Hernan S., Akshay G., Rohan S., Sumukh B., Supreeth B.

Computer Vision

**Avo Insights: Web AR / Browser Capable Plane Detection & 3D Object Placement**

1. Research on Plane Detection
   1. Business Aspect
      1. (Q1): Being able to detect a physical surface and place a 3D model of an object is useful for people who would need instructions on how to do something. For example:
         1. It could tell you which buttons to press when using a new device.
         2. The 3D model could be used in conjunction with augmented reality. This could potentially help fix hardware if there's a mini person fixing the hardware on your table top. This is better than a video because you could see all angles of what is being fixed.
         3. Avo Insights specifically showed a 3D model of how to clean a wheelchair.
      2. (Q2): According to Avo Insights (Industry Partner), this idea’s pain point, or really any technology, is the ease of access. Most people don’t want to install another app on their phones in order to do something. Being able to access it from anywhere like a web browser would be better for something that would only be used once to obtain information.
   2. Literature Review

# PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image: <https://arxiv.org/abs/1812.04072>

* + - 1. (Q3): Relevancy: PlaneRCNN outperformed the other top methods by achieving an increase in 10% recall (MSW-G).
      2. (Q4): This method is able to achieve detection of small surfaces, it doesn’t require the number of planes prior to detection, and generalizes better across domains compared to previous methods (PlaneNet and PlaneRecover).
      3. These results were achieved by comparing these detection models against the same input images (RGBD) and then analyzing the predicted detections.
      4. Possible Problem: These methods used stereo cameras which are able to generate RGBD images; this is a problem when implementing to mobile devices because most don’t have this type of camera.
    1. Capabilities of ARCore and ARKit Platforms for AR/VR Applications: <https://doi.org/10.1007/978-3-030-19501-4_36>
       1. (Q5): This research is relevant because it compares the AR libraries from both Apple (ARKit) and Google (ARCore) which is used on mobile devices.
       2. (Q6): The results when detecting a 35 x 24 inch surface were not as real time as I expected. ARKit detected it in 13.6s and ARCore in 12.98s.
       3. (Q7): These results were gathered by using different mobile devices (IOS and Android) to detect different types of surfaces
       4. (Interesting Problem): One thing that was interesting was that they couldn’t detect walls. ARKit did somewhat detect the wall but from a distance of 8 inches and that took 39.25 s to detect.
       5. Conclusion Note: They’re both slow, maybe things have improved since this paper was released (Jan. 2020).
    2. GroundNet: Monocular Ground Plane Normal Estimation with Geometric Consistency: <https://doi.org/10.1145/3343031.3351068>
       1. (Q8): This research appears relevant to AR because it can detect a ground plane’s 3D orientation with a monocular camera, which most phones already have.
       2. (Q9): [What is unique about this solution?] GroundNet does this by first performing ground segmentation to identify the ground and then uses the RANSAC method to perform depth estimation and extract the orientation of the ground.
  1. Open Source
     1. ARKit (Apple’s AR Library)
        1. Available for Javascript. So it will work on a web browser.
     2. ARCore (Google’s AR Library)
     3. (Q10): [Open-Source] These libraries are open source code that seem good for AR applications
     4. (Q11): Both these libraries don’t require training in order to perform the plane detection, therefore there’s not really any data that’s necessary in order to solve the problem.
     5. Conclusion:
        1. Both of these AR libraries are very active.
        2. These libraries are very good at abstracting the problem. Both can perform plane detection with built-in functions
  2. Industry Solutions
     1. (Q12): [Solving similar problems] The ARKit (Apple) and ARCore(Google) appear to be the biggest contenders when it comes to implementing AR on mobile and web platforms.
     2. (Q13): Reverse Engineering doesn’t need to occur as these implementations are open source.
     3. (Q14): [Talks/Documentation/Proprietary] Apple provides an overview of ARKit’s capabilities in these videos:
        1. <https://developer.apple.com/videos/graphics-and-games/ar-vr/>
        2. This might be useful when trying to implement other things in an app
     4. (Q15): [Are there available talks, documentation, or other resources from their engineering teams?] –
        1. **AR**.**js** is a lightweight library for Augmented Reality on the Web, coming with features like Image Tracking, Location based AR and Marker tracking.
        2. Another framework typically used for browser based VR experience is **A-Frame.** It is a web framework for building virtual reality (VR) experiences and is based on top of HTML. A-Frame is not just a 3D scene graph or a markup language; the core is a powerful entity-component framework that provides a declarative, extensible, and composable structure to three.js.