Marker based furniture Visualization

Augmented Reality – Winter term 24/25

By Sanchuan, Yichao and Ramon

Overview

Motivation

Methodology

Used Software

Implementation

Results

Motivation

The inspiration is from this trick I used to design my room,
IT WORKS,

After the **measurement**, I can have "**the models**" I need, and all left is to move and play around, also a brain to do a NAVIDA part for the view. **so** I guess with AR, it can be better for the visualization, to talk with others

Methodology

MindAR only recognizes .mind files based on feature point matching for image recognition.

The process includes:

1. Feature Extraction: Extracting key points from a target image and storing them in a .mind file.

2. Feature Matching: Matching real-time camera image features with the .mind file.

3. Pose Estimation: Using PnP or RANSAC algorithms to compute the 3D position.

4. Rendering AR Content: Overlaying 3D models or animations on the detected target.
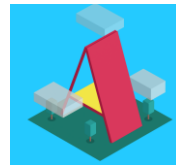
Used Software

Our starting point:

opensource web augmented reality library

HTML-Editor with Live-Server-Plugin
for favourite Browser

web framework for building VR experience
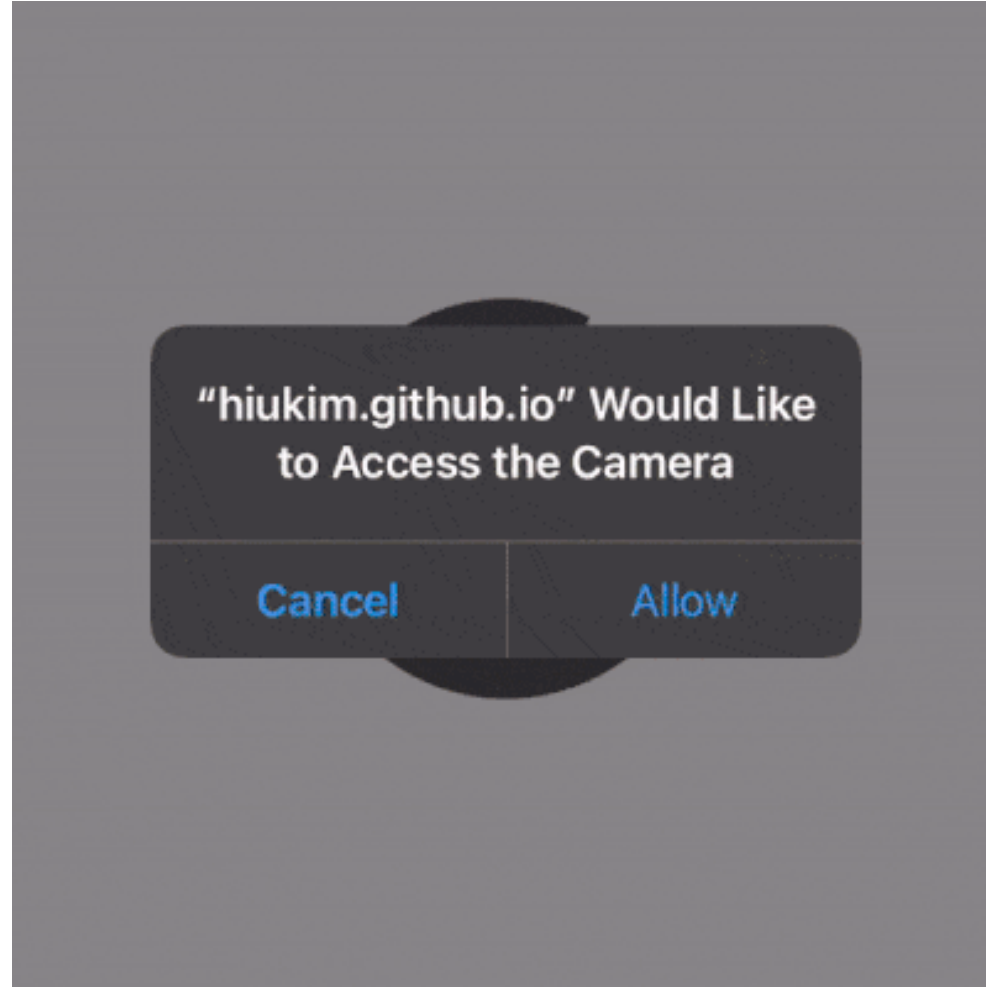
| Motivation | Methodology | **Used Software** | Implementation | Results | Limitations |

Used Software

Image Tracking Example

Used Software

Face Tracking Example

Model Overlap allowed

Video Source: https://hiukim.github.io/mind-ar-js-doc/face-tracking-quick-start/overview
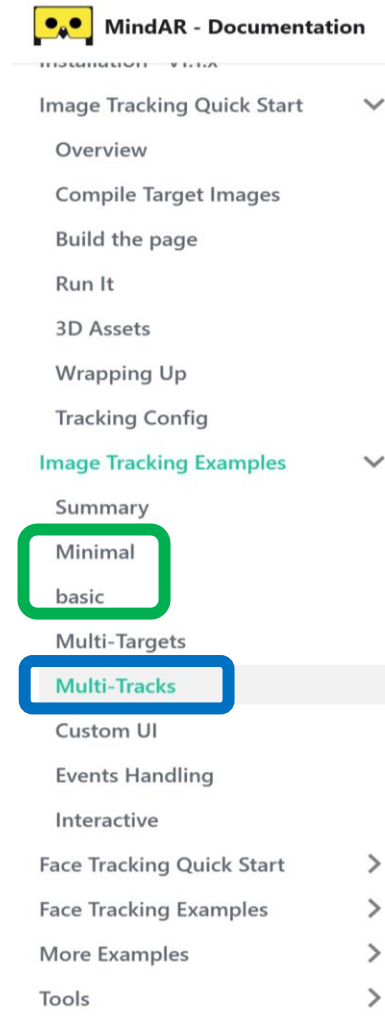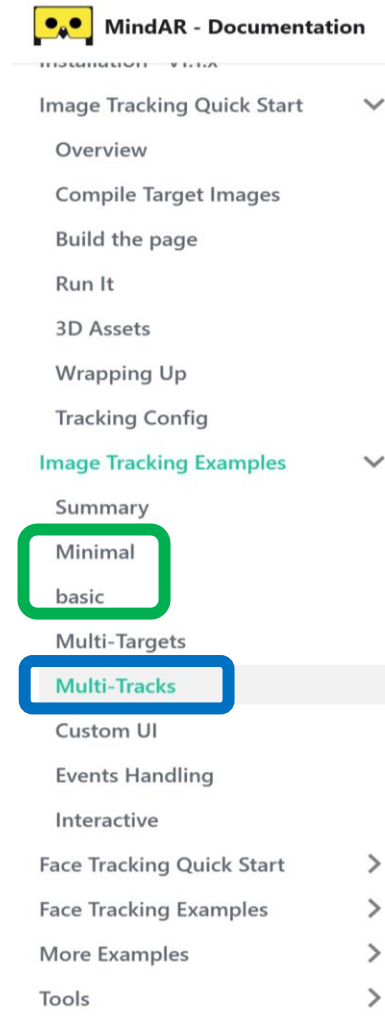
Used Software

Todo:

- Get simple Examples to work

- Expand on Multi-Tracks Example

- Add custom models and markers

Image Source: https://hiukim.github.io/mind-ar-js-doc/examples/multi-tracks

Used Software

Todo:

- Get simple Examples to work

- Expand on Multi-Tracks Example

- Add custom models and markers

Image Source: https://hiukim.github.io/mind-ar-js-doc/examples/multi-tracks

Implementation

1. Load libraries
2. Load .mind File containing marker info
3. Load custom 3D assets as .glb files
4. Define camera object for image acquisition
5. Assign 3D assets to marker counterpart


```
main_code_demonstration.html

1    <!-- AR Application to track markers to visualize different furniture positions -->
2    <html>
3        <head>
4            <meta name="viewport" content="width=device-width, initial-scale=1" />
5            <!-- Load the necessary files over the internet -->              ①
6            <script src="./mindar-image.prod-1.2.5.js"></script>
7            <script src="https://aframe.io/releases/1.5.0/aframe.min.js"></script>
8            <script src="./mindar-image-aframe.prod-1.2.5.js"></script>
9        </head>
10       <body>                                                                ②
11           <!-- Main part of the application. Image Target .mind file -->
12           <a-scene mindar-image="imageTargetSrc: targets.mind; maxTrack: 4" color-space="sRGB" renderer="colorManagement: true,
             physicallyCorrectLights" vr-mode-ui="enabled: false" device-orientation-permission-ui="enabled: false filterMinCF:0.1
             ; filterBeta: 10">
13               <!-- 2D/3D Assets to display over the target image, if it is tracked -->
14               <a-assets>                                                    ③
15                   <a-asset-item id="blenderCube" src="./assets/blender_cube.glb"></a-asset-item>
16                   <a-asset-item id="simple_table" src="./assets/simle_table.glb"></a-asset-item>
17               </a-assets>
18
19               <!-- Set camera position to origin of target image -->        ④
20               <a-camera position="0 0 0" look-controls="enabled: false"></a-camera>
21                                                                             ⑤
22               <!-- Target images to be tracked and assets to be displayed with it. Rotation in degree.-->
23               <a-entity mindar-image-target="targetIndex: 0">
24                   <a-plane color="blue" position="0 0 0" height="0.552" width="1" rotation="0 0 0"></a-plane>
25                   <a-gltf-model rotation="0 0 0" position="0 0 0.1" scale="0.5 0.5 0.5" src="#blenderCube"></a-gltf-model>
26               </a-entity>
27               <a-entity mindar-image-target="targetIndex: 1">
28                   <a-gltf-model rotation="0 0 0 " position="0 0 0.1" scale="0.2 0.2 0.2" src="#simple_table"></a-gltf-model>
29               </a-entity>
30           </a-scene>
31       </body>
32   </html>
```

Implementation

Implementation

1.

2.

3.

4.

5.

Implementation

- Used Markers: From simple online Generator

- Used 3D Assets:

  - Simple furniture geometries → downloaded

  - Custom scanned furniture models …
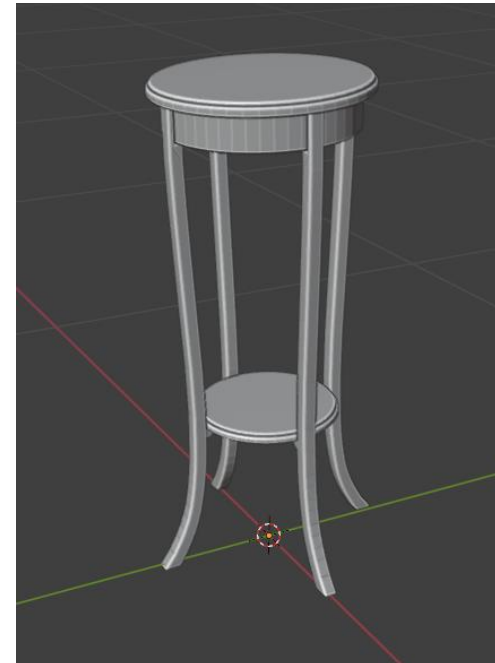
Results

Limitations

- We used **pure photos** to build the models

- The only problem is just the model is always too big, I can even **not** directly quickly see it on pc.

- So…
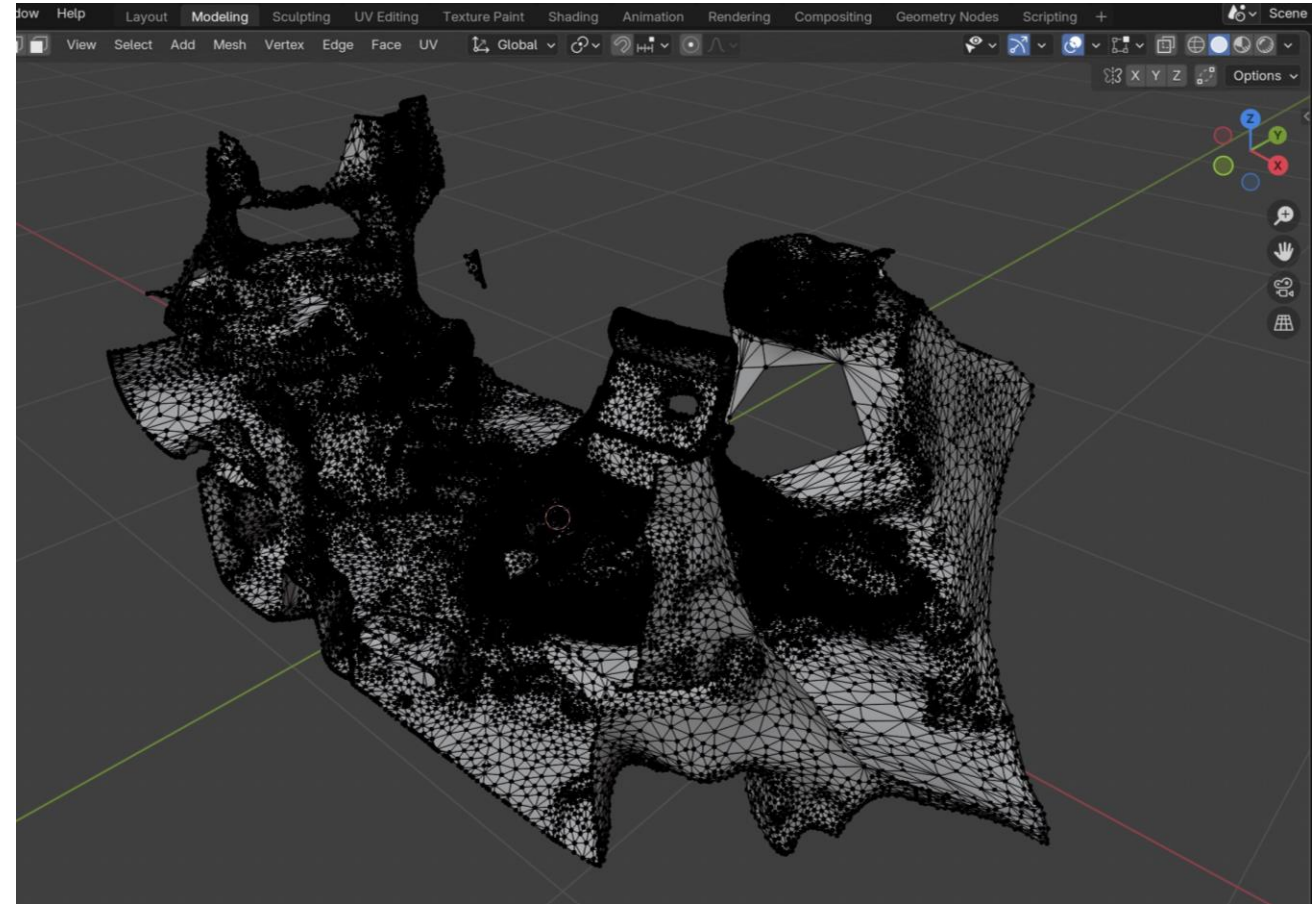
Limitations

- We edited the models in Blender to cut unneeded parts away and thin out the point clouds

- The models look cool, but can not <u>run on **my** computer</u>
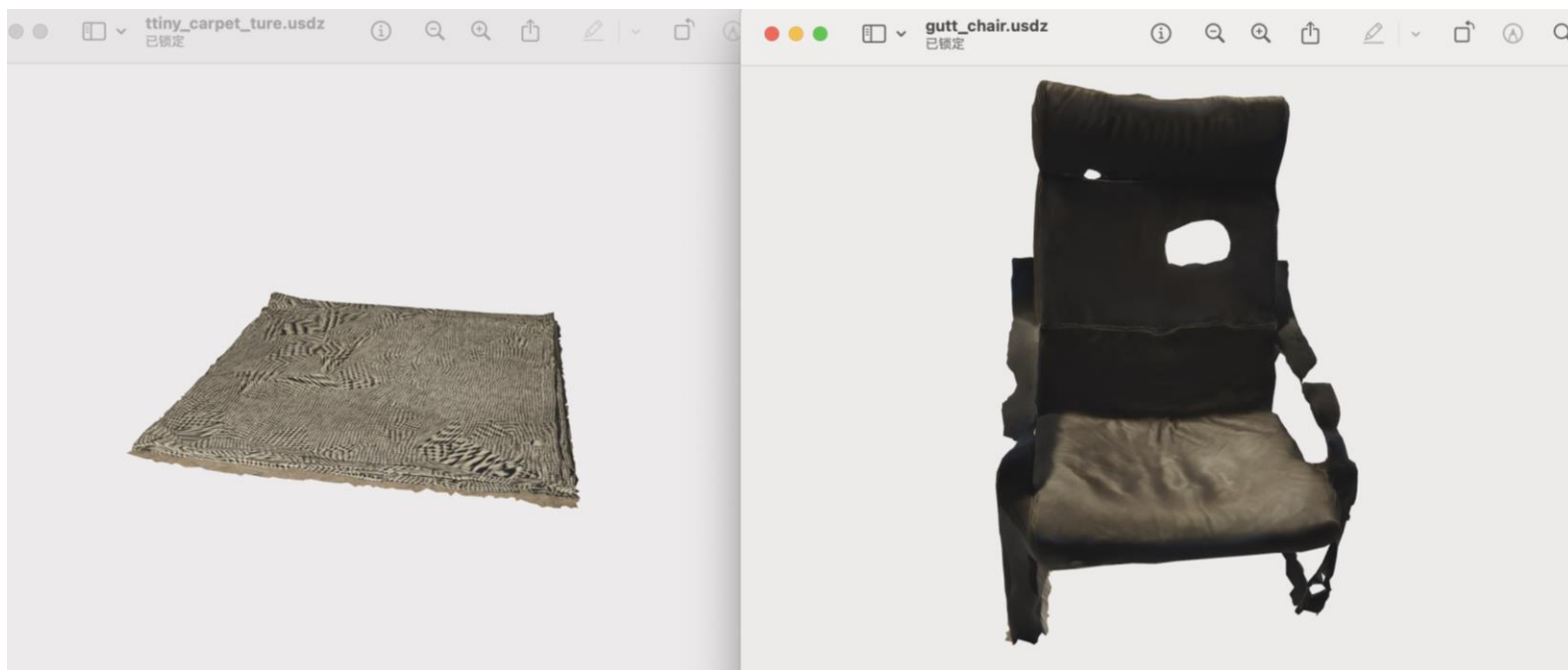
Limitations

So what **we find more,**
is that we always adjust our models into a
tiny size, adjusting **the scaling** to have the
view all within the paper

As the model, we used is always the real size, so
why not just keep it, we can play around in the
whole room

Limitations

THE GOOD PART WE FOUND IT, BAD PART IT DOESN'T WORK WELL

So, as we can see, WE HAVE **A Functioning** <u>**Workflow.**</u>

AND NEXT TIME <u>**WHEN A** store</u> wants to sell something that user may hard to move, **furniture**, or **just something** want to **show to the customer ,**

All they need is just to **attach the package** with <u>a marker .</u>

<u>**→ Okay, run** it</u>

Limitations

- Mistracking
- Limitation of **resolution** of camera
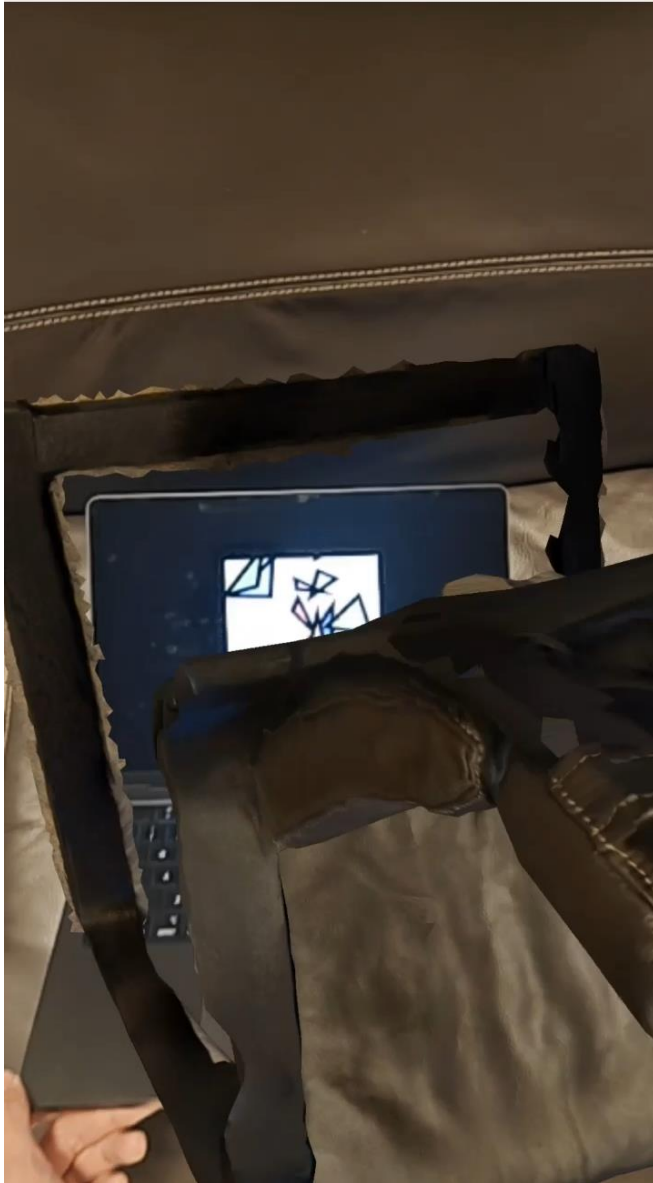- Also may we can find a **better marker**

Limitations

- Mistracking
- Limitation of **resolution** of camera
- Also may we can find a **better marker**

Limitations

- The furniture can be shown in real size

- size veries with the marker size

- But the problem is, that the display is often too small and the marker is too far away to be recognized for real scale visualization

I guess this will be somehow good <u>in the future</u>, not only the user can play around with ar, also
this system don't need to scan the whole room

I used **the system**, given by IKEA, when I want to buy the chair.
MAY caused by the ground is to homogenious,
so for the model still hard to coordinate**,**
even I ve used my whole heart during the scanning, it just doesn't work

Is Anyone interested in investment or buy our workflow, I guess all u left is just a good lawyer, who is good at user privacy..

References

- Gasques, Danilo, Alexander Scheurer, and Pedro Lopes (2021). danilogr/AR-Marker-Generator v1.0. Version v1.0. URL: https://doi.org/10.5281/zenodo.4466705.

- Marcos, Diego, Don McCurdy, and Kevin Ngo (2024). A-Frame – A web framework for building 3D/AR/VR experiences. URL: https://aframe.io/. [Accessed 07-02-2025].

- Wursthorn, Sven (2025). Lecture Slides - Augmented Reality, Winter semester 24/25. URL: https://ilias.studium.kit.edu/ilias.php?baseClass=ilrepositorygui&cmd=view&ref_id=2488247. [Accessed 11-02-2025].

- Yuen, HiuKim (2024a). MindAr Documentation. URL: https://hiukim.github.io/mind-ar-js-doc/quick-start/overview/. [Accessed 07-02-2025].

   – (2024b). MindAR Github. URL: https://github.com/hiukim/mind-ar-js. [Accessed 07-02-2025].

   – (2024c). MindAR Image Targets Compiler. [Accessed 08-02-2025]. URL: https://hiukim.github.io/mind-ar-js-doc/tools/compile.

Link to our Github Repo: