TranslateAR: Towards Style-Preserving In-Situ Text Translation in Augmented Reality

Minbeom Kim
KAIST
Daejeon, Republic of Korea
kmb3322@kaist.ac.kr

Jason Kim
University of Washington
Seattle, Washington, USA
kimjason@cs.washington.edu

Davin Win Kyi
University of Washington
Seattle, Washington, USA
davin123@cs.washington.edu

Seok-Young Kim
KAIST
Daejeon, Republic of Korea
seokyoung@kaist.ac.kr

Jaewook Lee
University of Washington
Seattle, Washington, USA
jaewook4@cs.washington.edu

Jon E. Froehlich
University of Washington
Seattle, Washington, USA
jonf@cs.uw.edu

Figure 1: We introduce *TranslateAR*, a mobile AR system that translates real-world text directly within the live camera view while preserving its visual style and spatial context. The examples show translations across diverse text forms: a handwritten note, cereal packaging, a street sign, a neon storefront sign, and an illustrated poster.

Abstract

Foreign-language texts are all around us—on signs, menus, labels, and packaging—and camera-based translation tools now let us read it. However, these tools render translations as plain overlays: the cursive on a cafe menu and the neon of a storefront sign both arrive in the same dull, default font. Typography, color, and layout carry a message's formality, tone, and identity—and sometimes a bit of delight worth saving. We introduce *TranslateAR*, a mobile AR prototype that translates text in the scene while preserving its style, regenerating the surface itself so the translation keeps the look of the original. Doing this in a live camera view is non-trivial: text must be detected and stabilized across frames, localized on a physical surface in 3D, and rectified before it can be convincingly regenerated. And because a faithful patch takes seconds to produce, we pair an immediate on-device translation with a slower generative pass. In this demo paper, we describe our initial prototype and the questions it raises for situated, visually faithful translation.

This work is licensed under a Creative Commons Attribution 4.0 International License.

UIST Adjunct '26, Detroit, MI, USA

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2855-6/26/11
https://doi.org/10.1145/3830397.3842478

CCS Concepts

• **Human-centered computing** → **Mixed / augmented reality**; *Natural language interfaces.*

Keywords

Augmented Reality, Text Translation, Style Preservation

ACM Reference Format:
Minbeom Kim, Jason Kim, Davin Win Kyi, Seok-Young Kim, Jaewook Lee, and Jon E. Froehlich. 2026. TranslateAR: Towards Style-Preserving In-Situ Text Translation in Augmented Reality. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830397.3842478

1 Introduction & Related Work

We constantly encounter foreign-language text, from a bag of chips in a grocery store to a sign at a tourist destination. Existing camera-based tools such as *Google Lens* [5], *Apple Live Translate* [2], and *Microsoft Translator* [11] can translate this text, but render the result as a floating overlay in a dull, default font. Little of the original delight survives—for example, the cursive handwriting on a cafe menu is turned into plain text—and the more of the world we translate this way, the more cluttered and less pleasant it becomes [7]. Yet typography, color, and layout carry a message's formality, tone,

emphasis, and identity [9, 18]; stripping text from its visual form preserves *what* it says while discarding *how* it is meant to be read.

Prior work has pursued situated translation and text-style preservation largely separately. In HCI, augmented reality has brought translation and language support into the user's view—from head-mounted text translation via eye gaze [16] to context-aware vocabulary prompts for language learning [6, 10]—yet none preserve the appearance of existing text when translating it. In computer vision, style-preserving methods edit and translate scene text, re-rendering font, position, and background via scene-text synthesis and vision-language models [1, 3, 8, 12–14, 17, 19, 20], but operate on 2D images or pre-recorded video. Most recently, *FontCraft* makes font style designable using generative AI [15], though for authoring new typefaces rather than translating text in a scene. To our knowledge, no prior system places style-preserving translations in-situ, anchored to a physical surface in a live AR view.

We designed and built *TranslateAR*, an iOS prototype that replaces real-world text with style-preserving translations, rendered in place on a physical surface (Figure 1). To achieve this, TranslateAR detects and stabilizes text across jittering frames, localizes it on a surface in 3D, and rectifies the region before regenerating it using *Nano Banana 2* [4]. There is also a tension between fidelity and responsiveness: a generative model can reproduce the original convincingly, but takes seconds—slow for a live camera view. TranslateAR decouples the two, pairing an immediate on-device translation with a slower generative pass that regenerates the surface in the target language and anchors it to the scene, so meaning arrives quickly and visual fidelity follows (Figure 2).

In this demonstration paper, we propose style-preserving, surface-anchored translation as a way to keep real-world text not only legible across languages, but as delightful as it was designed to be. For the demo, attendees can point an iPhone at a table of objects, select a region, freeze the surface, and watch the translation render back onto it, faithful to the original in both meaning and look.

2 System Implementation

We implemented TranslateAR as an iOS AR prototype (commodity iPhone 14 Pro in our implementation) built on *ARKit*[1], *Vision*[2], *MLKit*[3], and generative image editing. Users scan their surroundings, select a detected text region, freeze a surface patch, and generate an in-place translated replacement. To stay responsive despite slow generation, TranslateAR separates immediate semantic understanding from higher-fidelity visual replacement through two complementary tracks: a fast semantic translation track, and an asynchronous style-preserving track (Figure 2).

Text detection and tracking. TranslateAR runs Vision OCR on the live camera feed to detect text regions, estimate bounding boxes, and group observations when multiple lines appear together, using confidence thresholds to filter low-confidence detections. Because OCR fluctuates across frames, the system tracks regions over time—matching them across frames, smoothing bounding boxes, and locking onto text once the same content is recognized consistently—reducing jitter and keeping the interface stable while the user selects

Figure 2: TranslateAR system pipeline. Tapping or dragging over a text region localizes it in 3D from RGB camera, LiDAR, ARKit tracking, and plane detection, yielding a fast semantic translation (Track A). Freezing the region then triggers an asynchronous style-preserving track (Track B) that rectifies, translates, and regenerates the patch before rendering it back.

and translates. Once a region locks, TranslateAR passes the recognized text to on-device MLKit translation and displays the result in an on-screen panel, giving immediate access to meaning while the style-preserving patch renders (Track A, Figure 2).

Surface localization. To render translations in the physical world, TranslateAR estimates where the selected text lies in 3D space, using ARKit depth sensing, raycasting, plane detection, and scene understanding to fit a surface patch around it. The user can freeze this patch so that it stays anchored to the real-world surface as the camera moves.

Style-preserving translation. Once a patch is frozen, TranslateAR crops and rectifies the selected region, then constructs an image-editing prompt from the OCR results and on-device translation hints. A generative image model (*Nano Banana 2* [4]) replaces the source text with the target-language translation while preserving background, layout, perspective, lighting, color, and approximate typographic style, rendering the result back onto the tracked AR surface (≈ 7-9 seconds; Track B, Figure 2). We chose Nano Banana 2 for its accuracy and generation speed in our testing; this step dominates latency, which should fall as such models improve.

3 Discussion and Conclusion

TranslateAR is an early prototype, but it points toward situated translation that preserves not just what text says, but how it looks.

Preserving style may mean changing it. A cursive, handwritten English menu has no perfect Korean equivalent. Preserving typography across writing systems is thus underdefined—the system can copy the source's weight, color, and layout, but the connotations those choices carry are language-specific. Faithful style transfer may sometimes require *changing* the style, or adapting it to the target language, to actually preserve the meaning.

Fidelity is a double-edged sword. A style-preserving patch is discreet—it looks like the object itself, which makes a wrong translation dangerous, especially on AR glasses. The better TranslateAR gets, the less a user can tell when it is wrong, motivating confidence indicators and easy access to the original text.

Not all text needs to be translated. In Figure 1, *Kellogg's* is transliterated rather than translated, preserving the brand as

[1]https://developer.apple.com/documentation/arkit
[2]https://developer.apple.com/documentation/vision
[3]https://developers.google.com/ml-kit

a sound, while *FIFA* is not translated. Names, logos, and culturally specific terms may need transliteration. Deciding which—and making that decision legible to the user—remains open.

We hope TranslateAR motivates situated translation systems that move beyond detached overlays and preserve not just what text says, but how it was meant to be read.

References

[1] Jaided AI. 2026. EasyOCR. https://github.com/JaidedAI/EasyOCR. Accessed: 2026-05-29.

[2] Apple Inc. 2026. Translate with the Camera View on iPhone. https://support.apple.com/guide/iphone/translate-with-the-camera-view-iphea8b95631/ios. Apple iPhone User Guide. Accessed: 2026-05-28.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).

[4] Gemini Team. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805* (2023).

[5] Google. 2026. Google Translate. https://play.google.com/store/apps/details?id=com.google.android.apps.translate. Google Play listing. Accessed: 2026-05-28.

[6] Adam Ibrahim, Brandon Huynh, Jonathan Downey, Tobias Höllerer, Dorothy Chun, and John O'donovan. 2018. Arbis pictus: A study of vocabulary learning with augmented reality. *IEEE transactions on visualization and computer graphics* 24, 11 (2018), 2867–2874.

[7] Simon Julier, Marco Lanzagorta, Yohan Baillot, Lawrence Rosenblum, Steven Feiner, Tobias Hollerer, and Sabrina Sestito. 2000. Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*. IEEE, 3–11.

[8] Hour Kaing, Jiannan Mao, Makoto Morishita, Fei Huang, Tadashi Nomoto, Masao Utiyama, and Eiichiro Sumita. 2025. ImageTra: Real-Time Translation for Texts in Image and Video. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*. doi:10.18653/v1/2025.ijcnlp-demo.1

[9] Beth E Koch. 2012. Emotion in typographic design: An empirical examination. *Visible Language* 46, 3 (2012), 206–227.

[10] Jaewook Lee, Sieun Kim, Minji Park, Catherine L Rasgaitis, and Jon E. Froehlich. 2024. Embodied AR Language Learning Through Everyday Object Interactions: A Demonstration of EARLL. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST Adjunct '24)*. Association for Computing Machinery, New York, NY, USA, Article 52, 3 pages. doi:10.1145/3672539.3686746

[11] Microsoft. 2026. Microsoft Translator App Features. https://www.microsoft.com/en-us/translator/apps/features/. Accessed: 2026-05-28.

[12] Zhipeng Qian, Pei Zhang, Baosong Yang, Kai Fan, Yiwei Ma, Derek F. Wong, Xiaoshuai Sun, and Rongrong Ji. 2024. AnyTrans: Translate AnyText in the Image with Large Scale Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics.

[13] Qwen Team. 2024. Qwen1.5-7B-Chat. https://huggingface.co/Qwen/Qwen1.5-7B-Chat. Accessed: 2026-05-29.

[14] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. 2020. STEFANN: Scene Text Editor Using Font Adaptive Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Yuki Tatsukawa, I-Chao Shen, Mustafa Doga Dogan, Anran Qi, Yuki Koyama, Ariel Shamir, and Takeo Igarashi. 2025. FontCraft: Multimodal Font Design Using Interactive Bayesian Optimization. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–14.

[16] Takumi Toyama, Daniel Sonntag, Andreas Dengel, Takahiro Matsuda, Masakazu Iwamura, and Koichi Kise. 2014. A mixed reality head-mounted text translation system using eye gaze input. In *Proceedings of the 19th international conference on Intelligent User Interfaces*. 329–334.

[17] Shreyas Vaidya, Arvind Kumar Sharma, Prajwal Gatti, and Anand Mishra. 2024. Show Me the World in My Language: Establishing the First Baseline for Scene-Text to Scene-Text Translation. In *International Conference on Pattern Recognition (ICPR)*. 312–328.

[18] Theo Van Leeuwen. 2006. Towards a semiotics of typography. *Information design journal* 14, 2 (2006), 139–155.

[19] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. 2019. Editing Text in the Wild. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1500–1508.

[20] Qiangpeng Yang, Hongsheng Jin, Jun Huang, and Wei Lin. 2020. SwapText: Image Based Texts Transfer in Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14700–14709.