Under normal usage conditions, the error introduced by drift or other factors in many modern SLAM implementations is second-order compared to the error from human variance in annotating ground-truth data. This opens up the possibility for the user to interactively annotate the 3D environment in real-time, while the scanning is still in progress. The second insight is that depth cameras have become more accessible in recent years; newer models are cheap and compatible with smartphones, allowing us to take advantage of well-established mobile UI paradigms when designing tools.

In this paper we present *Augmented Annotations*, an iOS application that uses a depth sensor to consolidate the scanning and annotation processes for indoor scenes. Our application outputs high quality meshes of the environment alongside a list of labeled bounding boxes surrounding objects of interest. Users of our app use an iPad to scan the environment while simultaneously placing virtual bounding boxes that are localized relative to the real world. We show that through our method, users can pro-

*Contributed equally
[1] https://www.instagram.com/
[2] https://imgur.com/
[3] https://www.thingiverse.com/
[4] https://grabcad.com/

duce fully-annotated data at a faster rate than through traditional methods. We also show that the same procedure can be used to quickly capture annotated 2D images as well.

The outline of this paper is as follows. In Section 2, we review related works; Section 3 includes the workflow and user interface. Section 4 covers experimental results and Section 5 is conclusions and future work.

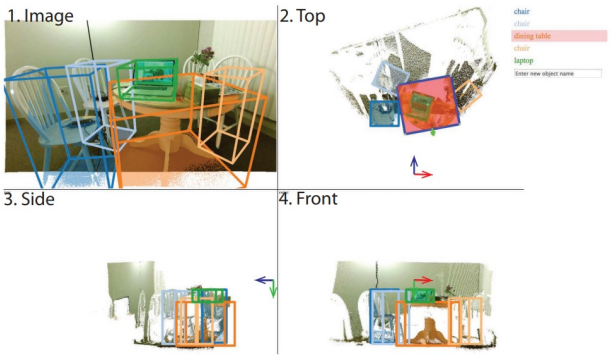

Figure 1: Screenshot from SUN RGB-D's annotation tool. Taken from Song et al.'s supplemental material (Song et al., 2015).

2. RELATED WORKS

Existing methods for annotating datasets rely primarily on the desktop computer. Russell et al. introduce a feature-rich program for annotating images called LabelMe, which allow users to draw polygons and query large-scale databases of images (Russell et al., 2008). Their work has inspired many others research groups to build their own annotation tools in a similar style. As annotation is a parallelizable task and requires relatively little training, these tools are also commonly made to work with crowd-sourcing platforms such as Mechanical Turk (Strickland and Stoops, 2018) or oDesk (Wenkart, 2014), which provide a way for researchers to connect to and distribute needed data to potential workers for an affordable cost.

In the 3D computer vision field specifically, one popular dataset is Silberman et al.'s NYUv2, a collection of 1449 RGB-D images taken of indoor scenes (Couprie et al., 2013). Their annotation is done in the 2D domain in that each image has a per-pixel labeling done through Mechanical Turk, allowing them to to take advantage of more well-developed 2D annotation procedures. Another popular dataset is SUN RGB-D, which contains 10,355 RGB-D images (Song et al., 2015). Unlike NYUv2, SUN RGB-D provides 3D bounding box annotation, also collected through a custom-made application on Mechanical Turk. Song et al.'s annotation tool, shown in Figure 1, presents scanned 3D scenes to each worker from various orthogonal perspectives. These workers then follow a procedure to create, modify, and label bounding boxes for each object of interest. The more recent ScanNet dataset uses a similar system to produce semantic segmentations on 3D scans (Dai et al., 2017). These tools are streamlined to minimize the amount of training required.

Augmented Reality (AR) as an interaction paradigm is still in its nascent stages, but nevertheless presents interesting implications for the computer vision community. Most new mobile devices have either multiple cameras or a depth sensor that allow them to perform SLAM. Industry developer SDKs such as 6D.AI (6D Development Team, 2017) and Placenote (Placenote Development Team, 2017) use AR to support persistent annotations for human consumption, allowing users to leave virtual, localized notes for for others or themselves, among other use cases. These notes remember their location in the physical world, even as the device itself moves. This style of interaction scheme naturally lends itself towards the goal of developing annotation tools.

There are cases where AR has been used to help further the computer vision field. For example, Alhaija et al. uses AR to generate realistic urban driving datasets (Abu Alhaija et al., 2018). They take real scenes of urban environments and augment them with virtual models of cars and other objects, thus producing endless variants of data from a much smaller library. That said, AR is still relatively new and we see great potential for further exploration on the subject.

Figure 2: Occipital Structure Sensor attached to an iPad. Image taken from (Hoffman, 2014).

Figure 3: Screenshot of the application in action: A scanned mesh, shown in white, is superimposed on top of camera feed in real-time. A bounding box, shown in translucent blue, has been placed around the chair.

3. METHODOLOGY

Our proposed system uses an iPad connected to an Occipital Structure Sensor, shown in Figure 2. The Structure sensor contains a depth camera and processing unit that coordinates with the iPad's camera to perform hybrid RGB-D SLAM (Occipital Development Team, 2012). During the scanning process, the sensor provides our application a dense mesh of the environment, which gets rendered over the camera's view of the real world, as shown

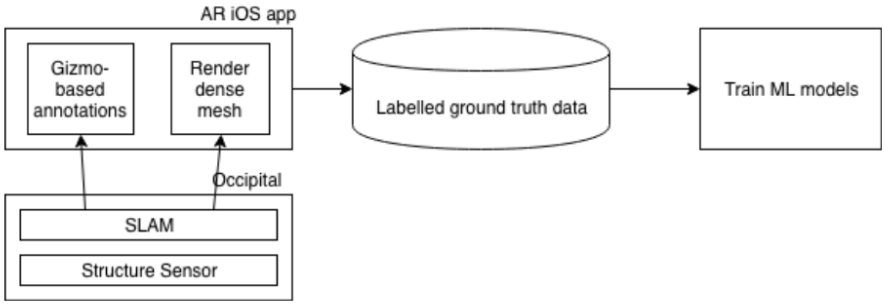

Figure 4: Overview of Augmented Annotations.

in Figure 3. A gizmo-style toolkit allows users to insert bounding boxes relative to the world and the mesh; an example of one is shown in Figure 3. Once the process is completed, the position, orientation, extents, and labels of the bounding boxes are exported alongside the mesh itself, where they can then be used as labeled ground truth data for training and testing machine learning models. An overview of the Augmented Annotations procedure is shown in Figure 4.

It is important to note that our system is not coupled tightly to the implementation of the underlying localization and mapping algorithms. As sensors become cheaper, more efficient, and higher quality, the Structure Sensor can be upgraded to take advantage of those improvements, with minimal changes to the software. Similarly, as RGB monocular SLAM approaches get more and more robust - or as smartphone manufacturers begin to integrate depth sensors in next-gen mobile devices such as the Samsung Galaxy S10 5G (Swider and McCann, 2019) - the external sensor will eventually become redundant and our system will work entirely off of suitably-equipped smartphones and tablets.

3.1 User Interface

The user interface for Augmented Annotations, shown in Figure 5, provides means of creating and manipulating annotations while the scan is ongoing. The supported functionality includes adding, removing, labeling, and transforming bounding boxes. The interface is operated entirely through intuitive taps and drags. Since the Occipital sensor performs SLAM for us, we can poll the position/rotation of the iPad relative to the physical environment at any time during the scanning process. This allows us to carry out the aforementioned operations in a coordinate frame that's locked to the real world.

When a user creates a new bounding box, it is initialized one meter in front of iPad. By default, the pitch and roll of the box are set such that the bottom face is perpendicular to the direction of gravity, since this is the most likely alignment of any physical object. The yaw is set to be the same angle as the yaw of the device, so that the edge of the box is parallel to the user at time of initialization. The size of the box is defaulted to 1 meter cubed.

After its creation, a bounding box can be manipulated through the use of tools called gizmos, shown in Figure 6. Gizmos are independent interaction schemes commonly used for 3D manipulation in CAD/CAM, animation, and game development workflows. Gizmos map 2D interactions on the screen (through single and multi-touch taps and drags) to corresponding transformations in three dimensions. Our primary three gizmos for scale, position, and rotation cover all possible 6DOF poses in space. As mentioned above, bounding boxes are positioned relative to the

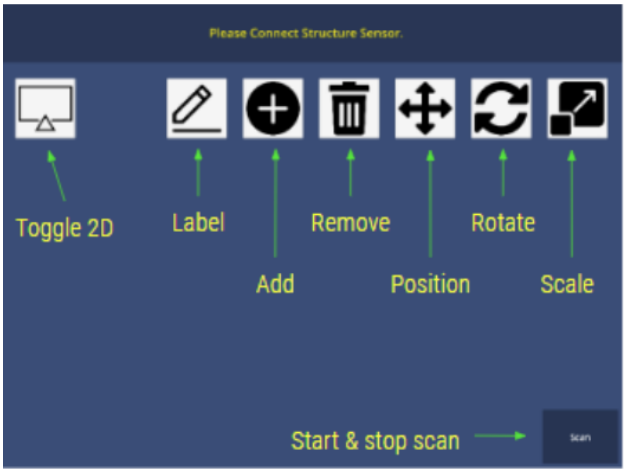

Figure 5: Augmented Annotation's user interface. Gizmo buttons have been labeled with their corresponding functionality.

real world environment, and maintain their positions even as the iPad moves. This enables users to physically adjust themselves in order to view the scene or a particular box from a better perspective. Tapping the label button brings up a text box which allows the user to assign a label to the selected bounding box, using the on-screen keyboard.

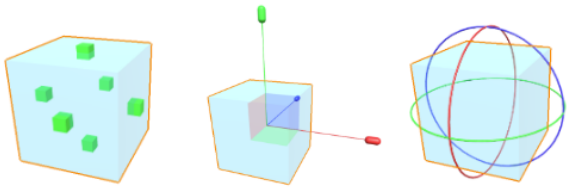

Figure 6: Gizmos for the scale, position, and rotation tools respectively.

The process of accurately positioning bounding boxes requires viewing the placed boxes from multiple angles. This has the added benefit of ensuring that the scanner observes the objects of interest from many angles. As a result, the resulting scan is complete, and there are few holes from occluded viewpoints. The quality of the mesh is especially good around the labelled objects, since the user spends the most time looking around those objects. In object recognition tasks, these are the most important parts of the mesh, so it is desirable to obtain high resolution at these locations.

3.2 Automatic 2D Bounding Box Generation

Given a 3D bounding box, our system can also generate a 2D bounding box around the object from any perspective, even after the capture is complete. This is done by taking the vertices of the 3D box, projecting them to the camera's image plane, and determining the minimum area rectangle that encapsulates those points as shown in Figure 7. This process requires no additional input from the user after creating the initial 3D bounding box, and unique perspectives can be generated as fast as the user can move the iPad. Although we do not go further than developing this feature in this paper, we believe this capability is invaluable in generating viewpoint variation of objects in the augmentation of 2D recognition datasets.

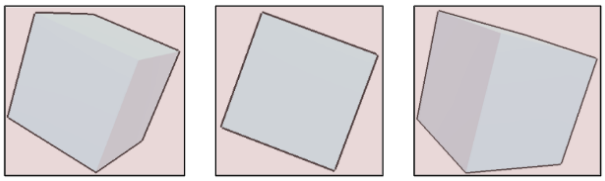

Figure 7: 2D bounding boxes created around a synthetic 3D bounding box. The same 3D box is used in each image, with only the camera perspective changing between images.

4. EXPERIMENTAL RESULTS

To evaluate the efficacy of our tool, we compare it to SUN RGB-D, a scanning and annotation system used to generate the eponymous dataset. In SUN's system, the environment is first scanned ahead of time using an RGB-D sensor. The mesh of the scan is then uploaded to the desktop tool, where workers annotate objects within the scene. For the purposes of comparison, we build a replica of that annotation tool, shown in Figure 8, based off of their description of the tool in (Song et al., 2015), using the Structure Sensor for RGB-D capture. This replica is evaluated alongside our Augmented Annotation system. Below we describe the procedure for our study and the results it produced.

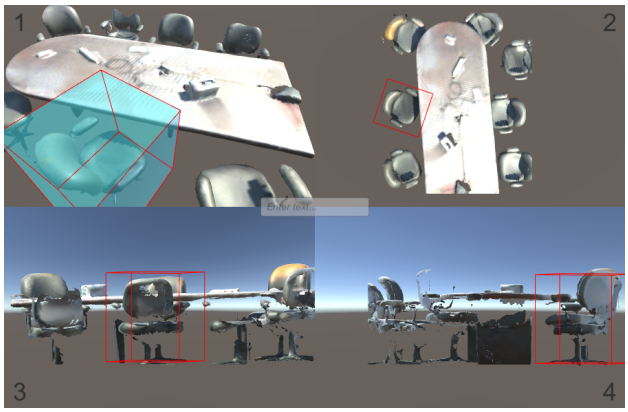

Figure 8: A screenshot from our replica of the SUN RGB-D annotation tool displaying environment 1. Top left: an arbitrary perspective view. Top right: bird's eye view. Bottom left and right: orthogonal side views.

4.1 Experimental Procedure

In our experiment, seven subjects are presented with three indoor environments, along with a list of notable objects in each scene. The area and number of objects for each environment is shown in Table 1, and pictures of each environment are shown in Figure A1 of the Appendix. Subjects are tasked with scanning each environment and annotating the listed objects. They accomplish this twice via two separate procedures: once using our replica of the SUN RGB-D tool, and once using the Augmented Annotations system. The total time to completion for each environment is recorded. To minimize bias, half of our participants start with the SUN tool first, while the other half start with the AA tool. To familiarize participants with the tools, they are instructed to practice scanning and annotating their nearby area at the beginning of the study. The times from this practice run are not recorded.

Environment	Area (sq. ft.)	No. of Objects
1	189	7
2	71	8
3	85	11

Table 1: The area and number of required objects for each of the environments used in the experiment.

For the SUN RGB-D procedure, an environment is first scanned with the Occipital Structure Sensor and iPad. The scan is then imported into our replica of SUN's annotation tool. The time it takes to import is not included, since with large-scale datasets this step is heavily batched and takes relatively little time. In this tool, the user is presented with four quadrants, each presenting a different view of the environment. Subjects click to draw out a rectangle in the top-down view, which initializes a bounding box. They then label the object using the center text box, and adjust the bounding box's height in one of the side views.

For the Augmented Annotations procedure, participants use our app on the Structure sensor and iPad. They make use of the tools and gizmos described above to create bounding boxes in the environment while simultaneously generating a scan of the environment.

4.2 Results

The resulting times of each participant and averages per room are shown in Table 2. We find that our system completes the scanning and annotation process significantly faster than SUN's approach in environments 2 and 3, and is only slightly slower in environment 1.

We see especially significant gains in environments that are cluttered. This is because such a scene projects poorly to 2D, making it difficult to distinguish and thus annotate objects using the 2D perspectives afforded by SUN's annotation tool. In large, uncluttered environments such as environment 1, we observe performance on par or slightly worse than the baseline desktop system. That is because such scenes can be described sufficiently well in 2D, as there is no variance along the top-down axis. From Figure 8, it is apparent that in environment 1, bounding boxes can be determined from the top-down view alone. On the depth axis the bottom and top faces are all uniform, and there is no overlap of objects. This effect is amplified in large rooms, because the user is forced to walk larger distances for relatively little viewpoint variation. Therefore, our system works best in small, cluttered environments comprised of irregular geometries with variance in all three dimensions. An example of one is environment 3, shown in Figure A2 of the Appendix, which is also the environment we see the most improvement in.

We also find variance produced as a result of differing familiarity with the technology. For example, subject A had significantly more experience with AR applications than subject D, and as a

Environment	Participants															
	A		B		C		D		E		F		G		Average	
	SUN	AA	SUN	AA	SUN	AA	SUN	AA	SUN	AA	SUN	AA	SUN	AA	SUN	AA
1	3:25	2:59	3:28	3:12	4:09	4:25	3:25	3:29	3:17	2:53	4:43	4:35	3:21	4:40	**3:41**	3:44
2	3:56	2:30	4:00	3:40	3:55	3:35	3:57	4:19	4:26	3:37	4:12	4:40	3:57	2:51	4:03	**3:36**
3	6:49	5:15	5:54	4:30	5:27	5:01	5:38	5:01	5:20	4:07	5:25	5:33	4:56	3:45	5:38	**4:44**

Table 2: The times taken for subjects to scan and annotate the environments shown in Figure A1. For *SUN* trials, the time required to scan and time required to annotate are summed together. All times are given in minutes and seconds.

result their Augmented Annotations trial times differed greatly. In contrast, all subjects had plenty of experience with mouse-based desktop applications, which can be seen from how the SUN trial times are much more tightly distributed per environment. This suggests that further user studies more rigorous than this one would be required to characterize the performance of our system in more depth.

While we did not quantitatively evaluate our 2D bounding box generation, examples of it used in physical scans can be found in Figure 9. We believe that one can expect significant speedups from this feature. Since the localization algorithm runs at approximately 30 frames per second on our hardware, we can generate 30 labelled 2D bounding boxes for every second of footage. As the user moves around the object and the environment, each 2D image would capture a different perspective of the object. Of course, not all 30 frames could be used; many frames would have to be discarded to prevent overfitting. Therefore the exact performance would depend on the architecture of the model being trained, characteristics of the data augmentation techniques applied, size of the training set, etc. Nonetheless, we expect this to be quite useful for 2D object recognition tasks.

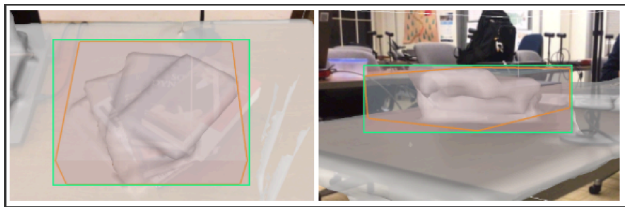

Figure 9: Screenshots of our app generating 2D bounding boxes around a stack of books. The orange wireframe is the outline of the 3D bounding box, and the shaded green rectangle is the 2D one. The computer mesh is shown in white, superimposed on top of the real world objects.

5. CONCLUSIONS AND FUTURE WORK

In this paper we introduce Augmented Annotations, a system for creating annotated 3D datasets that consolidates the scanning and annotation processes to save on time and effort. We build an iPad + Structure Sensor app that uses augmented reality to enable the real-time creation of bounding boxes relative to the physical world. Our experiment shows that our system outperforms or remains on-par with traditional methods in generating 3D and 2D bounding box data, with greater improvements seen in cluttered or irregular environments.

There are many potential improvements that could further improve Augmented Annotations. One participant suggested having a ”ghost box” that showed where a new box would be created. In general, we believe that the initialization process is key to making this process even faster - annotation would be significantly expedited if the system could make intelligent guesses about the initial placement of the bounding box. Some simple heuristics would be to align the bottom of the bounding box with the floor of the mesh, or scale the box based off of the camera’s current perspective. Going further, the system could make live predictions about where potential objects of interest might be based off of the contour or color of the mesh.

We also must consider hardware affordances of the system. Most obviously, our setup would not work with non-portable sensors. Next, we noticed some participants struggled to type on the on-screen keyboard, and believe that voice input would be a more efficient modality for object labeling. While the iPad + sensor performs SLAM and scans well at an affordable price, it also lacks a vital feature of higher-end AR headsets: depth perception, which would be extremely helpful in judging the depth of 3D objects and bounding boxes without having to switch perspectives.

On the UI/UX side, we designed the interface to be as intuitive and familiar to the user as possible by basing it off of pre-existing tools and interaction schemes. However, we potentially sacrificed efficiency to achieve this. Considering that most serious users of this application would have ample time to learn how to use the tool, gizmos with a harsher learning curve but higher skill ceiling might result in better performance.

Finally, bounding boxes are not the only form of annotation in the 3D domain. Other annotation tasks such as semantic segmentation could be improved through the mobile AR workflow presented by Augmented Annotations.

REFERENCES

6D Development Team, 2017. 6d reality platform. https://www.6d.ai/.

Abu Alhaija, H., Alhaija, H. A., Mustikovela, S. K., Mescheder, L., Geiger, A. and Rother, C., 2018. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. Int. J. Comput. Vis. 126(9), pp. 961–972.

Couprie, C., Farabet, C., Najman, L. and LeCun, Y., 2013. Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M., 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Vol. 1, p. 1.

Hays, J. and Efros, A. A., 2007. Scene completion using millions of photographs. ACM Transactions on Graphics (TOG) 26(3), pp. 4.

Hoffman, T., 2014. Occipital Structure Sensor Review Rating. https://www.pcmag.com/review/326635/occipital-structure-sensor.

Occipital Development Team, 2012. Structure Sensor. https://structure.io/.

Placenote Development Team, 2017. Placenote sdk. https://placenote.com/about-us/.

Russell, B. C., Torralba, A., Murphy, K. P. and Freeman, W. T., 2008. Labelme: a database and web-based tool for image annotation. International journal of computer vision 77(1-3), pp. 157–173.

Song, S., Lichtenberg, S. P. and Xiao, J., 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Strickland, J. C. and Stoops, W. W., 2018. The use of crowdsourcing in addiction science research: Amazon mechanical turk. Exp. Clin. Psychopharmacol.

Swider, M. and McCann, J., 2019. Samsung Galaxy S10 Plus Review. https://www.techradar.com/reviews/samsung-galaxy-s10-plus.

Wenkart, M., 2014. The Odesk Revolution: Borders are finally a thing of the past. BoD – Books on Demand.

APPENDIX

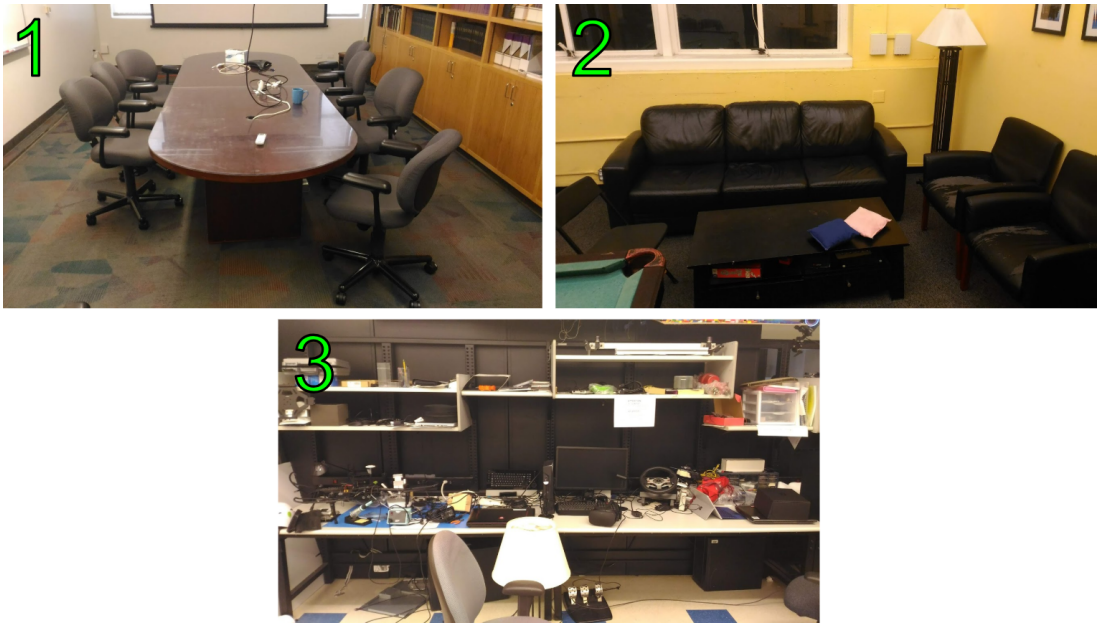

Figure A1: The three environments scanned and annotated for the study. The numbers on the picture correspond to the numbers in Tables 1 and 2.

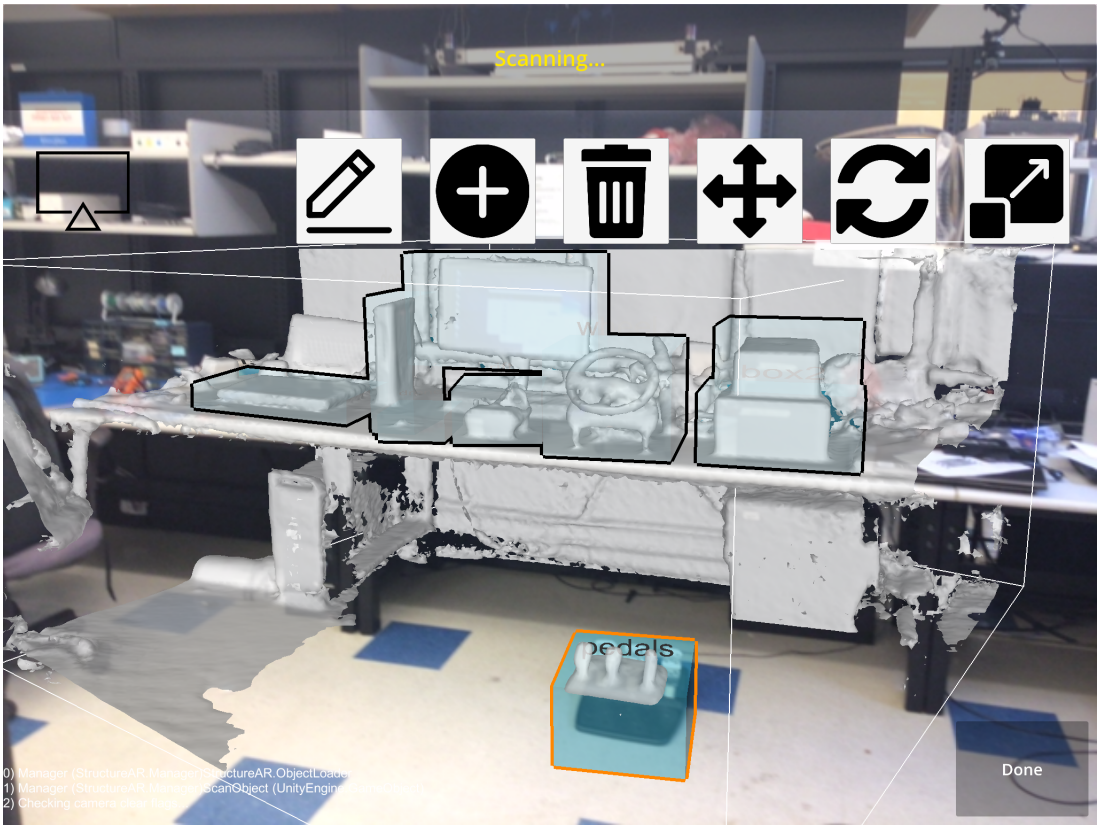

Figure A2: Screenshot of environment 3 after a user completed scanning and annotation using Augmented Annotations. Note the high amount of clutter and occlusion in multiple dimensions.