Semantic Labeling and Object Registration for Augmented Reality Language Learning

Brandon Huynh* University of California, Santa Barbara
Jason Orlosky† Osaka University
Tobias Höllerer‡ University of California, Santa Barbara

ABSTRACT

We propose an Augmented Reality vocabulary learning interface in which objects in a user's environment are automatically recognized and labeled in a foreign language. Using AR for language learning in this manner is still impractical for a number of reasons. Scalable object recognition and consistent labeling of objects is still a significant challenge, and interaction with arbitrary physical objects in AR scenes has consequently not been well explored. To help address these challenges, we present a system that utilizes real-time object recognition to perform semantic labeling and object registration in Augmented Reality. We discuss its implementation, our motivations in designing it, and how it can be applied to AR language learning applications.

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

1 INTRODUCTION AND BACKGROUND

Learning new vocabulary in a foreign language is often 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. One other interesting, albeit time consuming, method 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. This is also known as the method of loci [5].

Augmented Reality (AR) is a promising tool for this as it enables the integration and presentation of information over the real world. Recently, Ibrahim et al. examined how well in-situ AR can function as a language learning tool [2]. 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, 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.

Our goal is to replicate this in-situ learning process, but to do so automatically and with the support of AR, as shown on the right in Fig. 1. 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 AR learning paradigms. The practical use of this system can enable in-situ learning for languages, physical phenomena, and other new concepts.

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

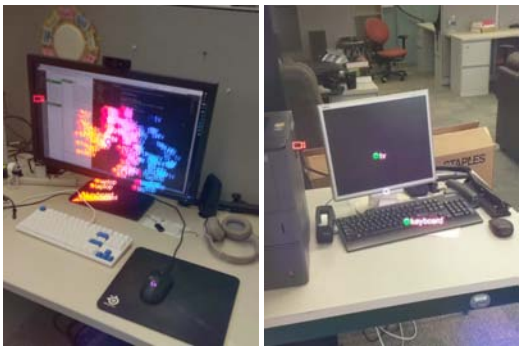

Figure 1: Left: Raw points returned from object recognition as projected into 3D space, accumulated over several frames. This shows the variance in predicted positions and false positive label predictions. Right: Scene correctly labeled after object registration.

2 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 system was implemented using a Microsoft HoloLens, with Pupil Labs eye tracker attachment.

2.1 Semantic 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. For our system, we want to be able to register objects such that they are resilient to failed recognition frames, jitter, radical changes to display orientation, and objects entering/leaving the display's field of view. Additionally, 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.

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. Frames are processed asynchronously using the Single Shot MultiBox Detector network [4]. The resulting 2D bounding boxes and labels are sent back to the HoloLens, along with the original camera pose, where we project the center point of each 2D bounding box onto the 3D mesh via raycast from the original camera pose. Our method runs in real-time (30 fps) on a per-frame basis.

2.2 Object Registration

The other major problem is establishing consistency of labeling, or object registration. CNN based object recognition approaches have no notion of object permanence as they are trained on data sets

2019 IEEE Conference on Virtual Reality and
3D User Interfaces
23-27 March, Osaka, Japan
978-1-7281-1377-7/19/$31.00 ©2019 IEEE

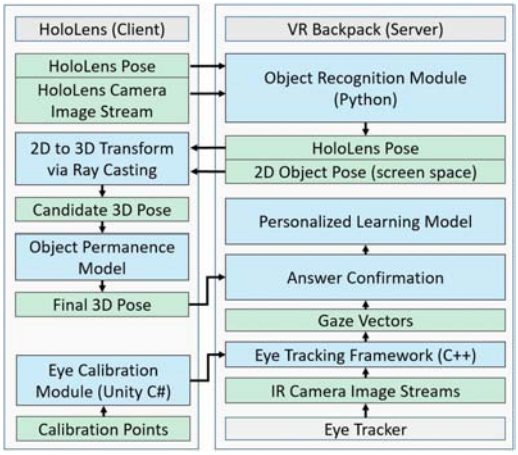

Figure 2: Architecture diagram, including hardware in grey, algorithms and systems in blue, and data flow in green. The left block includes all processing done on device, and the right block includes all processing done on the server.

with disparate images. They exhibit a large amount of variance in bounding box size and label predictions between sequential frames of the same object, as shown in the left side of Figure 1.

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 stable, properly registered augmentation that is persistent despite various camera rotations or traveling in and out of areas of a workspace. The algorithm we developed is described as follows:

First, we get an initial prediction from the network as described in the previous section. 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. 1. 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 $R = 20$ 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. 1.

We performed a preliminary evaluation of our object registration algorithm to determine the quality of the label positioning. 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 and measured the ground truth distance with millimeter accuracy between the target and the center of each object using a tape measure. We then asked 5 participants to map and generate labels for each object using the system. On average, our algorithm converges on an object position up to 4.6cm away from ground truth.

2.3 Eye Tracking and Calibration

Simply labelling all objects in the environment is not ideal as the objects would clutter the users view, so a method for selection or specification is necessary. We believe the natural solution is an attention based interface such as eye tracking. Such an interface allows us to deliver learning content only when the user is attending to the object. It provides an intuitive interface for managing AR content without the need for additional input devices or complex gestures. We implemented a calibration framework for our system to allow users to activate items via eye gaze.

Our calibration framework is based on the open source eye tracker built by Itoh et al. [3] 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.

3 Discussion and Future Work

When designing our system, we were motivated by the concept of Pervasive Augmented Reality [1], which predicts AR to be a continuous and multi-purpose experience that adapts to changes in the user's context. This is especially interesting for language learning, as it presents the opportunity for language immersion by sensing and translating the user's environment. This shifts the method of instruction to a passive learning model, which suggests the need for subtler selection methods that consider the user's attentiveness. Furthermore, we believe ubiquitous sensing of the user's cognitive state may be achievable through eye tracking or other sensors. This allows us to build a consistent and accurate model of a user's current understanding of a language. With such a model, we could dynamically adapt to learner growth. In vocabulary learning for example, we could replace learned words with new ones.

Our goal was to implement a practical system that could be used to study language learning applications in this emerging paradigm. Currently, we have implemented and described 2 of the components for this system: 1) Environment sensing (through object detection), and 2) Attention-based interaction (through eye tracking). As future work, we plan to implement the last component, a personalized learning model that tracks the user's current understanding of a language. We then plan to conduct a full evaluation of the system.

4 Conclusion

We introduced a framework for realizing in-situ augmented reality language learning and describe our current progress in implementing it. Our system performs object recognition and environment mapping in real-time using a CNN. 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 discuss how the combination of these technologies opens up new and interesting research directions for the growing field of AR language learning.

References

[1] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE transactions on visualization and computer graphics*, 23(6):1706–1724, 2017.

[2] 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.

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

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

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