OCTOPUS: Open-vocabulary Content Tracking and Object Placement Using Semantic Understanding in Mixed Reality

Luke Yoffe* Aditya Sharma* Tobias Höllerer

University of California, Santa Barbara

Abstract

One key challenge in augmented reality is the placement of virtual content in natural locations. Existing automated techniques are only able to work with a closed-vocabulary, fixed set of objects. In this paper, we introduce a new open-vocabulary method for object placement. Our eight-stage pipeline leverages recent advances in segmentation models, vision-language models, and LLMs to place any virtual object in any AR camera frame or scene. In a preliminary user study, we show that our method performs at least as well as human experts 57% of the time.[1]

Index Terms: Semantic Content Placement—Augmented Reality—Vision and Language—Large Language Models

1 Introduction

Augmented reality (AR) promises to seamlessly blend digital content with the real world, which requires placing virtual content in natural locations. For example, in Fig. 1, the plate is a natural location for a virtual cupcake to be placed. Currently, automated placement techniques do exist, but they are not able to work with arbitrary objects and scenes as the underlying machine learning models are closed-vocabulary. This means that the models are only able to handle a fixed set of words. Open-vocabulary models, on the other hand are able to adapt to words not seen during training. We combined several such models together to arrive at OCTOPUS, an eight-step method described in Sect. 3. OCTOPUS accepts as input an image of a scene and a text description of a virtual object, and determines where in the scene the object should be placed.

2 Related Work

In the context of automated virtual content placement, two interpretations of *natural* have been explored. First, virtual content should follow the laws of physics and be aligned with planar surfaces [8]. To evaluate object placement from a physical perspective, Rafi et al. [11] introduced a framework that predicts human ratings for object placements. *Natural* can also be interpreted from a semantic perspective, which is this paper's focus. Cheng et al. [3] and Lang et al. [6] used scene semantics to place virtual interface elements (such as virtual screens) and virtual agents respectively. These works focused on placing specific objects, whereas our goal is to create a single pipeline that can place any object with no dedicated training.

3 Method

Our virtual content placement pipeline is divided into eight steps:

1. Input: As input, our pipeline expects an image, which could be a camera frame from an AR application, and a text prompt naming the object to be placed.

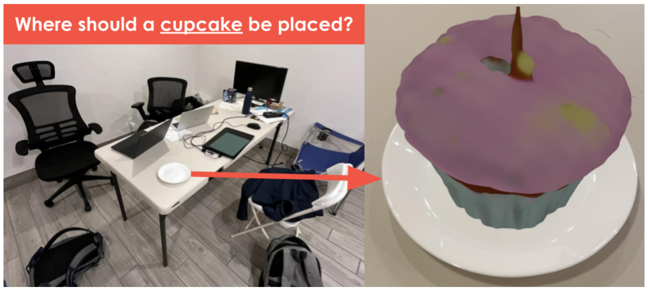

Figure 1: Result of our proposed method OCTOPUS, where we take an image of the scene (*left*) and determine a natural location for, e.g., a `cupcake` to be placed, such as the plate in the scene (*right*).

2. Image Segmentation: Next, we detect all potential objects in the scene using Segment Anything Model (SAM) [5]. SAM outputs many image regions that could potentially contain an object. We determine the bounding box for every region and retrieve the corresponding image patches.

3. Image Captioning: To identify the objects in each image patch, we leverage clip-text-decoder [9], an open source project. Clip-text-decoder generates a text caption for an image by encoding the image into an embedding (using CLIP: Contrastive Language-Image Pretraining [10]) and then decoding the embedding into text. For every image patch, we run clip-text-decoder to generate a caption.

4. Noun Extraction: Next, we list all objects referenced in each caption, since any of them could be sites to place the virtual object. All objects are nouns, so we use English Part-of-Speech tagging in Flair [1] to assign a part of speech to each word in each caption, keeping only the words marked as nouns.

5. Noun Filtration: Some of the nouns found may have been misidentified and we now filter out such cases. We use the Vision-and-language Transformer [4] (ViLT) model, which can perform visual question answering. We craft the question, "`Is there a` $\{noun\}$ `in the image?`", for each noun from Step 4. We include "`floor`" in the set of nouns as clip-text-decoder often failed to mention it. We then feed ViLT the image and ask the question for each noun, keeping the nouns that led ViLT to output "`yes`".

6. Noun Selection: To take advantage of LLM reasoning-like capabilities [2], we use prompt engineering on OpenAI's GPT-4 in order to select the noun where the object should be placed. We arrived at the following prompt for GPT-4: "`Give a one word response to fill in the blank using only one of these options:` $\{list\ of\ nouns\}$`. The` $\{object\}$ `was located on the _____.`", where the list of nouns is provided by Step 5. GPT-4 returns the most likely choice.

7. Location in Image: Next, we use CLIPSeg [7] to locate the selected noun in the image by feeding it the image and noun from GPT-4. CLIPSeg generates a heatmap indicating the similarity between each region in the image and the provided text prompt. We identify the brightest location (x,y) in the heatmap, which is the pixel in the image most related to the input noun.

8. Location in Scene: After determining the 2D (x,y) location

*Yoffe and Sharma contributed equally to this work

[1] This paper was published at the 2023 IEEE International Symposium on Mixed and Augmented Reality (https://ismar23.org/program-poster-presentations).

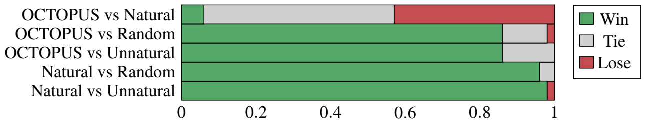

Figure 2: Evaluation results. “Win” and “Lose” indicate the proportion of time that the method listed first won or lost. “Tie” indicates the proportion of time that both methods’ placements were equally natural.

in the image, our last step is to find the corresponding 3D (x, y, z) position in the scene, which is where the virtual object will be placed in augmented reality. To accomplish this, we employ ray casting into the scene (modeled, e.g., by ARKit or ARCore). We place the object at the first intersection point between the ray and the scene.

4 Results

This chaining of semantic ML technologies provides remarkably robust performance with general scenes and objects. To measure the performance of the OCTOPUS method, we designed an experiment that compares four placement methods: (1) experts’ *natural* placements, (2) experts’ *unnatural* placements, (3) *random* placements, and (4) OCTOPUS placements.

Experiment Setup: In order to perform the experiment, a representative set of objects and images to test with was required. We created an unbiased diverse list of 15 objects that are commonly found indoors (`apple, cake, cup, plate, vase, stool, painting, lamp, book, bag, computer, pencil, shoes, cushion, cat`). We randomly sampled 100 indoor scene images from the NYU Depth Dataset [12] and Sun3D [13] to annotate with object placement locations.

Annotation: We had two experts annotate each of the 100 images with a *natural* and *unnatural* location to place each of the 15 objects. For example, in Fig. 1, it would be *natural* to place a cupcake on the plate, but *unnatural* to place a cupcake on the floor. Any objects that were deemed unsuitable or irrelevant for a specific image were excluded from further analysis throughout the experiment for that particular image (this happened in 573 of the 1,500 image-object combinations). We also generated placement coordinates using random point selection, and finally the OCTOPUS model. In the end, we arrived at 927 object location-image pairs for each of the four placement methods.

Evaluation: We compared two methods against each other at a time, omitting the comparison of unnatural and random placements, resulting in the five comparisons depicted in Fig. 2. In each one, evaluators were told what object was to be placed and were shown two images side-by-side. Both images were annotated with a red circle indicating the proposed placement location. The evaluators then selected which placement location was superior or declared a tie if both locations were deemed equally appropriate for the object in question. The evaluators did not know which method produced each placement location. We repeated this judgment task with 100 randomly sampled object-image pairs for the OCTOPUS vs. natural comparison, and 50 for each of the remaining four method duels.

The results, shown in Fig. 2, reveal that 57% of the time, OCTOPUS selected a location at least as natural as the human expert selecting a *natural* location. The experts’ natural locations won over the random and unnatural locations 96% and 98% of the time respectively, which confirms that they were indeed appropriate locations. OCTOPUS also won over the random and unnatural locations the vast majority of the time, demonstrating that it is tailored to human preferences and not far off from the experts’ *natural* placements.

5 Limitations and Future Work

While our method generally places objects naturally, it has limitations. First, it takes around 30 seconds to generate a single placement location on an NVIDIA RTX A4000, which could be impractical in real-world applications, in particular when making live queries with AR cameras. Additionally, while our method selects the best entity for virtual object placement, it does not consider where on the entity would appear the most natural. For example, OCTOPUS could place a painting on a wall, but would follow CLIPSeg’s highest heatmap response for `wall`, which may not match the natural eye level placement for paintings.

To support future work, we believe that an automated metric to determine the quality of semantic object placement would be of great value, as it could replace costly user studies.

6 Conclusion

We present OCTOPUS, a technique for placing virtual content in augmented reality. OCTOPUS takes as input an image and a text description of an object to be placed in the scene. It then detects entities in the image and uses LLM reasoning to determine the best entity for the object to be placed on. Lastly, it locates and places the object on the selected entity. The entire OCTOPUS pipeline is open-vocabulary, meaning it can be used to place any object in any scene out of the box, without any fine tuning. We find in preliminary evaluation that over 57% of the time OCTOPUS places objects at least as naturally as human experts.

References

[1] A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

[2] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[3] Y. Cheng, Y. Yan, X. Yi, Y. Shi, and D. Lindlbauer. Semanticadapt: Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 282–297, 2021.

[4] W. Kim, B. Son, and I. Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pp. 5583–5594. PMLR, 2021.

[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[6] Y. Lang, W. Liang, and L.-F. Yu. Virtual agent positioning driven by scene semantics in mixed reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 767–775. IEEE, 2019.

[7] T. Lüddecke and A. Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7086–7096, 2022.

[8] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson. Snaptoreality: Aligning augmented reality to the real world. In *Proc. of ACM CHI 2016*, pp. 1233–1244, 2016.

[9] F. Odom. clip-text-decoder. `https://github.com/fkodom/clip-text-decoder`, 2022.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[11] T. Rafi, X. Zhang, and X. Wang. PredART: Towards automatic oracle prediction of object placements in augmented reality testing. In *37th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1–13, 2022.

[12] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. *ECCV (5)*, 7576:746–760, 2012.

[13] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pp. 1625–1632, 2013.