LOBSTAR: Language Model-based Obstruction Detection for Augmented Reality

Yanming Xiu[1] Tim Scargill[2] Maria Gorlatova[3]

Department of Electrical and Computer Engineering, Duke University

ABSTRACT

In Augmented Reality (AR), improper virtual content placement can obstruct real-world elements, causing confusion and degrading the experience. To address this, we present **LOBSTAR** (**L**anguage model-based **OBST**ruction detection for **A**ugmented **R**eality), the first system leveraging a vision language model (VLM) to detect key objects and prevent obstructions in AR. We evaluated LOBSTAR using both real-world and virtual-scene images and developed a mobile app for AR content obstruction detection. Our results demonstrate that LOBSTAR effectively understands scenes and detects obstructive content with well-designed VLM prompts, achieving up to 96% accuracy and a detection latency of 580ms on a mobile app.

Index Terms: Mixed / augmented reality—Vision Language Models—Obstruction Attack—Prompt Engineering;

1 INTRODUCTION

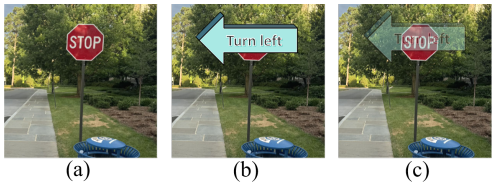
Augmented Reality (AR) places virtual content in the real world to assist users with various tasks. However, improper content placement can harm users by leading to the neglect or misunderstanding of real-world information [1]. A typical example is the obstruction attack [2, 8], where virtual content blocks real-world objects. This issue is particularly severe when the obstructed object is critical to task performance or user safety. For example, in Figure 1, a virtual navigation arrow obstructs a real stop sign, which may cause the user to turn directly onto the road and lead to potential accidents.

Therefore, methods are needed to evaluate the quality of AR content placement to tackle obstruction attacks. One approach is applying and extending full reference image quality assessment methods [3]. However, traditional methods often rely on local image features and lack a holistic understanding of environmental information, especially in complex scenes. This makes it difficult for these methods to accurately detect key objects or parts in the image, resulting in unreliable obstruction detection.

Recent advancements in scene understanding have been driven by Vision Language Models (VLMs) that combine object recognition and visual question answering capabilities. Unlike traditional methods, VLMs like GPT-4v [7] integrate visual and textual data, enabling a more comprehensive understanding of complex scenes. These models excel in context-aware analysis, grasping nuanced object relationships, and generating detailed descriptions that resemble human perception. Therefore, leveraging VLMs for their enhanced scene comprehension and robust contextual insights should offer an improvement over classic methods in detecting obstruction attacks.

In this work, we introduce **LOBSTAR**, a system based on VLMs and advanced object detection models for monitoring and evaluating AR content placement. By analyzing both raw and augmented images, LOBSTAR detects potential obstructions and provides feedback to optimize the user experience. To our knowledge, LOBSTAR is *the first system to leverage VLMs for detecting obstruction attacks in AR*. It provides robust obstruction detection and scene understanding with low latency when tested on Android mobile devices.

{yx246[1], ts352[2], maria.gorlatova[3]}@duke.edu



(a) (b) (c)

Figure 1: Example of an obstruction attack in AR: (a) real-world view; (b) AR view with a stop sign obstructed by a virtual arrow; (c) obstruction is mitigated by making the virtual arrow translucent.

2 SYSTEM DESIGN

2.1 System Architecture

The LOBSTAR architecture (Figure 2) is deployed across three devices: an AR device, an edge server, and a cloud server. The data transmission between them is conducted using the HTTP protocol.

AR device: continuously captures raw camera images and overlays virtual content. It sends those images and the spatial information of virtual content to the edge server and receives obstruction detection results. When an obstruction of a key object is detected, the virtual content is made translucent to ensure the key object is still visible.

Edge server: receives raw images and virtual content information from the AR device. Each image is encoded and sent to the cloud server with a text prompt, managed by a prompt controller to avoid unnecessary resource use, ensuring prompts occur only when requested by users. The VLM recognizes the key object in the image and returns it to the edge server. We designed a stack to store the key object name list, allowing LOBSTAR to recall what has been identified in the scene. Simultaneously, each raw image and the current key object's name are fed to a multi-modal object detection module, which produces bounding boxes for key objects. The boxes are put into a segmentation module to create binary masks of key objects, which are then compared with virtual content masks at the pixel level. The virtual content mask is generated based on the received spatial information. If a certain portion of the object is overlaid, a detection result "obstructed" is sent back to the AR device.

Cloud server: hosts a VLM. Deploying VLMs on edge servers is challenging due to their large number of parameters, making cloud servers a more feasible alternative. The VLM processes an encoded raw image and outputs the recognized key objects within it.

2.2 Machine Learning Model Selection and Integration

In LOBSTAR, three core modules use machine learning-based models to achieve robust scene understanding capabilities:

VLM: In this implementation, we use GPT-4v [7], a robust VLM ideal for tasks requiring integrating image and language information. Hosted on the OpenAI API, it offers scalable processing power for efficiently handling datasets and image streams.

Multi-modal object detection module: We use Grounding DINO [6], a state-of-the-art open-set object detection model. It can detect multiple objects based on a text prompt without the need to predefine the categories during training, making it ideal for LOBSTAR, as the category and number of key objects are unknown.

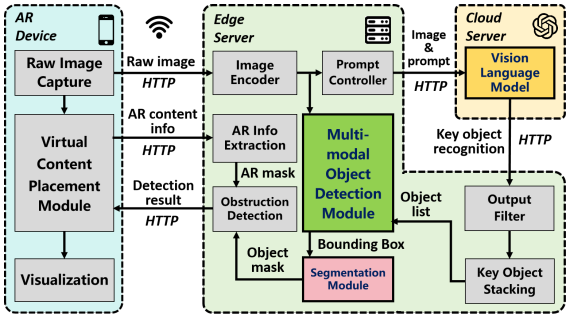


Figure 2: System architecture of the LOBSTAR system.

Segmentation module: We use the Segment Anything Model (SAM) [5] to generate binary masks of key objects. As a foundation model, SAM employs zero-shot learning to generalize across various domains without requiring extensive retraining. It also supports segmenting objects inside a specific bounding box, allowing for seamless integration with Grounding DINO.

3 System Evaluation

We evaluated LOBSTAR on (1) pre-collected image datasets using highly realistic virtual environments and (2) real-time image streams. We used a Google Pixel 7 Pro as the user AR device. The edge server was equipped with an NVIDIA GeForce RTX 3090. The cloud server was accessed through the OpenAI API.

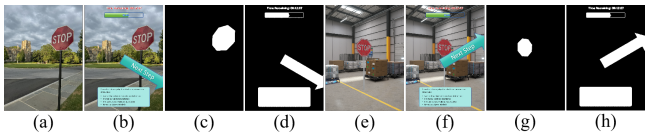
3.1 Dataset Results

We collected 144 images with 17 classes in real environments and 280 images with 15 classes in a high-definition virtual environment in Unity. Each image contained one key object. We randomly placed virtual contents on the images and created binary masks for both the key object and the virtual content. An image was labeled as "obstructed" if more than 15% of the key object was covered by virtual content. Figure 3 shows some examples from our dataset.

We crafted the following prompt for the VLM to identify key objects: "*Give me the name of the most important key object in the image. The objects can be signs, electrical devices, safety equipment... You can mention the color or shape if it is important.*" We also implemented three baselines and tested them with the datasets:
Blind: This baseline follows the same pipeline as LOBSTAR, but we prompt the VLM with minimal information and no object examples.
Prior: Directly inform the object detection module what key object is in the image, bypassing the need for the VLM.
Saliency: Calculate the mean saliency score [4] of the entire image $\overline{S_i}$ and the obstructed area $\overline{S_o}$. If $\overline{S_o} > \overline{S_i}$, indicating that the overlaid area is richer in information, we identify it as an obstruction.
Our results (Table 1) show that LOBSTAR achieved $> 96\%$ obstruction detection accuracy, $> 84\%$ mIoU and $> 75\%$ detection confidence in both real and virtual environments, close to or higher than when prior knowledge of key objects was given. LOBSTAR also performed significantly better than the saliency-based method.

3.2 Real-time Application Results

We developed an Android AR app for safety inspection to evaluate LOBSTAR on real-time image streams. A virtual text box with safety instructions was overlaid on the real environment. To measure latency, we first prompted GPT to initialize the key object list and



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3: Dataset samples: (a-d) real environment: original image, AR view, object mask, virtual content mask; (e-h) virtual environment: original image, AR view, object mask, virtual content mask.

Table 1: Comparison of obstruction detection accuracy, segmentation mIoU, and mean object detection confidence in real and virtual environments using LOBSTAR and three baseline approaches.

Dataset	Experiment	Obstruction Detection Accuracy	Segmentation Mean IoU	Detection Mean Confidence
Real	**LOBSTAR**	**96.53%**	**84.65%**	**75.35%**
	Blind	91.67%	76.78%	71.43%
	Prior	94.44%	87.14%	69.83%
	Saliency	56.25%	-	-
Virtual	**LOBSTAR**	**96.42%**	**85.99%**	**77.76%**
	Blind	92.14%	78.44%	74.57%
	Prior	97.14%	87.56%	73.73%
	Saliency	62.00%	-	-

measured the time between obstruction and response. We tested two network configurations: the phone and server on the same network and different networks, conducting 20 trials for each. With a 5GHz WiFi (802.11ac) connection, the mean latency was 580ms. When tested in a remote communication scenario, it took six hops for the phone to connect to the server, and the mean latency was 950ms. This low latency facilitates a swift response to obstruction attacks.

4 Conclusion and Future Work

In this work, we presented LOBSTAR, the first system using a VLM to detect obstruction attacks in AR. Our system demonstrated robustness and efficiency across static datasets and real-time image streams. In our future work, we will conduct user studies with real-world AR-assisted tasks to evaluate the impact of LOBSTAR on users' ability to perceive the environment accurately and perform those tasks correctly. We will also apply LOBSTAR to head-mounted devices such as the Apple Vision Pro, where obstruction attacks are more significant because the AR view covers the entire field of view of the user. Through these efforts, we will further refine AR content evaluation to optimize AR users' safety and quality of experience.

Acknowledgments

We thank Junfeng Lin for assisting with image collection for the dataset. This work was supported in part by NSF grants CSR-2312760, CNS-2112562 and IIS-2231975, NSF CAREER Award IIS-2046072, NSF NAIAD Award 2332744, and Defense Advanced Research Projects Agency Young Faculty Award HR0011-24-1-0001. This paper has been approved for public release; distribution is unlimited. The contents of the paper do not necessarily reflect the position or the policy of the Defense Advanced Research Projects Agency. No official endorsement should be inferred.

References

[1] K. Cheng, J. F. Tian, T. Kohno, and F. Roesner. Exploring user reactions and mental models towards perceptual manipulation attacks in mixed reality. In *Proceedings of USENIX Security*, 2023.
[2] S. Davari, F. Lu, and D. A. Bowman. Occlusion management techniques for everyday glanceable AR interfaces. In *IEEE VR Abstracts and Workshops (VRW)*, 2020.
[3] H. Duan, X. Min, Y. Zhu, G. Zhai, X. Yang, and P. Le Callet. Confusing image quality assessment: Toward better augmented reality experience. *IEEE Transactions on Image Processing*, 31:7206–7221, 2022.
[4] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proceedings of IEEE/CVF CVPR*, 2007.
[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.
[6] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv:2303.05499*, 2023.
[7] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.
[8] M. Satkowski, R. Rzayev, E. Goebel, and R. Dachselt. ABOVE & BELOW: Investigating ceiling and floor for augmented reality content placement. In *Proceedings of IEEE ISMAR*, 2022.