In-Situ Labeling for Augmented Reality Language Learning

Brandon Huynh[*]
University of California, Santa Barbara

Jason Orlosky[†]
Osaka University

Tobias Höllerer[‡]
University of California, Santa Barbara

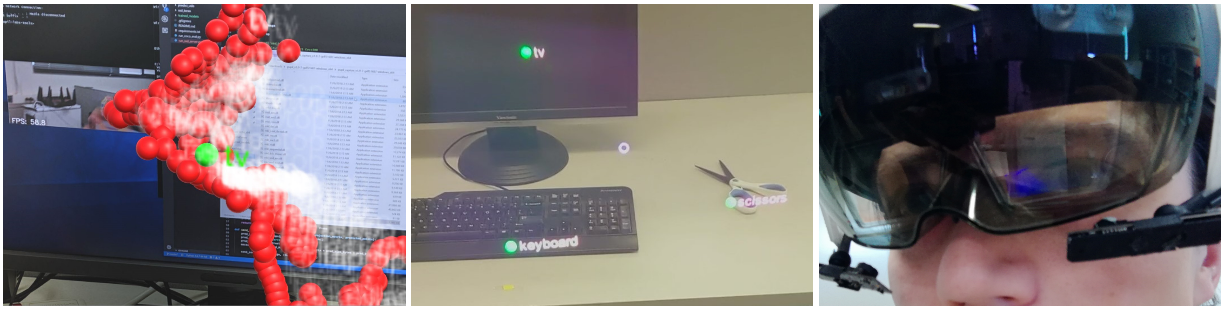

Figure 1: Images showing a) our object registration algorithm, which uses a set of uncertain candidate object positions (in red) to establish consistent labels (in green) of items in the real world b) a view directly through the HoloLens of resulting labels from our method in a previously unknown environment, and c) a photo of a user wearing the system and calibrated eye tracker used for label selection.

Abstract

Augmented Reality is a promising interaction paradigm for learning applications. It has the potential to improve learning outcomes by merging educational content with spatial cues and semantically relevant objects within a learner's everyday environment. The impact of such an interface could be comparable to the method of loci, a well known memory enhancement technique used by memory champions and polyglots. However, using Augmented Reality in this manner is still impractical for a number of reasons. Scalable object recognition and consistent labeling of objects is a significant challenge, and interaction with arbitrary (unmodeled) physical objects in AR scenes has consequently not been well explored. To help address these challenges, we present a framework for in-situ object labeling and selection in Augmented Reality, with a particular focus on language learning applications. Our framework uses a generalized object recognition model to identify objects in the world in real time, integrates eye tracking to facilitate selection and interaction within the interface, and incorporates a personalized learning model that dynamically adapts to student's growth. We show our current progress in the development of this system, including preliminary tests and benchmarks. We explore challenges with using such a system in practice, and discuss our vision for the future of AR language learning applications.

Index Terms: Human-centered computing — Mixed and augmented reality; Theory and algorithms for application domains — Semi-supervised learning;

1 Introduction

For many years, learning new words has often been accomplished by memorization techniques such as flash cards and phone or tablet based applications. These often use temporal spacing algorithms to modulate word presentation frequency such as Anki [11] and Duolingo [32]. A more effective, albeit time consuming, method of language learning is to attach notes with words and illustrated concepts to real world objects in a familiar physical space, taking advantage of the learner's capacity for spatial memory. Learners constantly see a particular object, recall the associated word and learn that concept more effectively since the object is in its natural context and is consistently viewed over time. This type of learning is also referred to as the method of loci [4, 23, 33].

Our goal is to replicate this in-situ learning process, but to do so automatically and with the support of augmented reality (AR), as represented in Fig. 1 b. In other words, when a user views an object, we want to automatically display the concept(s) associated with that object in the target language and provide a method for both the viewing and selection of a particular term or concept. Deploying such an interface in a real-world, generalized context is still a very challenging task.

As a step towards this goal, we introduce a more practical framework that can function as a cornerstone for improving in-situ learning paradigms. In addition to the process of trial and error to find a more effective and practical approach to designing such a system, our contributions include:

1. a client-server architecture that allows for real-time labelling of objects in an AR device (Microsoft HoloLens),
2. a description and solution to the object registration problem resulting from the use of real-time object detectors (Fig. 1 a),
3. a practical framework for exploring challenges in the implementation of AR language learning, and a discussion of novel interaction paradigms that our framework enables.

The practical use of this system can enable in-situ learning for languages, physical phenomena, and other new concepts.

2 Related Work

Prior work falls into three primary categories, 1) the implementation of object recognition, semantic modeling, and tracking for in-situ labeling, 2) view management techniques for labeling in AR, and 3) the use of AR and VR to facilitate learning of concepts and language. While all of these three categories are typically different areas of research, they are each essential for the effective implementation of in-situ AR language learning.

[*] e-mail: bhuynh@cs.ucsb.edu
[†] e-mail: orlosky@lab.ime.cmc.osaka-u.ac.jp
[‡] e-mail: holl@cs.ucsb.edu

2.1 Object Recognition and Semantic Modeling

Real-time object detection is a fairly new development, and there are not many works discussing the integration of these technologies into an augmented reality system. Current detection approaches utilize object recognition in 2D image frames, using learning representations such as Deep and Hierarchical CNNs and Fully-Connected Conditional Random Fields [6, 20], or, for fastest real-time evaluation performance just a single neural network applied to the entire image frame [28]. Combined 2D/3D approaches [1, 21] or object detection in 3D point cloud space [7, 27] may become increasingly feasible for real-time approaches in the not-too-far future as more 3D datasets [1, 7] become available, but currently, approaches that apply 2D object detection to the 3D meshes generated by AR devices such as HoloLens or MagicLeap One yield better performance.

Huang et al. [13] compare the general performance of 3 popular meta architectures for real-time object detection. They show that the Single Shot Detector (SSD) family of detectors, which predicts class and bounding boxes directly from image features, has the best performance to accuracy tradeoff. This is compared to approaches which predict bounding box proposals first (Faster-RCNN and R-FCN). We experimented with the performance of both types of detectors and ultimately settled on an implementation of SSD.

The most recent and closest work to our approach is that of Runz et al. [29] in 2018. Using machine learning and an RGBD camera, they were able to segment the 3D shapes of certain objects in real time for use in AR applications. Their approach utilized the Mask-RCNN architecture to predict per-pixel object labels, which comes at a higher performance cost. In contrast, our approach is implemented directly on an optical see-through HMD (HoloLens) using a client-server architecture, and uses traditional bounding box detectors which can run in true real-time (30fps) with few dropped frames.

Our work links objects that are recognized in real time in 2D frames to positions in the modeled 3D scene, which is akin to projecting and disambiguating 2D hand-drawn annotations into 3D scene space [18].

2.2 View Management for Object Labeling

A body of work in AR research focuses on optimized label placement and appearance modulation. In a similar fashion that we use 2D bounding boxes of recognized objects in the image plane to determine a 3D label position for that object, several view management approaches optimize the placement of annotations based on the 2D rectangular extent of 3D objects in the image plane [2, 3, 12]. Other approaches allow the adjustment of labels in 3D space [26, 30], a feature that might be gainfully employed in our system to subtly optimize the location of an initially placed label over time as multiple vantage points accumulate. However, this would pose the additional problem of disruptive label movement, due to loss of temporal coherence. Since potential mislabeling actions due to occlusions – the main motivation for 3D label adjustment – are automatically resolved by the HoloLens' continuous scene modeling (occluders are automatically modeled as occluding phantom objects), we can simply avoid label adjustment after we arrived at a good initial placement. Label appearance optimization [9] and assurance of legibility [10, 22] are beyond the scope of this paper.

2.3 Memory and Learning Interfaces

The idea of augmenting human memory or facilitating learning with computers appeared almost simultaneously with the history of modern computing. For example, early work by Siklossy in 1968 proposed the idea of natural language learning using a computer [31]. Since then, much progress has been made, for example by turning the learning process into a serious game [16]. Though not in an in-situ environment, Liu et al. proposed the use of 2D barcodes for supporting English learning. Though relatively simple, this method helps motivate the use of AR for learning new concepts, as a form of fully contextualized learning [25].

In addition to language learning, some work has been presented that seeks to augment or improve memory in general. For example, the infrastructure proposed by Chang et al. facilitated adaptive learning using mobile phones in outdoor environments [5]. Similarly, Orlosky et al. proposed the use of a system that recorded the location of objects, such as words in books, based on eye gaze, with the purpose of improving access to forgotten items or words [24].

Other studies like that of Dunleavy et al. found that learning in AR is engaging, but still faces a number of technical and cognitive challenges [8]. Kukulska-Hulme et al. further reviewed the affordances of mobile learning, having similar findings that AR was engaging and fun for the purpose of education, but found that technology limitations like tracking accuracy interfered with learning [17]. One more attempt at facilitating language learning by Santos et al. used a marker based approach on a tablet and tested vocabulary acquisition with marker-based AR. In contrast, our approach is designed to be automatic, and is a hands-free in-situ approach.

Most recently, Ibrahim et al. examined how well in-situ AR can function as a language learning tool [14]. They studied in-situ object labelling in comparison to a traditional flash card learning approach, and found that those who used AR remembered more words after a 4 day delayed post-test. However, this method was set up manually in terms of the object labels. In other words, the objects needed to be labelled manually for use with the display in real time. In order to use the display for learning in practice, these labels need to be placed automatically, without manual interaction.

This is the main problem our paper tackles. We have developed the framework necessary to perform this recognition, and at the same time we solve problems like object jitter due to improper bounding boxes. This sets the stage for a more effective implementation of learning via the method of loci, and can even enable reinforcement type schemes like spacing algorithms [11] that adapt to the pace of the user based on real world learning.

3 AR Language Learning Framework

As further motivation for this system, we envision a future where Augmented Reality headsets are smaller and more ubiquitous, and are capable of being worn and used on a daily basis much like current smart phones and smart watches. In such an ”always-on AR” future, augmented reality has the potential to transform language learning by adapting educational material to the user's own environment, which may improve learning and recall. Learning content may also be presented throughout the day, providing spontaneous learning moments that are more memorable by taking advantage of unique experiences or environmental conditions. Furthermore, an always-on AR device allows us to take into consideration the cognitive state of the user through emerging technologies for vitals sensing. Using this information, we can gain a better understanding of the user's attention, and more readily adapt to their needs. To enable research into these interaction paradigms, we propose a practical framework that can be implemented and deployed on current hardware using current sensing techniques. We believe the fundamental building blocks for AR language learning include three components:

- Environment sensing with object level semantics
- Attention-aware interaction
- Personalized learning models

These components provide the necessary set of capabilities required by the AR language learning applications we envision. In the next section, we will introduce a system design which implements this framework using existing technologies. Then, we will describe the realization of the first component of our framework, through an object level semantic labeling system. Finally, we will discuss our ongoing work regarding the second and third components.

4 System Design

In this section, we introduce a client-server architecture composed of several interconnected components, including the hardware used for AR and eye tracking, the object recognition system, the gaze tracking system, and the language learning and reinforcement model. The overall design and information flow between these pieces and parts is shown in Figure 2.

The combination of these pieces and parts allow us to detect new objects, robustly localize them in 3D despite jitter, shaking, and occlusion, and label the objects properly despite improper detection. Our current implementation targets English as a Second Language (ESL) students, thus our labels are presented in English. But the label concepts could be translated and adapted to many other languages.

4.1 Hardware

We chose the Microsoft Hololens for our display, primarily because it provides access to the 3D structure of the environment and can stream the 2D camera image to a server for object recognition. How we project, synchronize, and preserve the 2D recognition points onto their 3D positions in the world will be described later.

The HoloLens is also equipped with a 3D printed mount that houses two Pupil-Labs infrared (IR) eye tracking cameras, as shown in Fig. 1 c). These cameras are each equipped with two IR LEDs, and have adjustable arms that allow us to adjust the camera positions for individual users. The eye tracking framework employs a novel drift correction algorithm that can account for shifts on the users face.

For the server side of our interface, we utilized a VR backpack with an Intel Core i7-7820HK and Nvidia Geforce GTX 1070 graphics card. Since the backpack is designed for mobile use, this allows both the Hololens and Server to be mobile, as long as they are connected via network. To maximize throughput during testing and experimentation, we connected both devices on the same subnet.

4.2 Summary of Data Flow

Our system starts by initializing the Unity world to the same tracking space as the Hololens. Next, we begin streaming images from the Hololens forward-facing camera, which are sent to and from the server-side backpack via custom encoding. Upon reaching the server, they are decoded and input into the object recognition module, which returns a of 2D bounding box with an object label. The center of this bounding box is then sent back to the Hololens and projected into 3D world space by raycasting against the mesh provided by the Hololens. This projected point is treated as a "candidate point", which is fed into our object registration algorithm. The object registration algorithm looks over the set of candidate points over time to decide where to assign a final object label and position. Once an object and its position have been correctly assigned, the object is synchronized with the Unity space on the server side. Finally, labels on the objects are activated using eye-gaze selection, giving the user a method for interaction. The results from this interaction are fed into a personalized learning model, providing the ability to design content that adapts to the growth of the user.

5 In-Situ Labeling

The success of Convolutional Neural Networks (CNNs) has lead to technological breakthroughs in object recognition. However, it is not yet obvious how to integrate these technologies into AR. Three major parts need to be in place for these tools to be used practically. First, they need to be tested in practice (not just on individual image data sets) and provide good enough recognition to label an object correctly over time. Secondly, we need to establish object registration that is resilient to failed recognition frames, jitter, radical changes to display orientation, and objects entering/leaving the display's field of view (FoV). Finally, current AR devices are not powerful enough to run state-of-the-art CNNs. We need to handle the synchronization and reprojection between streamed frames from the AR device and recognition results from a server with a powerful GPU.

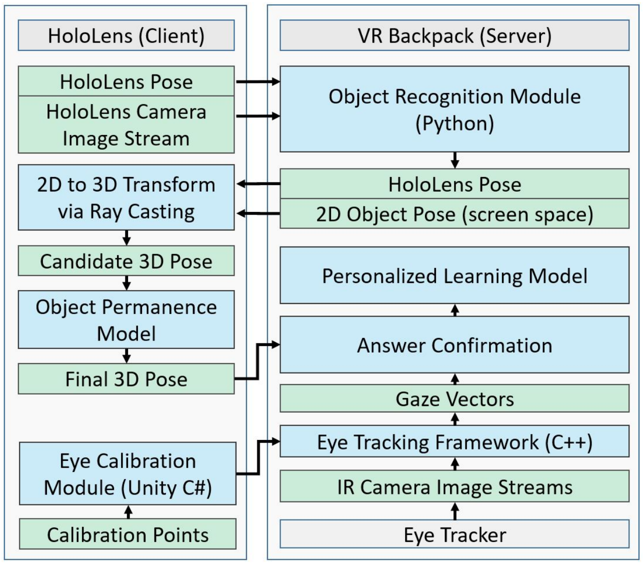
HoloLens (Client)
HoloLens Pose
HoloLens Camera Image Stream
2D to 3D Transform via Ray Casting
Candidate 3D Pose
Object Permanence Model
Final 3D Pose
Eye Calibration Module (Unity C#)
Calibration Points
VR Backpack (Server)
Object Recognition Module (Python)
HoloLens Pose
2D Object Pose (screen space)
Personalized Learning Model
Answer Confirmation
Gaze Vectors
Eye Tracking Framework (C++)
IR Camera Image Streams
Eye Tracker

Figure 2: Diagram of our entire architecture, including hardware in grey, algorithms and systems in blue, and data flow in green. The left-hand block includes all processing done on the Hololens and the right-hand block includes all processing done on the VR backpack.

5.1 Object Recognition Module

The first step for the development of our system was finding a scalable object recognition approach that could be used with the forward facing camera on the HoloLens. Due to the real-time performance constraint, we had to test and refine a variety of approaches before finding one that worked. We finally found the Single Shot MultiBox Detector (SSD) by Liu et al. to be effective [19]. Specifically, we use the implementation provided by the TensorFlow Object Detection API, using the ssd_mobilenet_v1_coco model, which has been pre-trained on MS COCO.

We stream video frames from the built-in HoloLens front facing camera to a server running on an MSI VR backpack. To keep packet sizes small, we used the lowest available camera resolution of 896x504. Each frame is encoded into JPEG at 50% quality, so that their final size fits into a single UDP packet. We also encode and send the current camera pose along with each frame. On the server side, we place all frames into an input queue. An asynchronous processing thread takes the most recent frame from the input queue and feeds it through the SSD network. The resulting 2D bounding boxes and labels are then sent back to the HoloLens, along with the original camera pose. Back on the HoloLens, we project the center point of each 2D bounding box onto the 3D mesh by performing a raycast from the original camera pose.

This particular implementation of SSD takes 30ms per prediction on the VR backpack, which just barely allows us to achieve 30fps under ideal network conditions. There is a slight delay due to network latency, as our network has a round trip time of 150ms.

SSD and similar CNN based real-time object recognition architectures are known to perform poorly with small objects [13]. In practice, we found that small objects, such as spoons and forks, experience much higher false positive rates and predictions are not

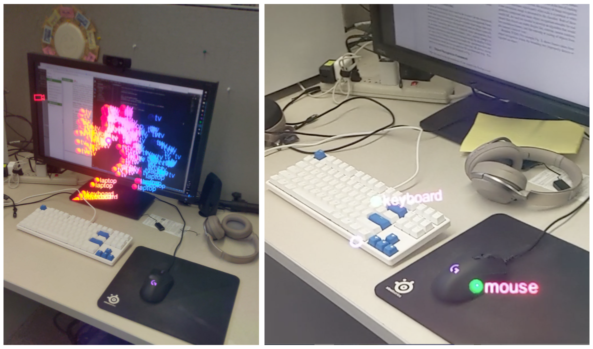

Figure 3: Left: Raw points returned from object recognition as projected into 3D space, accumulated over several frames. This shows the variance in predicted positions and false positive label predictions. Right: Scene correctly labeled with object-permanent labels.

consistent across frames. Large objects are more reliable, such as predictions for TVs, chairs, and people. For medium sized objects, typically performance improves under realistic environmental conditions where the camera is able to capture more contextual information, such as keyboards and mouses being near each other.

To solve this problem, we make use of multiple streamed frames to establish an initial estimate of the object's location, confirm this location using a sliding window approach based on past labels and proximity, and finally assign a position for the label. This results in a very stable, properly registered augmentation that is persistent despite various camera rotations or traveling in and out of various areas of a workspace. The algorithm we use for this purpose is described as follows:

First, an image streamed from the forward-facing HoloLens camera is passed to the SSD network, which then provides an initial prediction for a given object location in the form of a 2D bounding box. This 2D pose (i.e. the center of the bounding box in screen space) is then sent back to the HoloLens, and it is projected into 3D space as summarized previously.

Second, for every subsequent prediction, we check every instance of the same label in 3D space for the past W frames. A grouping of some of these labels can be seen on the left of Fig. 3. If the Euclidean distance between these subsequent 3D positions are within a threshold D (e.g. 50 centimeters away for a keyboard object), we average these positions and affix the object. After thorough testing and refinement, we found that object predictions converge well if there are 20 positively identified instances over a window of $W = 60$ frames under the defined threshold. An example of successful assignment of objects can be seen on the right of Fig. 3.

One advantage of this approach is that we can use semantic information to help guide the distance threshold. For example, a sofa might use points spaced one meter away versus a pencil with points less than ten centimeters away.

5.2 Evaluation of Object Registration

We performed a simple evaluation of our object registration algorithm in order to determine the quality of the label positioning (registration). To do so, we laid out 5 objects on a table: a computer monitor, keyboard, scissors, plastic bottle, and a paper cup. We marked a target point on the desk from which to compare each object and measured the distance with millimeter accuracy between the target point and the center of each object using a tape measure. This measurement served as the ground truth (GT in Table 1) for our position estimation.

During the evaluation, a user stood in a fixed position in front of the desk wearing the HoloLens and was given a handheld input device (a small bluetooth keyboard). The user is asked not to move or rotate their body but only their head. The user is instructed to look around the desk until the mesh is constructed, which is indicated by the appearance of a blue cursor in the center of the display. They were then asked to look at each object and confirm that a label has been placed for each object. Afterwards, the user was directed to point the blue cursor onto the marked target point and click a button on the handheld input device. This triggers a raycast from the center of the display in order to determine the target point pose within the HoloLens' coordinate system. We then measure the distance between the estimated label positions and the target point and compare them to ground truth in Table 1. This evaluation was conducted by 3 users who had some prior experience with the HoloLens.

Table 1: Data for ground truth (GT) and Estimation error in cm of the Euclidean distance between user-selected center points of each object in cm and a known 3D point in the tracking space.

Object	GT	User 1	User 2	User 3	Avg Error
TV	49.5	57.29	57.33	56.75	7.62
Keyboard	17.8	18.69	18.14	24.9	3.19
Scissors	50.8	50.3	51.06	51.32	0.90
Bottle	61	74.46	63.2	67.88	7.51
Cup	66	67.33	60.77	61.63	3.64
Overall					4.57

These preliminary results show that, on average, our object registration algorithm automatically converges on an object position up to 4.6cm away from the actual center position. Naturally, this is influenced by a number of factors, such as the size of the object to be labeled, and the initial vantage point when the label is first placed, but these values proved to be quite stable between users and repetitions.

In the future, we plan to evaluate performance on more challenging conditions. For instance, where the user is moving around the environment, or under poor lighting conditions. For now, the current registration performance is good enough for our needs.

6 Eye Tracking, Interaction, and Discussion

One more challenge in achieving a practical AR Language Learning system is the implementation of a method for selecting or activating an item for labelling. Simply labelling all objects in the environment is not feasible since the objects would clutter the users view, so a method (either active or passive) for selection or specification is necessary. We believe the natural solution is an attention-aware interface such as eye tracking. Such an interface allows us to deliver learning content when the user is in an amicable state, and provides interaction without a cumbersome external device or difficult to use gestures.

In order to facilitate basic interaction with content, we implemented a calibration framework for our system to allow users to activate items via eye gaze. Though the evaluation of this area is a work in progress, we describe the implementation, how eye gaze fits into our overall framework, and several possible mechanisms for interaction below.

6.1 Eye Tracking and Calibration Module

Gaze based selection of objects provides an intuitive interface for managing AR content without the need for additional input devices or complex gestures. Since individuals almost always tend to gaze upon an item or object when learning through the method of loci, unknown concepts should be displayed quickly. In this way, our learning framework allows us to explore the effects of passive learning, in which educational content may be consumed throughout a users daily routine.

Our calibration framework is based on the open source eye tracker built by Itoh et al. [15] for VR headsets, but with modifications made for the HoloLens. Much like a typical eye-to-video tracker calibration, we utilize a 5-point calibration interface in the Hololens. However, most eye tracking calibration procedures are executed with a sufficiently large field of view (FoV); i.e. the user gazes at several points on a 2D screen within the world-camera's wide FoV. In VR implementations, calibration points are often affixed to the display rather than registered in the world to counteract head movement. Since the Hololens FoV is only 35 degrees, we modified the same procedure used for VR and located vertical calibration points on the viewable portion of the screen. Though this can result in a minor reduction in vertical calibration accuracy, it sufficed for the purposes of activating labels on objects of interest.

6.2 Personalized Learning Model

The final component of our language learning framework is a personalized learning model. Specifically one that automatically adapts to the learners growth. We believe this is a fundamental difference between AR language learning and other existing language learning technologies. In our view, the future of augmented reality includes a collection of other vitals sensors which can monitor the physical and mental state of the user, similar to the trend of including health sensors in smartwatches. Already, we see devices like the Magic Leap One which include built-in eye trackers. This provides the ability to gauge the user's current understanding of the foreign language through continued monitoring of their cognitive response when consuming educational content.

As a first step, we plan to utilize eye and gaze signals, which have been shown to be good indicators of a users point of focus. To validate a users understanding of foreign words, we can use the duration of focus as an indicator of understanding. For example, labels that are gazed upon longer or multiple times within a short time period are likely to be unlearned. We plan to use these eye signals to develop a machine learning classifier that can detect whether a user understands or is confused about a foreign word. With such a classifier, we could identify how much foreign vocabulary a student has learned, and adapt by modifying the content (i.e, by introducing new words and removing words they have already learned).

We have recorded some preliminary results through a pilot study of 15 users. During the study, we presented English words in increasing difficulty to non-native English speakers while they wore a head mounted eye tracker. When presented with a word, the participants responded whether they did or did not know the meaning of the word. Afterwards, we developed an SVM classifier using the eye signals that was able to achieve 75% accuracy on the most difficult words. We plan to improve the performance by gathering more data and testing other classification techniques such as Recurrent Neural Networks.

6.3 Discussion and Future Work

Upon trying to implement a practical object labelling system in AR, we encountered many challenges that are not present in other object recognition implementations. For example, even though object recognition rates can exceed 90% on many 2D image datasets, this does not guarantee consistent use in the real world. Especially for a lower resolution camera that uses compressed images (such as the camera on the HoloLens), recognition from these algorithms is almost unusable unless modified as described in Section 5.1.

One other approach that we would like to explore is the re-training of object recognition models on video streams. Since integrated eye tracking in combination with the environment mesh can help determine the scale and depth of an object, we could potentially use this information to continuously re-train recognition for that particular object. User confirmation of recognition results also deserves consideration. For example, classification results may return the terms "tool" and "pen" for a ball-point pen. Allowing the user to select the term pen from a list could not only confirm the registered label in the immediate environment but improve recognition of that item upon the next encounter.

Our framework also tracks eye metrics such as pupil diameter and eye movement while users consume learning content in AR. As future work, we are investigating the use of machine learning based approaches to fuse and classify these signals for real time use. If we can automatically determine when a user understands a word, we can automate the learning algorithm used and suggest better, more relevant words to learn.

7 Conclusion

In this paper, we introduced a framework for realizing in-situ augmented reality language learning. As part of this framework, we describe our current progress implementing a client-server architecture that provides the ability to conduct both object recognition and environment mapping in real-time using a convolutional neural network. We explored the problem of object registration when using such a network, and provide a solution that accounts for the mismatched recognition errors that may occur. Our method is implemented directly on an AR headset. We described how to integrate eye tracking into our framework to allow for user selection or activation of annotations. We also described how to integrate a personalized learning model into our framework including initial results. We hope that this work will open up new avenues of research into methods and interactions for AR language learning and encourage others to contribute to this growing field.

Acknowledgements

This work was funded in part by the United States Department of the Navy, Office of Naval Research, Grants #N62909-18-1-2036 and #N00014-16-1-3002. Many thanks to Takemura Lab at Osaka University and the Four Eyes Lab at the University of California, Santa Barbara for supporting this collaboration.

References

[1] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. Feb. 2017. http://arxiv.org/abs/1702.01105.

[2] R. Azuma and C. Furmanski. Evaluating label placement for augmented reality view management. In *Proceedings of the 2nd IEEE/ACM international Symposium on Mixed and Augmented Reality*, p. 66. IEEE Computer Society, 2003.

[3] B. Bell, S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 101–110. ACM, 2001.

[4] G. H. Bower. Analysis of a mnemonic device: Modern psychology uncovers the powerful components of an ancient system for improving memory. *American Scientist*, 58(5):496–510, 1970.

[5] W. Chang and Q. Tan. Augmented reality system design and scenario study for location-based adaptive mobile learning. In *Computational Science and Engineering (CSE), 2010 IEEE 13th International Conference on*, pp. 20–27. IEEE, 2010.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Dec. 2014. http://arxiv.org/abs/1412.7062.

[7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:2432–2443, 2017. doi: 10.1109/CVPR.2017.261

[8] M. Dunleavy, C. Dede, and R. Mitchell. Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. *Journal of science Education and Technology*, 18(1):7–22, 2009.

[9] J. L. Gabbard, J. E. Swan, and D. Hix. The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor

augmented reality. *Presence: Teleoperators & Virtual Environments*, 15(1):16–32, 2006.

[10] J. L. Gabbard, J. E. Swan, D. Hix, S.-J. Kim, and G. Fitch. Active text drawing styles for outdoor augmented reality: A user-based study and design implications. In *Virtual Reality Conference, 2007. VR'07. IEEE*, pp. 35–42. IEEE, 2007.

[11] R. Godwin-Jones. Emerging technologies from memory palaces to spacing algorithms: approaches to secondlanguage vocabulary learning. *Language, Learning & Technology*, 14(2):4–11, 2010.

[12] R. Grasset, T. Langlotz, D. Kalkofen, M. Tatzgern, and D. Schmalstieg. Image-driven view management for augmented reality browsers. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pp. 177–186. IEEE, 2012.

[13] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, vol. 4, 2017.

[14] A. Ibrahim, B. Huynh, J. Downey, T. Höllerer, D. Chun, and J. O'donovan. Arbis pictus: A study of vocabulary learning with augmented reality. *IEEE transactions on visualization and computer graphics*, 24(11):2867–2874, 2018.

[15] Y. Itoh, J. Orlosky, and L. Swirski. 3D Eye Tracker Source. 2017. https://github.com/YutaItoh/3D-Eye-Tracker, accessed March 12th, 2018.

[16] W. L. Johnson, H. H. Vilhjálmsson, and S. Marsella. Serious games for language learning: How much game, how much ai? In *AIED*, vol. 125, pp. 306–313, 2005.

[17] A. Kukulska-Hulme. Will mobile learning change language learning? *ReCALL*, 21(2):157–165, 2009.

[18] K.-C. Lien, B. Nuernberger, T. Höllerer, and M. Turk. Ppv: Pixel-point-volume segmentation for object referencing in collaborative augmented reality. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pp. 77–83. IEEE, 2016.

[19] T.-Y. Liu, T.-H. Tan, and Y.-L. Chu. 2d barcode and augmented reality supported english learning system. In *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on*, pp. 5–10. IEEE, 2007.

[20] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical Convolutional Features for Visual Tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3074–3082. IEEE, 2015. doi: 10.1109/ICCV.2015.352

[21] F. Ma and S. Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. 2017. http://arxiv.org/abs/1709.07492.

[22] E. Mendez, S. Feiner, and D. Schmalstieg. Focus and context in mixed reality by modulating first order salient features. In *International Symposium on Smart Graphics*, pp. 232–243. Springer, 2010.

[23] A. Metivier. *How to Learn and Memorize German Vocabulary: ... Using a Memory Palace Specifically Designed for the German Language (and Adaptable to Many Other Languages Too)*. CreateSpace Independent Publishing Platform, 2012.

[24] J. Orlosky, T. Toyama, D. Sonntag, and K. Kiyokawa. Using eye-gaze and visualization to augment memory. In *International Conference on Distributed, Ambient, and Pervasive Interactions*, pp. 282–291. Springer, 2014.

[25] R. Oxford and D. Crookall. Vocabulary learning: A critical analysis of techniques. *TESL Canada Journal*, 7(2):09–30, 1990.

[26] S. Pick, B. Hentschel, I. Tedjo-Palczynski, M. Wolter, and T. Kuhlen. Automated positioning of annotations in immersive virtual environments. In *Proceedings of the 16th Eurographics conference on Virtual Environments & Second Joint Virtual Reality*, pp. 1–8. Eurographics Association, 2010.

[27] C. R. Qi, H. Su, M. Kaichun, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85. IEEE, July 2017. doi: 10.1109/CVPR.2017.16

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[29] M. Rünz and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. *arXiv preprint arXiv:1804.09194*, 2018.

[30] F. Shibata, H. Nakamoto, R. Sasaki, A. Kimura, and H. Tamura. A view management method for mobile mixed reality systems. In *IPT/EGVE*, pp. 17–24. Citeseer, 2008.

[31] L. Siklóssy. Natural language learning by computer. Technical report, Carnegie-Mellon University, Pittsburgh, PA, Dept. of Computer Science, 1968.

[32] L. Von Ahn. Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pp. 1–2. ACM, 2013.

[33] F. A. Yates. *The Art of Memory*. University of Chicago Press, 1966.