Multimodal Memory Palace

Fabian Yanez Laguna
fyanez@mit.edu

1 Introduction

The memory palace technique is a powerful ancient memorization technique that leverages spatial memory to enhance learning and recall. It works by associating vivid mental images with locations of a "memory palace", such as a home or any other place that a person is very familiar with. People can then visualize themselves walking through this familiar place while seeing the images that they associated with each location, allowing them to memorize sequences in a natural way. This technique has many applications, including memorizing objects, events, and numbers, ultimately helping people memorize sequences or lists for academic, professional, and personal settings. For instance, it can be used to memorize information to study for exams and to prepare for public speaking. In fact, the technique is used by many professional memory athletes and has proven to improve retention rates in academic settings. To many people, this technique may seem obscure, but in practice it is very effective even for obscure use cases, such as memorizing arbitrary strings of hundreds or thousands of digits.

While there are interactive memory palace systems that exist, all of them are either completely virtual or rely on expensive AR/VR equipment. My solution is to develop a multimodal augmented reality iOS app that allows users to create and use memory palaces without requiring any expensive equipment other than a mobile device. Using Apple's computer vision models and built-in LiDAR sensor available on iOS 13.4+, the app allows users to walk through a location while pointing the camera and overlaying images at different locations to simulate a memory palace. By being able to physically see the memory palace through a phone, this application amplifies the effectiveness of the memory palace technique, which is typically performed entirely in one's mind.

Introducing additional senses, namely, hearing and sight, to the traditional memory palace usage significantly enhances one's ability to reinforce memory through this technique. When recalling memorized content, not being able to remember the content with one modality is not only complemented by the other modality, but they strengthen each other due to the brain's natural way of learning things with inherently intertwined modalities.

2 System Overview

2.1 Usage

To illustrate the effectiveness of using a multimodal memory palace app as described above, let us walk through an example. Suppose a user wants to memorize a sequence of historical events that occurred around the same time period as World War II. The first step is to "build" the memory palace. The user picks the home as the memory palace and starts at the front door. At each location that they want to associate a historical event with, they point the camera at the location and attach an image from their camera roll. The size and orientation of the image can then be adjusted while still pointing at the desired location. For example, let's say that the user wants to associate the dining room with the invasion of Poland 1939. The user could place an image of the Polish flag on top of the dining table. After the dining room, the user walks into the kitchen and wants to associate the stove with the Attack on Pearl Harbor that occurred in 1941. The user then places an image of a Japanese battleship on top of the stove. After walking through the rest of the desired path and attaching an image at each location, the user can click a button to finish the building phase of the memory palace. Let's say that the user wants to study for an exam and put the memory palace to use. The user would select the memory palace that they built and begin at the starting location of the memory palace, which is the front door of the house. Pointing the camera throughout the whole process, the user walks through the house using the same route. On the phone, they see everything that's in front of them but with the images they placed at each location. This allows the user to reinforce their memory palace by physically seeing the images. Finally, during the exam, the user can visualize themselves walking through the memory palace, seeing the historical events at each of the locations in order.

2.2 System Architecture

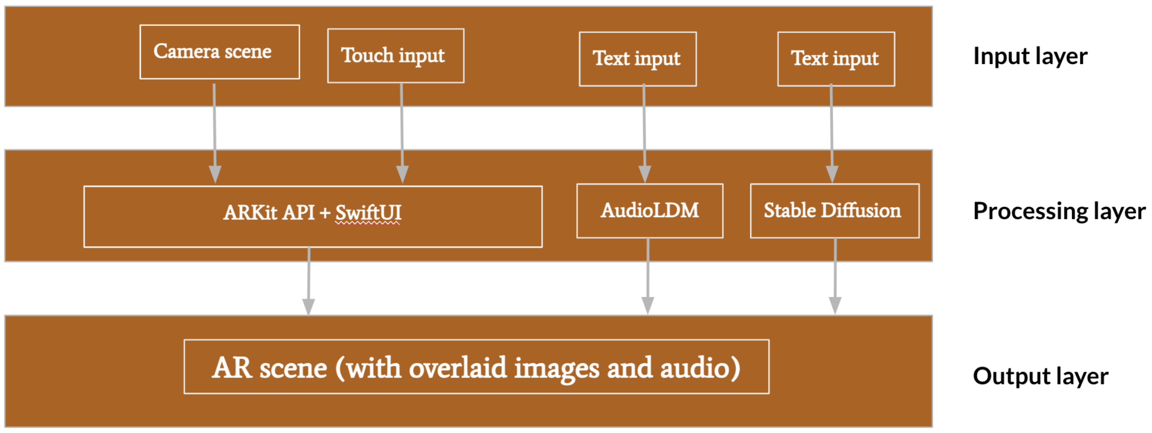

Figure 1: System Diagram

2.2.1 Augmented Reality

The system is broken down into three major layers: the input layer, the processing layer, and the output layer. ARKit, developed by Apple, is the central augmented reality API that deals with plane detection, image placement, scaling, and localization. The only requirements for the API to successfully detect planes and place objects on them are for the camera to capture a large enough space for the environment's planes and scale to be determined. This can typically be done by simply moving the camera around 15 degrees in each direction. The user, who is ultimately in control of choosing where to place content in their environment, naturally provides the necessary camera input by holding the camera and moving wherever they want to. The API relies on LiDAR and computer vision to accurately detect planes. When it comes to localizing placed images, iPhone's built-in IMU is able to effectively track a user's location relative to the starting location by leveraging accelerometers and gyroscopes. Note that it does not calculate absolute location, but once an image is placed, user's can walk long distances and come back to see the placed image in essentially the same exact location.

When users tap on the screen, an image is placed on the face of the plane that is detected. In the case that either a plane is not successfully detected or the image/audio are not fully generated yet, nothing happens. When an image is placed for the first time, an ARScene is created to keep track of all the images that are being placed in the user's environment. Since the generated audio is looped, the sound fades and gets louder as users get farther and closer to the images, making the auditory part of the experience more natural and realistic.

2.2.2 Image and Audio Generation

For the text-to-image model, a lightweight stable diffusion model was chosen to balance latency and image quality. Using a Python Flask server, the images are generated in about 3 minutes using the CPU. For the text-to-audio model, AudioLDM was chosen due to its accuracy superiority over other lightweight models. AudioLDM typically took around 1 minute to generate, so the image generation was the bottleneck. Note that the text-to-audio prompt is not for a text-to-speech model; rather, it is a short audio clip generated from a prompt describing that audio, such as "birds chirping in the air" or "knocking on a door".

2.2.3 User Interface

The iOS user interface is built with Apple's SwiftUI framework. To assist first-time users with the platform, ARCoachingOverlayView is used to overlay messages to guide the user, such as telling the user to tap a plane or to move the camera at the start to calibrate the augmented reality system. This is part of the ARKit framework and API. On the bottom of the screen, users can type natural-language prompts to generate both the audio and image, allowing them to use a different prompt for each. If the user wishes to reset the scene, they can simply click the exit button and go back to the home screen, which contains the name of the app and a button to enter the memory palace.

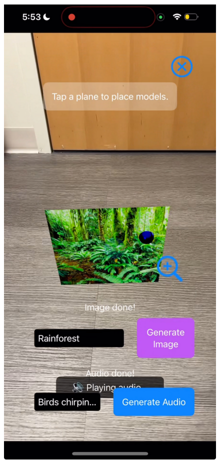

Figure 2: User interface

3 Evaluation

3.1 User studies

To uncover usability issues and validate design decisions, I designed a series of tasks for 5 different users to perform. I had each user try to place a specific audio and image in three different locations, walk through the created memory palace, and recite the content that the placed without explicitly telling them to memorize the content at the beginning. Simple instructions were given to the users and they were not aided throughout the process in order to uncover issues with the lack of clarity in the UI, bugs, and limitations of the system. In addition to spectating each of the user tests without helping, I debriefed each of the users afterwards as well.

After doing the user tests, the main takeaways were related to the system's limitations, rather than the UI. In particular, the system fails to detect surfaces that are very bumpy in nature or non-uniform in color. Additionally, the planes also have to be almost perfectly flat and the system works better on horizontal surfaces in general, which is a limitation of the augmented reality API, not the system design in and of itself. Lastly, users had trouble placing images in cluttered areas or areas with low lighting, often getting frutrated after many attempts. This made me realize that adding messages in the UI whenever an image is failed to be placed would be very helpful in guiding the user.

4 Limitations and Modifications

Initially, I planned on using USDZ format 3D models in the app. The ARKit API is specialized to be able to handle 3D models, which is why I did this for my v0. However, my ultimate goal was to allow users to build any memory palace they had in mind, so I did not want to be limited by the finite number of preset 3D models that can be found online. Therefore, I turned to AI models to generate the desired content. Since there are no existing models that generate 3D models from text prompts effectively, I decided to pivot to images. Nonetheless, the ARKit API supports images very well by essentially turning them into a 3D model that looks like a sheet of paper.

For the text-to-audio model, I experimented with several models before choosing AudioLDM to balance latency and quality. I applied the same process when choosing the text-to-image model. I also considered using an LLM to overlay descriptions of the images being generated. However, I chose not to do this to stick to the core principle of a memory palace technique, which only involves the objects or images and sensory information related to them, such as audio, touch, and visual animations.

In practice, if the models were cloud-hosted and used GPU processing rather than a CPU, the image and audio generation would be in the scale of seconds, rather than minutes. This app is a proof of concept that, with the right hardware and software resources, a multimodal augmented reality app can be built on a mobile phone, unlocking powerful memory capabilities for users in their everyday environments.

5 Key Resources

ARKit: `https://developer.apple.com/documentation/arkit`

UI starter repo: `https://github.com/ynagatomo/ARBasicApp`

Stable Diffusion model: `https://huggingface.co/blog/stable_diffusion`

AudioLDM model: `https://huggingface.co/docs/diffusers/en/api/pipelines/audioldm`

References

MIT Media Lab NeverMind Project: `https://www.media.mit.edu/projects/nevermind/overview/`

NIH paper on effectiveness of memory palaces: `https://pubmed.ncbi.nlm.nih.gov/25039085/#:~:text=The%20method%20of%20loci%20(MOL,arrange%20and%20recollect%20memorial%20content.`

Benefits of augmented reality in learning: `https://nsflow.com/blog/9-benefits-of-augmented-reality-in-education`

AR enhancing learning experience: `https://www.nature.com/articles/s41599-023-01650-w`