An instant and accessible AR canvas in iPad ecosystem

Zhaohan(Chester) Pan
Stanford University
Stanford, USA
yoas0b1@stanford.edu

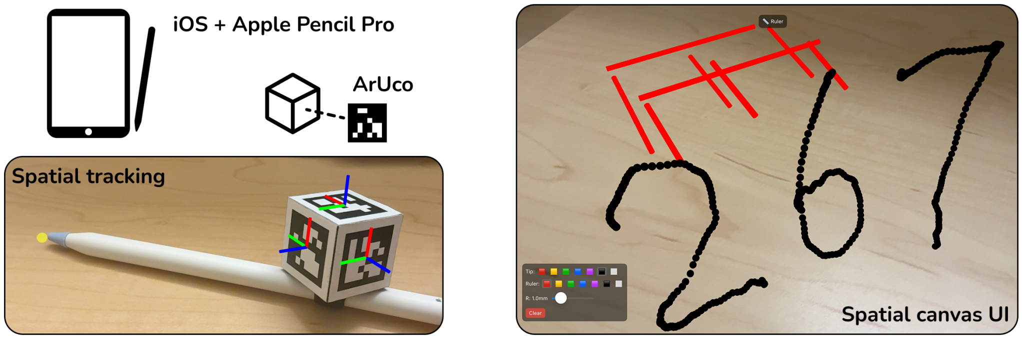

Figure 1: Spatial tracking implementation and XR sketching UI demonstration of our experiential prototype.

1 Introduction

Augmented Reality(AR) introduces a new visual interface for displaying virtual objects in the physical world. Video-based AR, the current mainstream trend, offers a relatively mature development environment and brings scalability to applications.

Spatial sketching is a popular topic within AR, which is already developed in HUD-based AR environments, such as drawing with controllers on Meta Quest or freehand gesture sketching on Apple Vision Pro. In contrast, MAR (Mobile Augmented Reality) platforms are less commonly considered for such tasks. This is primarily due to MAR's reliance on screen-based interaction and the limited field of view (FOV) of mobile devices. However, tablets and other MAR devices are already widely adopted as sketching tools, especially like the iPad + Apple Pencil ecosystem. Given this prevalence, deploying spatial sketching tools (especially those that are software only or paired with lightweight, scalable hardware) holds significant value.

2 Related Works

There are already a few prior studies on spatial sketching in MAR. For instance, Mobi3DSketch[2] fixes the "pen tip" in camera space, using MAR as both the display and pen controller for single-handed sketching. Portalware[3] connects a stereo camera to a smartphone to enable bimanual freehand sketching in MAR. However, the former lacks robust manipulation freedom for global scene vision, while the latter requires a complex hardware setup. ARPen[4] offers a more balanced solution, employing a bimanual interaction style through a 3D-printed pen and ArUco-based tracking [1]. Although ArUco may not achieve the same accuracy as stereo cameras, its scalability and accessibility are unmatched.

3 Scope of Work

ARPen was developed early on and featured a custom pen with Bluetooth and button controls via PCB. This design could now be replaced by Apple's new API of Apple Pencil Pro announced in WWDC 24, which supports more gesture input and has inherent communication with iPad. Specifically, `doubleTap` and `squeeze` gestures are interactable during non-hover or non-contact operations, which enables two input interfaces on mid-air & contactless sketching. With these features, we can focus solely on solving the tracking problem.

The primary goal of this project is to adopt the iPad + Apple Pencil ecosystem as a mature sketching platform, while introducing minimal & accessible add-ons to produce a preliminary XR sketch experiential prototype.

4 Implementation

The experiential prototype is mainly composed of two parts: (1) a tracker attached to the Apple Pencil Pro, and (2) an iOS AR app for iPad serving as a tracking camera and a user interface simultaneously.

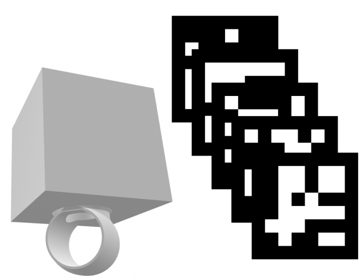

Figure 2: Tracking cube model and ArUco markers used.

4.1 Tracking Cube Fabrication

The tracking component is modeled in a cube shape with a ring fixing on the Apple Pencil (**Figure 2**). We left a hole inside the cube / on one side of the ring to attach a magnet to the tracker. This design is inspired by utilizing the magnet attachment between the Apple Pencil and iPad: we design the cube to be easily located at exactly the rear magnet, while the magnet can also contribute to stabilizing the tracker with the pencil. These are critical for the tracking procedure later. The ArUco dictionary marker selected here is `DICT_ARUCO_MIP_36h12`, with IDs ranging from 2 to 6 for 5 fully visible square surfaces, respectively.

4.2 iOS App development

Our iOS app is developed under iOS 17.5 with essential packages like ARKit and SwiftUI to manage the AR session and interface. The app is experimented on an iPad Pro M4 model.

4.2.1 Pose Tracking. The tracking procedure is shown in **Figure 3** (with an algorithm version in the appendix). At a high level, we first leveraged the OpenCV framework to accomplish basic pose tracking of markers, then implemented several frames-window average and one-euro filter to stabilize and smooth the tracking-to-render pipeline.

OpenCV 4.10.2 framework with contrib package is added into the project dependency. AR session delegated by ARKit, where the default frame buffer is `1920*1080`. We downscale the frame buffer to `480*270` to ensure the real-time transmission on an M4 model. The frame buffer is then processed with OpenCV tracking, in principle solving $0 \leq n \leq 3$ PnP problems (from detecting no markers to observing at most 3 faces).

Each marker pose is pre-processed with an average frames-window filter and a one-Euro filter, then the 3D position vector of the pen tip is computed by applying translation to the pose matrix of the marker. Note that each marker has its own relative pose toward the pen tip, so that each one should be translated by cases upon detection (cases are in fact similar by geometric symmetry). The $0 \leq n \leq 3$ tips are simply averaged, post-processed again with another frames-window filter and a one-Euro filter, and then the resulting 3D vector is what the virtual tip's position state in camera space.

4.2.2 Gesture Interface. With the captured and computed virtual tip state, we render a dynamic sphere on the newest tip state in the camera space. This is purposed for providing a visual content of the current virtual tip for the user in case of any position fluctuation, noise, or delay.

The Apple Pencil Pro Gesture API is the main interface for controlling the virtual drawing process. We introduce two modes for demonstration purposes in this App: regular mode and ruler mode (**Figure 4**). The `doubleTap` gesture is designed as the switch for the two modes, where the `squeeze` gesture is designed as the on-off switch for each mode.

The regular mode is basically freehand drawing, where the stroke is recorded as the trajectories of virtual tip movement as turning on. For simplicity, the strokes are just the sequential tip spheres that are transformed back to world space and recorded there. The stroke is visibly scattered during faster drawing since it is recorded by frame, but is considered good for concept demo purposes.

The ruler mode is selecting two spatial points using the on-off switch and making a connective stroke between them. The on-squeeze will first record the current virtual tip state as a starting point and transform it to the world space. During sequential

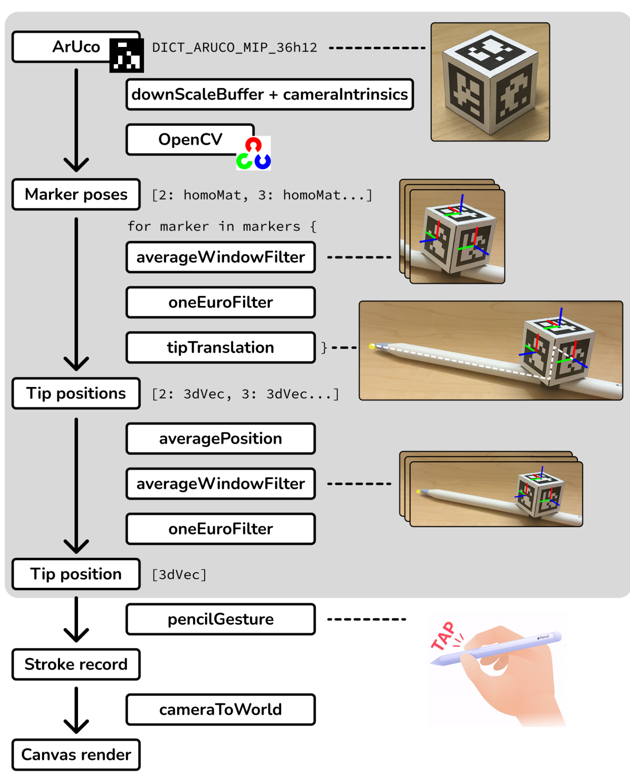

Figure 3: Pipeline of the sketching App. The shaded region denote specifically the tracking implementation.

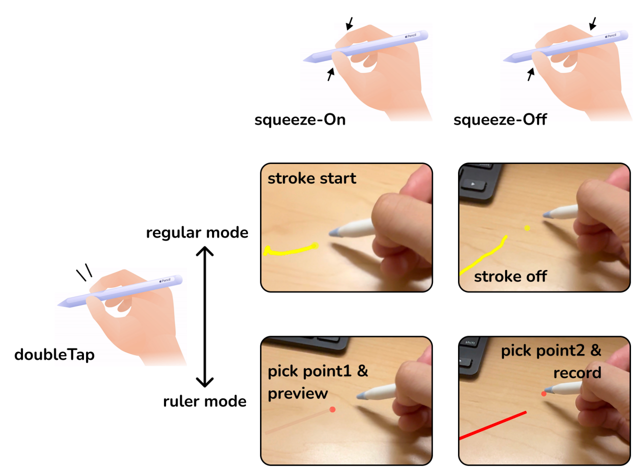

Figure 4: Demonstration of doubleTap and squeeze gestures implementation.

on-frames, the new tip state will be sent to world space dynamically, and then with that value, a high transparency cylinder is rendered as the ruler stroke preview. When the off-switch is detected, the previous-frame cylinder will be turned into a non-transparent straight stroke, and the start point will be cleared for another iteration.

These two designed features are considered representatives for preliminary sketch experiences. In fact, there are some memory issues with too much entity storage and rendering, especially in the regular mode where strokes are scattered spheres. The implementation of those methods in vector illustration is expected to be helpful in our cases, but they are out of the scope of this work.

5 Spatial Sketching Scene and User Interface

Figure 5 demonstrates some sketches created by using our prototype. It can be widely used in scenarios like (1) mid-air spatial scene creation, (2) object tracing, or (3) object virtual decorative content augmentation.

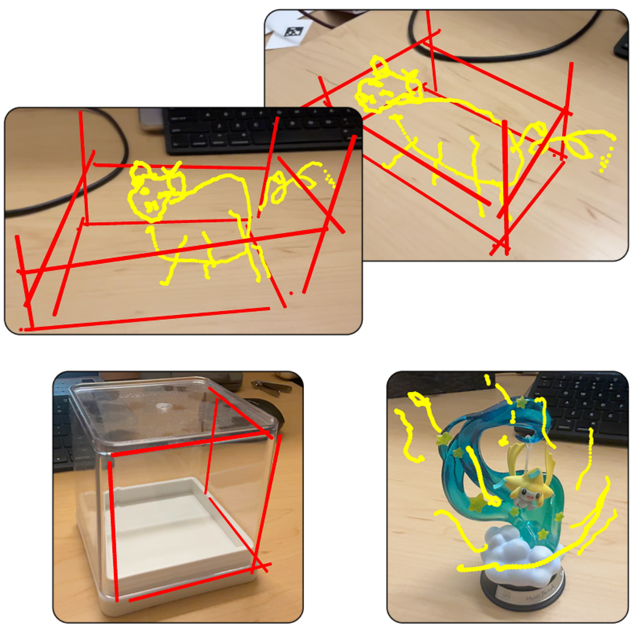

Figure 5: Demonstration of sketches created by the prototype.

6 Evaluation

The main challenge and possible improvement within this system is obtaining accurate tracking while rendering in real-time, and these two are always hard to balance. Although not measured precisely and quantitatively, there is observable latency in the virtual tip movement regarding the physical pencil, with small uncertain fluctuations in position. In fact, the tip position has no latency without filters, but the fluctuation was extremely uncontrollable. This is because the distance between a marker and the tip is approximately 100 mm, that any small movement in the pose will result in an amplified fluctuation. Therefore, the filters came in place and they significantly resolved this problem, while the trade-off is latency.

The other major challenge within the tracking pipeline is the flipping fluctuation of a single marker when detection is poor. This is likely observable when the markers are tilted or under poor lighting conditions, where the uncertainty of PnP problem's surface up-direction is high. In this case, average calculation is influenced by the outlier and induces shaking on tip tracking.

In general, more robust tracking algorithm, stronger filters, stronger tracking models, and a cleaner physical sketching scene would help in improving the system.

7 Conclusion

Our experiential prototype truly achieved an accessible and DIY-friendly design in delivering XR sketching experiences by adopting resources of iPad and Apple Pencil Pro ecosystem. This design is expected to facilitate the spatial drawing venues and contribute to Human-Computer Interaction research.

References

[1] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[2] Kin Chung Kwan and Hongbo Fu. 2019. Mobi3DSketch: 3D Sketching in Mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300406

[3] Jing Qian, Tongyu Zhou, Meredith Young-Ng, Jiaju Ma, Angel Cheung, Xiangyu Li, Ian Gonsher, and Jeff Huang. 2021. Portalware: Exploring Free-Hand AR Drawing with a Dual-Display Smartphone-Wearable Paradigm. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference* (Virtual Event, USA) *(DIS '21)*. Association for Computing Machinery, New York, NY, USA, 205–219. https://doi.org/10.1145/3461778.3462098

[4] Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. 2019. ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300849

A Tracking Algorithm Pseudo Procedure

Algorithm 1 Tracking procedures in AR session

```
procedure TRAKCING(frame)

    // raw PnP solving
    img ⟵ DOWNSCALE(frame)
    pose ⟵ OPENCV(img)
    real.pose ⟵ camera.intrinsics, pose
    [marker.infos].append(real.pose)

    // marker vertex refining
    for info ∈ marker.infos do
        f1 ⟵ AVGWINDOWFILTER(info)
        f2 ⟵ 1€-FILTER(f1)
        tip ⟵ TIPTRANSLATION(f2)
        [tip.infos].append(tip)
    end for

    // tip smoothing & stabilizing
    avg.tip ⟵ AVGPOSITION([tip.infos])
    f3 ⟵ AVGWINDOWFILTER(avg.tip)
    f4 ⟵ 1€-FILTER(f3)
    final.tip ⟵ f4
end procedure
```