A User-Guided and Local Motion-Adaptive Framework for Virtual Product Placement in Video

Tianwen Zhang, Ju-Won Seo, Kang-Min Kim, and Keunsoo Ko
The Catholic University of Korea
E-mail: {twzhang, tjwndnjs, kangmin89, ksko}@catholic.ac.kr

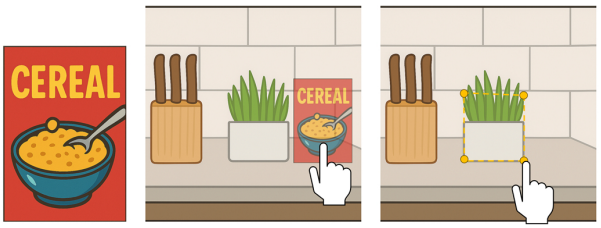
***Abstract*—Virtual product placement (VPP) aims to seamlessly insert a product into a pre-recorded video. It is gaining attention as a flexible and scalable strategy for digital advertising, where user preferences and subjective choices should be considered. However, the conventional VPP methods adopt fully automatic insertion frameworks without user involvement, which limits their adaptability to subjective scenarios. To address this limitation, we proposed a user-guided and local motion-adaptive framework, called ULM-VPP, for video VPPs. Given a video and a virtual product, the proposed system obtains an initial target region for the product placement and nearby keypoints in the first frame for minimal user interaction. For each subsequent frame, both the target region and virtual product are realigned based on local motion inferred from the keypoints. Finally, the virtual product is inserted seamlessly into the updated region over all frames while maintaining temporal consistency. Experiments across various video sequences demonstrate that ULM-VPP produces geometrically stable and visually coherent results. The source code is available at https://github.com/halikes/ULM-VPP**

I. Introduction

Virtual product placement (VPP) is a post-production technique that seamlessly inserts objects into pre-recorded videos, serving as a flexible and scalable strategy for digital advertising. The demand for VPP has increased with the rise of personalized and short-form content. Recently, with the great success of generative models [4], [5], several VPP techniques [7], [16] have been proposed that enable realistic and automated product insertions. However, since these methods are designed for still images, they often struggle to maintain geometric consistency and temporal stability across consecutive video frames.

To address this limitation, several video VPP methods [2], [1], [8] have been explored by modeling camera motion. For instance, Bhargavi *et al.* [2] estimate a homography between frames by matching keypoints, which typically correspond to corners or other salient points. In [8], [1], optical flow is employed to capture pixel-wise motion. These methods rely on globally distributed matching points to estimate motion between adjacent frames and align the virtual product accordingly to maintain temporal stability.

While the global motion estimation has shown promise in aligning inserted objects across frames, it can be sensitive to independently moving objects or complex background elements commonly observed in real-world scenes. Additionally, in practical scenarios, a virtual product is usually placed within a confined region to avoid disrupting the viewer's experience. This highlights the need to focus on local motion estimation, which can provide more reliable alignment within the designated target region.



(a) Product (b) Target Region (c) Keypoints

Fig. 1: With minimal user interaction, the target region and nearby keypoints are specified by a user only in the first frame.

Moreover, in digital advertising, the placement of a virtual product is highly subjective, influenced by aesthetic harmony, brand visibility, and narrative focus, often requiring human judgment. Thus, automatically identifying a suitable region is difficult. To address these limitations, we introduce minimal user interaction in the first frame to specify the target region for produce placement and nearby keypoints as shown in Fig. 1.

In this paper, we propose a user-guided and local motion-adaptive framework, called ULM-VPP, which combines minimal user interaction with local motion modeling for robust and temporally consistent product insertion in video. The user specifies the target placement region and a small set of neighboring keypoints in the first frame. For each subsequent frame, we estimate the local motion of the target region based on specified keypoints and use it to compute a homography transformation that aligns the target region and virtual product. The virtual product is then inserted at the updated target region in harmony with its surroundings. Through this user-guided and local motion-adaptive design, ULM-VPP achieves both personalized VPP and seamless integration across diverse video scenes. Experiment results demonstrate that the proposed algorithm stably and naturally inserts virtual products into real-world videos.

Our contributions can be summarized as follows.

- We propose a user-guided VPP framework that reflects individual preferences by allowing users to specify a placement region directly.
- We incorporate local motion estimation to track product placement robustly.
- We evaluate the effectiveness of the proposed framework across real-world video sequences.

979-8-3315-7206-8/25/$31.00

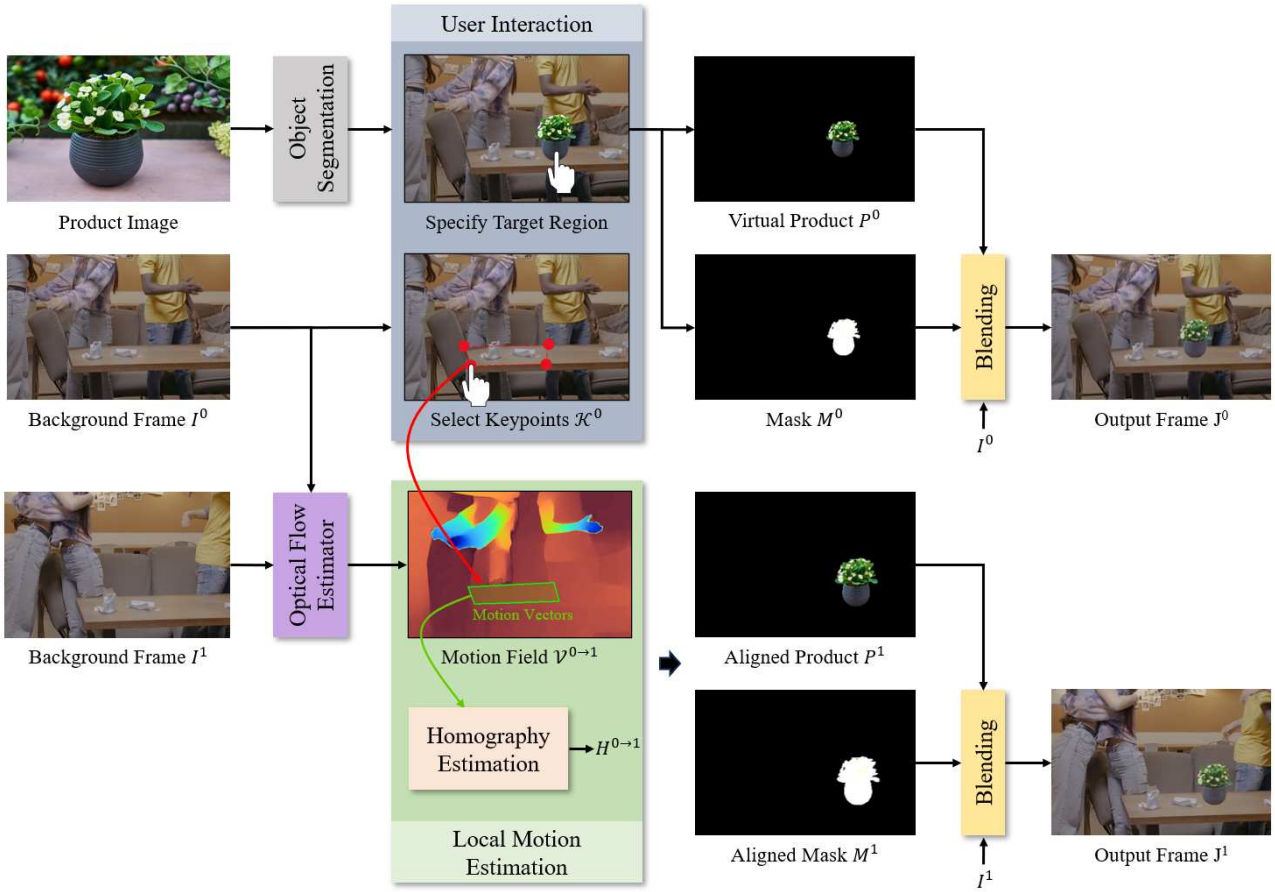


Fig. 2: Overview of the proposed ULM-VPP framework.

The rest of this paper is organized as follows: Section 2 describes the proposed VPP algorithm, and Section 3 discusses experimental results. Section 4 concludes this work.

II. Proposed Algorithm

Fig. 2 shows an overview of the proposed ULM-VPP framework. Given the first background frame I^0 and a virtual product P^0, a user specifies the target region where the product will be placed and a set of neighboring keypoints $\mathcal{K}^0$ near that region. We insert P^0 into I^0 at the specified location using the proposed blending module. Next, we predict a global motion field $\mathcal{V}^{0\to1}$ and then sample motion vectors within the convex region defined by $\mathcal{K}^0$. Based on the sampled vectors, we update the target location and transform P^0 to obtain P^1, which is then blended into I^1. This process, comprising local motion estimation, transformation, and blending, is repeated for all subsequent frames.

A. User Interaction and Initialization

Given an input product image, we extract the product from its background with a semantic segmentation algorithm, SAM [6]. After the user specifies the target location and size at the first frame I^0, we construct a virtual product P^0 and a binary mask M^0, where pixels within the specified region are set to 1 and the rest to 0. This explicit user interaction allows the proposed VPP system to reflect subjective preferences, which are difficult to infer automatically.

In addition to the target region, the user also selects a few keypoints to guide motion estimation. Since it is difficult for the user to select precise keypoints, we refine each user-selected point by replacing it with the nearest keypoint extracted by SIFT [9]. These keypoints $\mathcal{K}^0 = \{\mathbf{k}_1^0, \ldots, \mathbf{k}_N^0\}$ are usually found at salient corners, helping to identify motion stably in consecutive frames.

B. Local Motion Estimation and Transformation

The target region for VPP may change across frames due to camera motion or scene dynamics. The conventional VPP methods [2], [1], [8] address this by estimating a global motion field and applying it to update the target region, transforming the virtual product accordingly. However, this global approximation might not accurately capture the local motion, especially when there are independently moving objects or complex background elements. Moreover, the target region is often located in a low-texture background where accurate motion estimation is difficult. To overcome these limitations,

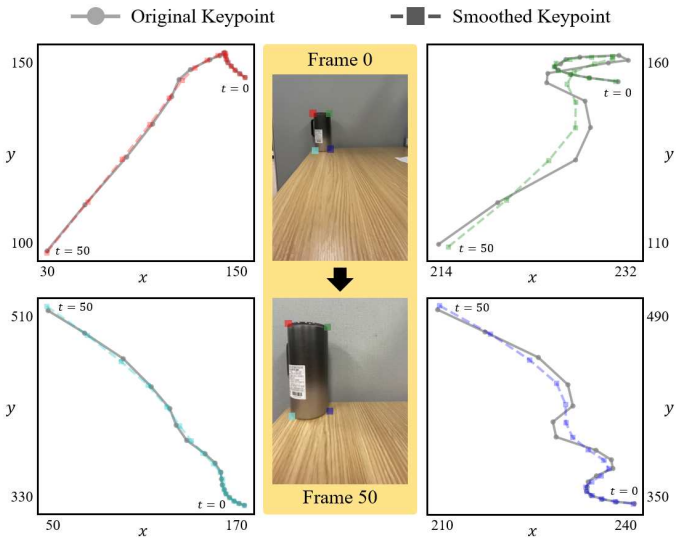


Fig. 3: Visualization of four keypoint trajectories before and after smoothing. The original keypoints (gray lines with circular markers) exhibit jitter and instability, while the smoothed trajectories (colored dashed lines with square markers) show improved temporal stability. The initial and final frames are displayed where each keypoint is annotated with the color corresponding to its trajectory.

we infer the motion of the target region by tracking its neighboring keypoints.

Specifically, we first estimate a global motion field $\mathcal{V}^{t-1\to t}$ between frames I^{t-1} and I^t using an optical flow estimator, RAFT [15]. We sample motion vectors from $\mathcal{V}^{t-1\to t}$ that lie within the convex hull enclosing the keypoints $\mathcal{K}^{t-1}$. Here, each motion vector represents a 2D point correspondence from I^{t-1} to I^t. A homography $H^{t-1\to t}$ is estimated from the inlier correspondences, obtained by applying RANSAC [3] to the sampled motion vectors. This homography, derived from keypoints near the target region, approximates the local planar motion of the target region.

Although this local motion estimation effectively tracks short-term motion, it may be susceptible to noise, non-rigid deformation, and occlusion, leading to artifacts on the inserted product, such as jitter, drift, or geometric distortion. To overcome this issue, we introduce motion smoothing and regularization strategies as a post-processing step.

Let $\mathbf{k}_i^{t-1}$ denote the position of the i-th keypoint at frame $t-1$. We update it with $H^{t-1\to t}$ as

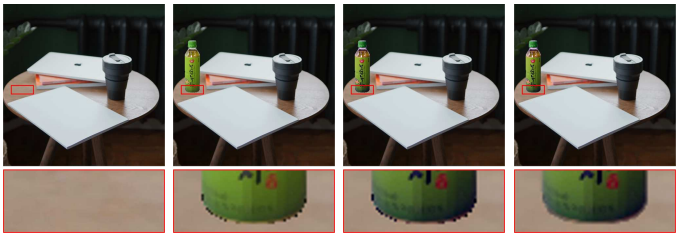
$$\mathbf{k}_i^t = H^{t-1\to t}\mathbf{k}_i^{t-1}. \tag{1}$$

To reduce temporal fluctuations, we smooth the trajectory of keypoints by applying Savitzky-Golay filtering $\mathcal{G}$ [14] by

$$\tilde{\mathbf{k}}_i^t = \mathcal{G}\left(\mathbf{k}_i^{t-w}, \ldots, \mathbf{k}_i^t, \ldots, \mathbf{k}_i^{t+w}\right), \tag{2}$$

where w is empirically set as 21. Finally, we estimate a refined homography $\tilde{H}^{t-1\to t}$ based on the smoothed keypoints.

This temporal smoothing preserves the structural consistency of each keypoint trajectory while effectively suppressing local jitter and irregular motion. Fig. 3 determine that the proposed method yields smoother and more stable trajectories across frames.



(a) Background (b) Alpha Blending (c) Harmonization (d) Boundary Blur

Fig. 4: Blending process of the proposed method. To insert the virtual product into (a) the background frame, we sequentially apply (b) alpha blending as in (3), (c) appearance harmonization as in (4), and (d) boundary blurring.

C. Blending

For frame t, we obtain the virtual product P^t and the binary mask M^t by applying the homography $\tilde{H}^{t-1\to t}$ to P^{t-1} and M^{t-1}, respectively. Then, we insert P^t into I^t using M^t via alpha blending [12] by

$$\hat{I}^t = M^t \otimes P^t + (1 - M^t) \otimes I^t, \tag{3}$$

where $\otimes$ denotes element-wise multiplication.

To ensure a visually coherent and seamless composition with the background, we perform appearance harmonization by applying localized color and luminance adjustments. Let $\Omega^t = \{(x,y) \mid M^t_{(x,y)} > 0\}$ be the insertion region. For all $(x,y) \in \Omega^t$, the refined pixel values $\tilde{I}^t_{(x,y,c)}$ is computed with a balancing factor β as follows:

$$\begin{aligned}\tilde{I}^t_{(x,y,c)} = \beta \cdot \left(\frac{\hat{I}^t_{(x,y,c)} - \mu_s}{\sigma_s} \cdot \sigma_r + \mu_r\right) \\ + (1-\beta)\cdot \hat{I}^t_{(x,y,c)},\end{aligned} \tag{4}$$

where (μ_s, σ_s) and (μ_r, σ_r) are the mean and standard deviation of the virtual product P^t and the background frame I^t, respectively, computed over Ω for each channel c. Here, the blending factor is set to 0.4 empirically.

Finally, we apply Gaussian blurring to the boundary region of the inserted product using a window of size 7 and a standard deviation of 1 to reduce sharp discontinuities and enhance perceptual quality. Fig. 4 shows the output at each stage of the proposed blending process.

This combined strategy of blending, harmonization, and boundary refinement ensures that the inserted object not only maintains geometric stability but also integrates seamlessly into the dynamic background of real-world video.

III. Experiments

We evaluate the performance of the proposed algorithm on several real-world video sequences[1]. Fig 5 shows qualitative

[1]The output videos can be seen in https://halikes.github.io/ULM-VPP/

Fig. 5: Qualitative results of the proposed method on three real-world video sequences.

Fig. 6: Qualitative comparison on three frames.

results on three examples. The inserted virtual objects consistently maintain geometric stability, and they are blended naturally into the dynamic backgrounds.

To analyze the efficacy of each component in ULM-VPP, we compare it with its alternatives as shown in Fig. 6. First, 'No-Align' directly uses the user-provided target region without any alignment. 'LK Flow' estimates optical flow via the Lucas-Kanade method [10] instead of the deep learning-based estimator, RAFT [15]. 'Global Motion' employs global motion fields as done in [2], [1], [8]. In 'Inpainting', the virtual product is inserted using a diffusion-based inpainting model, LDM [13], similar to [1], [16]. As shown in the results, the proposed ULM-VPP yields more temporally stable and perceptually seamless insertion.

TABLE I: User study results, reported as mean ± standard deviation, on stability and naturalness.

Method	**Stability ↑**	**Naturalness ↑**
No-Align	1.8 ± 0.3	1.8 ± 0.4
Global Motion	2.8 ± 0.4	2.7 ± 0.5
LK Flow	1.7 ± 0.3	1.6 ± 0.4
ULM-VPP (Ours)	**4.2** ± **0.5**	**4.3** ± **0.6**

Finally, we conducted a user study to evaluate the quality

of synthesized videos subjectively. Each of the ten participants was asked to assign a score from 1 (bad quality) to 5 (excellent quality) to the VPP results in terms of stability and naturalness on two video sequences. Here, the four methods, No-Align, Global Motion, LK Flow, and ULM-VPP, were validated, and their results were presented in a random order to minimize bias. As shown in Table I, the proposed ULM-VPP achieves the best scores with a large margin. This demonstrates the superior stability and perceptual realism of our framework.

IV. Conclusion

In this paper, we proposed a user-guided and local motion-adaptive framework, called ULM-VPP, for VPP in video. By incorporating minimal user interaction, the proposed method provides a personalized VPP. While conventional approaches rely on global motion estimation, ULM-VPP focuses on local motion guided by user-specified keypoints. Based on local motion, we compute the homography transformation with temporal smoothing and update the target insertion region. We insert a product in the target region using blending approaches, ensuring seamless integration of the virtual product into diverse videos. Experiments validated the effectiveness of ULM-VPP and offer the possibility as a flexible and effective solution for practical VPP applications.

While ULM-VPP provided outstanding performance, the proposed blending strategy may be limited in illumination variations, occlusion, or complex 3D camera movements. As future work, the proposed framework can be extended using recent generative techniques, such as NeRF [11] or diffusion models [5], to better handle realistic and challenging scenarios with complex motion and lighting conditions.

References

[1] C. Bai, Z. Shao, G. Zhang, D. Liang, J. Yang, Z. Zhang, Y. Guo, C. Zhong, Y. Qiu, Z. Wang, et al. Anything in any scene: Photorealistic video object insertion. *arXiv preprint arXiv:2401.17509*, 2024.

[2] D. Bhargavi, K. Sindwani, and S. Gholami. Zero-shot virtual product placement in videos. In *ACM IMX*, 2023.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Trans. Commun.*, 24(6):381–395, 1981.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial Nets. In *NeurIPS*, 2014.

[5] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[6] 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. In *IEEE ICCV*, 2023.

[7] R. Liang, Z. Gojcic, M. Nimier-David, D. Acuna, N. Vijaykumar, S. Fidler, and Z. Wang. Photorealistic object insertion with diffusion-guided inverse rendering. In *ECCV*, 2024.

[8] Z. Liu, J. Yang, M. Gao, and F. Zheng. Place anything into any video. *IJCAI*, 2024.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60:91–110, 2004.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[11] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[12] T. Porter and T. Duff. Compositing digital images. In *SIGGRAPH*, 1984.

[13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[14] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, 36(8):1627–1639, 1964.

[15] Z. Teed and J. Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[16] Y. Tewel, R. Gal, D. Samuel, Y. Atzmon, L. Wolf, and G. Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. In *ICLR*, 2025.