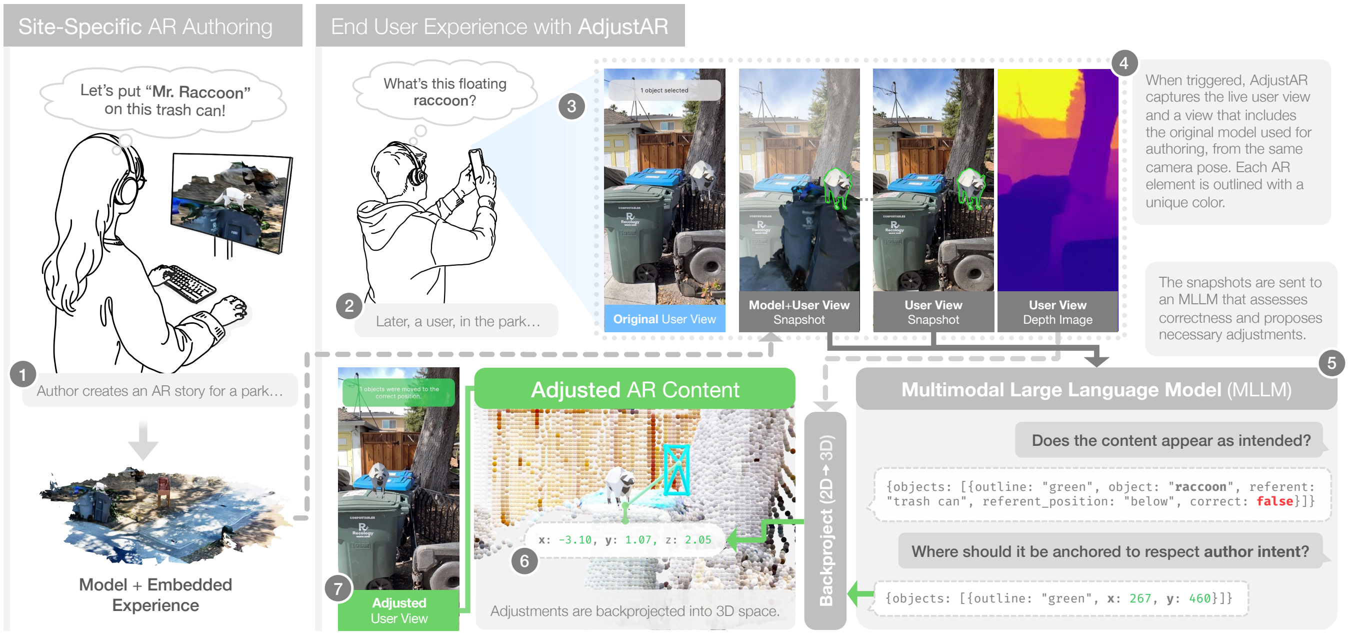
AdjustAR: AI-Driven In-Situ Adjustment of Site-Specific Augmented Reality Content

Nels Numan
University College London
London, United Kingdom
nels.numan@ucl.ac.uk

Jessica Van Brummelen
Niantic Spatial, Inc.
London, United Kingdom
jess@nianticspatial.com

Ziwen Lu
University College London
London, United Kingdom
ziwen.lu@ucl.ac.uk

Anthony Steed
University College London
London, United Kingdom
a.steed@ucl.ac.uk


Site-Specific AR Authoring
Let's put "Mr. Raccoon" on this trash can!
1
Author creates an AR story for a park…
Model + Embedded Experience
End User Experience with AdjustAR
What's this floating raccoon?
2
Later, a user, in the park…
3
1 object selected
Original User View
Model+User View Snapshot
User View Snapshot
User View Depth Image
4
5
Multimodal Large Language Model (MLLM)
Does the content appear as intended?
Where should it be anchored to respect author intent?
Backproject (2D → 3D)
Adjusted AR Content
6
7
1 objects were moved to the correct position.
Adjusted User View

When triggered, AdjustAR captures the live user view and a view that includes the original model used for authoring, from the same camera pose. Each AR element is outlined with a unique color.

The snapshots are sent to an MLLM that assesses correctness and proposes necessary adjustments.

```
{objects: [{outline: "green", object: "raccoon", referent:
"trash can", referent_position: "below", correct: false}]}
```

```
{objects: [{outline: "green", x: 267, y: 460}]}
```

```
x: -3.10, y: 1.07, z: 2.05
```

Adjustments are backprojected into 3D space.

Figure 1: AdjustAR corrects misaligned AR content at runtime: (1) authors place content relative to a georeferenced 3D model of the target site; (2–3) users localize and view the scene *in-situ*, where misalignments may occur due to environmental changes; (4) the system composites live and authored views; (5) an MLLM detects misalignments and infers corrected 2D anchors; (6) corrections are backprojected into 3D and updated in the scene; (7) the user's AR view is updated.

Abstract

Site-specific outdoor AR experiences are typically authored using static 3D models, but are deployed in physical environments that change over time. As a result, virtual content may become misaligned with its intended real-world referents, degrading user experience and compromising contextual interpretation. We present AdjustAR, a system that supports in-situ correction of AR content in dynamic environments using multimodal large language models (MLLMs). Given a composite image comprising the originally authored view and the current live user view from the same perspective, an MLLM detects contextual misalignments and proposes revised 2D placements for affected AR elements. These corrections are backprojected into 3D space to update the scene at runtime. By leveraging MLLMs for visual-semantic reasoning, this approach enables automated runtime corrections to maintain alignment with the authored intent as real-world target environments evolve.

CCS Concepts

- **Human-centered computing** → **Mixed / augmented reality**;
- **Computing methodologies** → *Scene understanding*.

Keywords

augmented reality, multimodal large language models, runtime adaptation, site-specific, authoring tools, context-adaptive systems

UIST Adjunct '25, Busan, Republic of Korea
2025. ACM ISBN 979-8-4007-2036-9/2025/09
https://doi.org/10.1145/3746058.3758362

ACM Reference Format:
Nels Numan, Jessica Van Brummelen, Ziwen Lu, and Anthony Steed. 2025. AdjustAR: AI-Driven In-Situ Adjustment of Site-Specific Augmented Reality Content. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25), September 28-October 1, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3746058.3758362

1 Introduction

Model-based authoring is the dominant approach for designing outdoor site-specific AR experiences [16]. Remote authoring tools [6, 14] enable anchoring virtual content relative to georeferenced 3D models via visual positioning systems (VPS). However, these models are static. Changes in the physical environment can cause misalignments between virtual content and its real-world referents, degrading user experience and requiring repeated site visits [16].

While prior systems have explored runtime adaptation through semantic constraints or layout optimization [2, 4, 12, 13], these efforts largely focused on user interfaces. *ScalAR* [17] applied this concept to AR authoring, introducing semantic-driven virtual proxies that supported model-based authoring. Li et al. [10] presented a system that extracted semantic structure from point clouds to retarget interactive narratives through story graph sampling. However, both approaches remained limited to indoor environments with narrow semantic vocabularies and relied on offline processing.

Recent advances in large language models (LLMs) and their multimodal variants (MLLMs) enable spatially grounded visual-semantic reasoning [1, 18]. Recent work has begun to explore LLMs for mixed-reality authoring [3, 5], with *ImaginateAR* [9] extending this concept to outdoor contexts through complex scene understanding and asset generation pipelines. While these systems expand the expressive potential of authoring, they have not focused on maintaining spatial consistency when the target environment changes.

To address this gap, we present AdjustAR, a system that combines model-based authoring with MLLM-guided visual-semantic correction. Authors follow standard creation workflows, while at runtime, AdjustAR compares rendered and live views to identify and correct misalignments that disrupt the intended experience. Rather than supporting re-authoring [16], AdjustAR aims to adaptively preserve the original design, treating the original authored scene as the canonical expression of author intent.

2 AdjustAR System

AdjustAR extends standard site-specific AR pipelines with a runtime correction mechanism driven by MLLMs (Fig. 1). Authors create experiences using the Niantic SDK for Unity [14], placing AR elements relative to site-specific 3D models. At runtime, users localize via VPS to align the scene with their view. When the physical target environment is unchanged, AR content appears correctly. However, when referents have moved or disappeared, visual misalignments may occur. To handle such cases, AdjustAR introduces a visual-semantic correction process that aims to address misalignments and restore author intent.

Visual Comparison for In-Situ Adjustment. AdjustAR performs a combined evaluation and adjustment process that can be triggered manually or periodically. When triggered, the system captures two snapshots from the current camera pose: the live AR view and a synthetic rendering of the authored scene, generated using the model that was originally used for authoring. Virtual elements are uniquely color-coded for content-agnostic object references.

The system concurrently caches the camera intrinsics, extrinsics, and depth map for each snapshot. These are later used to compute corrected 3D anchors once MLLM feedback is available, ensuring that repositioning aligns with the visual context at trigger time, even if the user has since moved.

The two images are passed to an MLLM (*Gemini 2.5 Flash* [7]) to assess whether each element appears aligned with its physical referent. For misaligned elements, another MLLM (*Gemini 2.5 Pro* [8]) is prompted to provide corrected 2D anchor points in image space, optionally including a 3D vertical offset. Both responses are returned in a JSON format based on a pre-defined schema. If the MLLM determines that a physical referent is no longer present or is fully occluded, AdjustAR displays a rendered snapshot of the original authored experience to provide the user with an indication of the intended experience. Prompts are detailed in Appendix A.

3D Repositioning via Backprojection. Corrected 2D anchors are backprojected into 3D world coordinates using the depth map and camera information cached at trigger time. The resulting 3D point becomes the new anchor, located at the bottom center of the element's bounding box. Optional vertical 3D offsets are applied to support elevated placements (e.g., hovering arrows). Finally, the AR scene is updated to reflect the adjusted anchor positions and is displayed to the user.

3 Future Work

Future work on AdjustAR should address system performance, authoring support, and empirical validation, including both quantitative evaluation and user studies in diverse deployment contexts.

Performance improvements may include reducing the correction pipeline latency (currently ~10-20 seconds) and improving accuracy, potentially through prompt optimization and additional input modalities such as depth, multi-view, or mesh data [19]. Advances in MLLMs, especially in spatial reasoning and 3D grounding, are likely to support these improvements [1, 18].

The current correction mechanism operates on static image pairs where referents and AR elements are visible within the same frame. Future work could incorporate spatiotemporal observations to handle occlusion or out-of-frame references. When referents are missing, virtual proxies may help preserve semantic continuity.

Placement decisions are currently based on a bottom-center heuristic, where the MLLM anchors objects relative to their base with an optional vertical offset. Future work could explore other anchoring strategies referencing surfaces, edges, or other geometric features [15, 17]. While prior work has examined semantically meaningful placements in narratives [10, 11], how closely AR content should align with target environments requires further exploration. For example, for a site-specific AR story, a character placed by a specific tree might appear near a similar one if the original is absent, whereas training applications may require exact replication.

Finally, supporting author-defined semantic constraints (e.g., "must be visible from entrance") could enable more precise intent specification and guide adaptation to contextual changes such as crowdedness, seasonal change, or other situational changes [12]. MLLMs offer a mechanism for interpreting such constraints flexibly, enabling adaptive behavior from sparse multimodal input.

Acknowledgments

This work was partially supported by the European Union's Horizon 2020 Research and Innovation program as part of project RISE under grant agreement No. 739578. We thank Simon Julier, Gabriel Brostow, and Niladri Dutt for valuable research discussions, and Dat Chu for assistance with testing logistics. The raccoon model used in this work was sourced from Google Poly.

References

[1] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14455–14465.

[2] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. 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*. ACM, Virtual Event USA, 282–297. doi:10/gm5z7h

[3] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. LLMR: Real-time Prompting of Interactive Worlds Using Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–22. doi:10.1145/3613904.3642579

[4] João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. 2022. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. In *The 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–16. doi:10.1145/3526113.3545651

[5] Daniele Giunchi, Nels Numan, Elia Gatti, and Anthony Steed. 2024. DreamCodeVR: Towards Democratizing Behavior Design in Virtual Reality with Speech-Driven Programming. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 579–589. doi:10.1109/VR58804.2024.00078

[6] Google. 2024. Geospatial Creator: Make the World Your Canvas. Retrieved 2025-06-02 from https://developers.google.com/ar/geospatialcreator

[7] Google. 2025. Gemini Flash 2.5. Retrieved 2025-06-02 from https://deepmind.google/models/gemini/flash/

[8] Google. 2025. Gemini Pro 2.5. Retrieved 2025-06-02 from https://deepmind.google/models/gemini/pro/

[9] Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. 2025. ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3746059.3747635

[10] Changyang Li, Wanwan Li, Haikun Huang, and Lap-Fai Yu. 2022. Interactive Augmented Reality Storytelling Guided by Scene Semantics. *ACM Transactions on Graphics* 41, 4 (July 2022), 1–15. doi:10.1145/3528223.3530061

[11] Wanwan Li, Changyang Li, Minyoung Kim, Haikun Huang, and Lap-Fai Yu. 2023. Location-Aware Adaptation of Augmented Reality Narratives. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3544548.3580978

[12] Zhipeng Li, Christoph Gebhardt, Yves Inglin, Nicolas Steck, Paul Streli, and Christian Holz. 2024. SituationAdapt: Contextual UI Optimization in Mixed Reality with Situation Awareness via LLM Reasoning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3654777.3676470

[13] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 147–160. doi:10.1145/3332165.3347945

[14] Niantic. 2024. Niantic SDK for Unity. Retrieved 2025-06-02 from https://www.nianticspatial.com/augment/sdk-for-unity

[15] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. 2016. SnapToReality: Aligning Augmented Reality to the Real World. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1233–1244. doi:10.1145/2858036.2858250

[16] Nels Numan, Gabriel Brostow, Suhyun Park, Simon Julier, Anthony Steed, and Jessica Van Brummelen. 2025. CoCreatAR: Enhancing Authoring of Outdoor Augmented Reality Experiences Through Asymmetric Collaboration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–22. doi:10.1145/3706598.3714274

[17] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–18. doi:10.1145/3491102.3517665

[18] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 10632–10643.

[19] Chun-Hsiao Yeh, Chenyu Wang, Shengbang Tong, Ta-Ying Cheng, Ruoyu Wang, Tianzhe Chu, Yuexiang Zhai, Yubei Chen, Shenghua Gao, and Yi Ma. 2025. Seeing from Another Perspective: Evaluating Multi-View Understanding in MLLMs. doi:10.48550/arXiv.2504.15280 arXiv:2504.15280 [cs]

A System Prompts

A.1 Initial Check for Correct Alignment (*Gemini Flash*)

```
You are a visual analysis AI agent. You are given a
side-by-side image showing two versions of the same
augmented reality (AR) scene:
-   Left image: AR as captured in real-world use.
-   Right image: AR as authored and intended to
appear.

For each outlined object, assess whether its
placement in the left image matches its placement in
the right image, relative to its physical referent. A
 physical referent is the real-world object, surface,
 or spatial location to which the AR content is
anchored or aligned. It provides the spatial or
semantic basis for interpreting the AR element in the
 physical environment and is often the nearest
visible surface or object.

For each object, indicate:
* The name of the physical referent.
* The position of the referent relative to the
outlined object in the right image (from the camera's
 perspective).
* Whether the placement in the left image is correct.
* Whether the physical referent is visible in the
left image.
```

A.2 Request for Adjusted Anchor Points (*Gemini Pro*)

You are a visual analysis AI agent. You are given a side-by-side image showing two versions of the same augmented reality (AR) scene:
- Left image: AR as captured in real-world use.
- Right image: AR as authored and intended to appear.

Your task is to determine whether each AR element (e.g., arrows, labels, icons) in the left image is correctly aligned with the same physical referent as in the right image.

A physical referent is the real-world object, surface, or spatial location to which AR content is anchored or aligned. It provides the spatial or semantic basis for interpreting the AR element in the physical environment and is typically the nearest visible surface or object.

If an AR element in the left image is misaligned, your task is to provide a corrected anchor position directly on the physical referent in the left image. If needed, also specify a vertical Y offset (in centimeters) indicating how far above or below this point the AR element should appear in 3D space. Your task is as follows:

1. Identify misalignments
Examine all AR elements in the left image. Each element is outlined in a unique color: "green", "blue", "magenta", "red", "orange", "yellow", or "cyan". For each element, assess whether it is anchored to the same physical referent as in the right image.

2. Correct misaligned elements
For each element that is not aligned with the correct referent:

- Specify a corrected anchor point in the left image.
- This point must lie directly on the same component of the physical referent, not near it or floating above it.
- Prioritize spatial accuracy. The position must align with the physical referent even if that referent has moved or changed appearance.
- Emphasize local visual consistency: prefer alignment with the object or surface the AR element was originally intended to refer to, rather than matching unrelated global features such as sky, shadows, or pavement.
- If either image is visually degraded or ambiguous, make the best possible contextual inference about the referent's location based on visible cues.
- Provide coordinates as follows:
* X and Y: Normalized to the left image dimensions, ranging from 0 to 1000. Origin (0, 0) is top-left; (1000, 1000) is bottom-right.
* Y offset (in centimeters): A vertical offset in 3D space from the anchor point. Positive values are upward, negative values are downward. If the element is placed directly on the referent or the vertical offset is unknown, use `0`.

3. Handle missing physical referents
If an AR element in the right image is meant to point to a physical referent that is no longer present or fully occluded in the left image, mark the element as "Missing".

4. Skip correct elements
If the AR element in the left image correctly points to or aligns with the same referent as in the right image, mark it as "Correct". Do not suggest changes.