Continuity Report

GroZi: Grocery Shopping Assistant for the Blind

ENG 100L



Faculty Advisor: Serge Belongie T.A.: Brent Lee

Team Lead: Robert Tran

Team Members: Chung Chen Jose Melendez Castro Donnie Douangchak Alrik Firl David Ing Kayla Kimura Anna Shilova

Goal and Quarter Approach

The overarching goal of the GroZi project was to create a system to aid the visually impaired in shopping for groceries. The scope of the project thus far is limited to the shelved section of the grocery store. Because this is the final quarter of GroZi as a TIES project, we focused on developing a fully functional Android application that can recognize cereal box images and play a sound clip of the name of the cereal. Our team felt that developing an application on the Android platform to work with commodity hardware would offer the most simple, future-proof and robust means for reaching the project goal.

Client

Our client is the National Federation of the Blind (NFB), the largest and most influential membership organization of blind people in the United States. The NFB improves blind people's lives through advocacy, education, research, technology, and programs encouraging independence and self-confidence. The NFB representative that we corresponded with during the quarter is John Miller, a San Diego resident.

Methods and System Design

In previous quarters, the focus of the GroZi TIES project was to design a multi-touch surface and camera mounted on a shopping cart. However, this approach was plagued by multiple downsides; the solution was ultimately limited to grocery shopping, significant restraints were placed on system portability, and the multi-touch surface was error-prone. While previous GroZi TIES teams made significant progress with these methods, we decided to take a new approach this quarter -- an Android application based on Qualcomm's Augmented Reality Software Development Kit. We felt like this approach was the best because it is portable, not necessarily limited to grocery shopping because it eliminates the cart, and is not dependent on specialty hardware because it will work with on most Android devices.

Qualcomm's AR SDK provides computer vision technology to align graphics with printed surfaces and simple 3D objects, support for multiple development tools, and royalty-free development and distribution. A QCAR SDK-based AR application is composed of the following core components (Figure 1):

- 1. *Camera* Ensures that every preview frame is captured and passed efficiently to the tracker.
- 2. *Image Converter* The pixel format converter converts between the camera format to a format suitable for OpenGL ES rendering and for tracking.
- 3. *Tracker* Contains the computer vision algorithms that detect and track real world objects in camera video frames. Based on the camera image, different algorithms take care of detecting new targets or markers, and evaluating virtual buttons. The results are stored in a state object that is used by the video background renderer and can be accessed from application code.
- 4. *Video Background Renderer* Renders the camera image stored in the state object. The performance of the background video rendering is optimized for specific devices.
- 5. *Application Code-* Application developers must initialize all above the components and perform following key steps in the application code:
 - a. Query the state object for newly detected targets, markers or

updated states of these elements

- b. Update the application logic with the new input data
- c. Render the augmented graphics overlay
- 6. *Target Resources* Created using the on-line Target Management System. The downloaded assets contain an XML configuration file - config.xml - that allows the developer to configure certain trackable features and a binary file that contains the trackable database. These are compiled by the application developer into the app installer package and used at run-time by the QCAR SDK.



Figure

Data flow diagram of the QCAR SDK in application development.

1

In order to create our Android Application, our team was divided into two sub-teams: Image Recognition and Event Handling. The sub-teams were in constant communication through the quarter, working within their sub-groups and coming together on a weekly basis to discuss progress and problems.

Image Recognition Team

Alrik Firl, Kayla Kimura, Jose Mendelez Castro, Charles Taira

The goal of the Image Recognition team was to collect high quality images to build a library for the QCAR image recognition algorithms to identify and to provide additional information about the image for easier recognition. The team registered for an account on Qualcomm's Augmented Reality Developer Portal (Figure 2), and selected cereal images based on the requirements of the SDK. These original images were scanned and uploaded to the My Trackables section on the Qualcomm Developer Network Augmented Reality website (Figure 3-a). The trackables are rated based on two aspects of the original image-- the number of features in the image and the amount of local contrast. Features are typically localized around sharp, spiky, chiseled detail in the image, such as the ones present in textured objects (or more generally, areas with large gradient magnitudes). The location of the image's feature vectors are represented on the images by small yellow crosses in the image analyzer (Figure 3-b). Local contrast provides additional richness in detail so that the image can be tracked properly.

As the application runs, it will acquire video and calculate feature vectors for each of the video frames. From here, it will match the frames' features against the database images, and if a sufficiently close match is found, classify the object in the scene. Unfortunately, the actual feature vector generation and matching algorithm(s) are proprietary, and as such are black boxes to us end-users, and cannot be tweaked or inspected. As such, we ourselves cannot improve recognition rates (short of writing our own recognition system), should the SDK prove to be unreliable in certain circumstances. This places additional emphasis on choosing high-quality database images, without noise or other aberrations.



Figure 2 My Trackables on Qualcomm Developer Network



(a)

(b)

Figure 3 Original Scanned image (a) and image with Features shown (b)

Event Handling Team

Robert Tran, David Ing, Chung Chen, Anna Shilova, Donnie Douangchak

While the Image Recognition team worked on ways to recognize the various grocery store products, the Event Handling team was tasked with having our application announce to the user what exactly is the product they have in their view. This would allow our blind user to be able to easily know what exactly they have in front of them as they were casually shopping around the grocery stores.

With the Android Qualcomm Augmented Reality code that we received from Jeffrey Su and the help of Android text-to-speech, most of the functionality was already implemented for us. Jeffrey Su was our Qualcomm consultant that was gracious enough to help us get our feet on the ground with this project. In the early stages of our development our main goal was simply have the application recognize the product, and when prompted by the user, the application would announce what the product was. Obviously this would not be ideal for a blind user, but we simply wanted to implement the bare-bones functionality before moving onto automatic recognition.

The Qualcomm AR code that we received from Jeffrey was designed to recognize various movie posters and play a movie trailer from the Internet when prompted. The user received visual feedback whenever the application recognized the poster in the form of a box superimposed on the image (Figure 4). Given the high degree of correlation between the original Qualcomm AR

functionality and our proposed task, we decided to use this as the backbone for our implementation and have our GroZi application draw a box around the box.

Upon recognition of the cereal box, we played a sound clip of the cereal name with SoundPool, an Android media class that manages and plays audio resources for applications. This was a better choice than Android MediaPlayer which is better suited for long sound files. SoundPool is best for short rendered sounds. It uses 16 bit pulse code modulation (PCM), which is roughly CD quality, and does not need to dynamically decompress the sound files which decreases latency.



Figure 4 The application recognized the Lucky Charms cereal box trackable, and draws a white box superimposed around cereal box image

Future Plans

Now that we have created a solid foundation for the GroZi application to use and move forward with, a pressing area of focus should be in testing and receiving feedback from non-sighted individuals. To address this, we plan to upload our product onto the Android Market for regular consumer use. This will not only allow us to get our product out into the real world, but it will also give us a much wider testing pool to draw feedback from. In doing so, we hope to obtain much needed feedback on what we have accomplished thus far, and what we have yet to do. As it stands, our application is still in its early-beta phase; it would be of little practical value in a real-world grocery store setting, and next to useless for a completely blind user. Some of this is correctable, other parts less so. The first issue to address is the need for user interaction in the detection process; rather than requiring the (blind) user to manually select the item on the tablet screen in order for the application to vocalize, making the application speak automatically upon detection would greatly facilitate usability. One area that can not be actively improved upon is the accuracy of the recognition algorithm. Since the interest points are generated under the hood by the SDK, we can only indirectly influence or improve upon the error rates (potentially via pre-processing techniques).

Although GroZi is officially done with the TIES program, our faculty advisor Dr. Serge Belongie still wants to continue on with the project and have a fully functional and test application to be given out to the community. Due to this GroZi will continue on as an upper division independent study for undergraduates who are also interested in working in something like this. Students will be able to continue working on the GorZi project closely with Dr. Belongie while obtaining upper division elective credit towards their degree. We assume that as it moves forward they will continue to use the Android Qualcomm Augmented Reality Research documentation.

Acknowledgements

Dr. Serge Belongie: Faculty advisor Dr. Mandy Bratton: Program Director Jeffrey Su: Qualcomm Consultant Barb Donovan: Program Specialist Brent Lee: Teacher Assistant Zach Salin: Global TIES Fellow