Continuity Report

Grocery Shopping Assistance

Grozi

Faculty Advisor: Serge Belongie
T.A.: Brent Lee
Visiting Scholar: Kris Kitani

Ted Chu
Jeffrey Wurzbach
Sebastian Ortega Zafra
Shauna Thomas
Jeffrey Su
Cankut Guven
Yu-Tseng Chou
Nil Kumar

UCSD ENG 100L
Global Ties Fall 2010

Introduction

Goal and Quarter Approach

GroZi's goal is to create a system to aid the visually impaired in shopping for groceries. The scope of GroZi is currently limited to the shelved section of the store. This project has been developing for some quarters now, but in this quarter it was decided that we would start with a new approach that would allow the project to improve the communication and assistance for the blind user.

This quarter we focused on user interface development using sonification. Sonification is the process of transforming information or data into sounds. For this quarter, there was a division from within the team in order to present better and faster results. The team was divided in 3 subteams that were mainly identified by the kind of information they processed and received. These were the 3 teams:

- Image Team
- Sonification Team
- Interface Team

All of the subteams needed to have constant communication and feedback in order to process efficiently the information that is being extracted from the user and its environment. Each one of the teams used different platforms and programs to process the information.

As an additional work, a member of our team worked during the quarter with gestural interfaces to provide the general team a better understanding of better ways for a person to interact with a device, like the one we attempt to create. This work is resumed in the last part of this continuity report to provide an insight of what will be studied in depth during the next quarter.

Image Team

Ted Chu, Jeffrey Wurzbach

For the fall quarter of 2010, the image team provided distance transform (or distance map) images created from the Microsoft Research (MSRC) database. The MSRC dataset has pre-segmented images that have been re-colored based on the segmentation. The MSRC dataset was segmented and re-colored by humans to help in creating object recognition systems. The sonification team uses the distance transform images to create sounds for the blind user. Pre-calculating the distance transforms allows the sonification team to simply look up the data from memory, rather than recalculating the transform every time the UI passes new values to the sonification team.

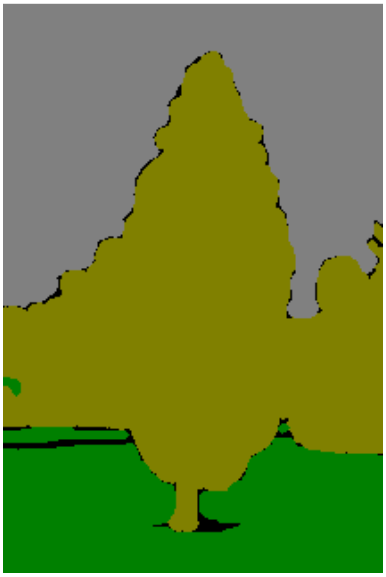


Figure 1: Segmented image with a tree, grass, and a sky

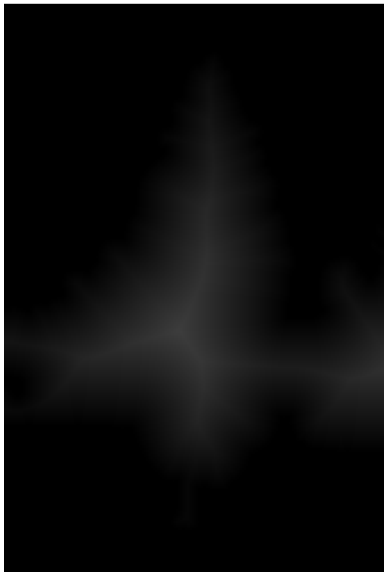


Figure 2: Distance transform of the tree region in the gray color space

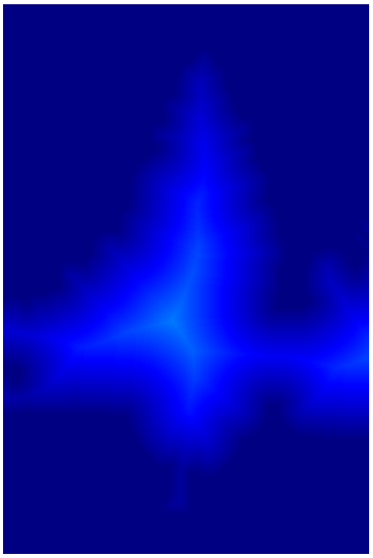


Figure 3: Figure 2 re-rendered in the jet color space

To prepare the images for use by the other sub-teams, we produced distance transform images based on certain features in the source image. The distance transform creates a grayscale image from a binary image that represents the distance a pixel is from the TRUE-FALSE boundary of the TRUE region as the intensity (brightness) of the pixel. There are several methods for determining the intensity value, but we used the Euclidian method.

$$I_{row,column} = \sqrt{\left(B_{column} - P_{column}\right) + \left(B_{row} - P_{row}\right)}$$

Equation 1: Euclidian Distance Transform

In Equation 1, the variable **I** represents the intensity of the pixel at the location (row, column). **B** represents the closest boundary pixel and **P** represents the pixel we are at which the distance transform is being evaluated. To forum the required binary input image for the distance transform, a logical mask was used to remove unwanted features from the image. The logical masks used to form our binary images looked at each color channel and set the pixel to TRUE if it met the values we furnished for region of interest. Table 1 shows the RGB values for each region of interest.

Region	Red Value	Green Value	Blue Value
Sky	128	128	128
Tree	128	128	0
Grass	0	128	0
Building	128	0	0
Plane	192	0	0
Cow	0	0	128

Table 1: RGB Values for the Regions in MSRC Images used to create binary images in preparation for creation of distance transforms

The logical mask is formed in MATLAB by using nested logical operations. Listing 1 shows an example of MATLAB code used to make a binary image.

```
ASky= not(and(Ain(:,:,1)==128,and(Ain(:,:,2)==128,Ain(:,:,3)==128)));
```

Listing 1: MATLAB code for a logical mask marking all pixels of sky TRUE

The structure of Listing 1 examines the three-color channels of the input image, Ain. Two AND functions are called to create a three input AND, one input per color. Output of the AND operations creates a binary image; however, it is inverted with respect to the desired result. The NOT call inverts the values of the image, changing a FALSE to a TRUE and a TRUE to a FALSE. The output is fed into the variable ASky.

There are two ways the distance transform can be shown. One method is to sum the distance transforms of the images, created a composite image transform. The other method is to leave the distance transform separate. The main advantage of leaving the transforms separate is that it

allows the sonification team to let the user to search for a given object within the image and find its center. To create the first method, the three images can be quickly summed on the fly. The first mode is useful for determining where things are in the image.

A MATLAB function is under development to obtain a distance transform from a color input image. The function converts the image to grayscale, translates it to a binary image and then performs the distance transform. It resizes the image to specified size and then returns the resized image. There are some bugs that need to be resolved with the function. Another MATLAB script was created to track IR LEDs; however, it was not generalized and is only works on a still image. It needs to be generalized and modified to accept live video. Only 6 images where transformed this quarter, more images will be required as testing progresses. The balance of the MSRC dataset should be converted to ensure a large supply of testing images.

Sonification Subteam

Sebastian Ortega, Shauna Thomas

Goals

- Receive information from Image and Interface teams (coordinates and distance transform) and transform it into specific sounds
- Define general sound characteristics (pitch, shape, volume, tempo) for guidance
- Create a voice database associated to the MSRC objects

Sonification

This quarter we were able to research and get acquainted with the sonification concept. In the beginning of the quarter we were able to also know some of the most interesting approaches that had been already used in other products or projects. By doing this, we gained a bigger perspective on what our task would be and eventually we were able to apply this in developing a system for basic sonification using Chuck programming language.

Sonification is the process of transforming data or information into sound. There are many approaches to this particular theme but in our specific project we tried to give this sonification a guidance approach. In this way a blind person would receive information from the image in front of him by hearing sounds that depend on the object and (i.e. the position of his fingers within a touchscreen virtually representing the image).

The information we have to receive and then transform into sound was the next:

- Category (Object)
    - MSRC Database
- Position of user input (x,y)
- Distance transform of (x,y)

The platform used for this quarter to produce the sonification was Chuck Programming Language. More specifically, the MiniAudicle platform which is able to run chuck programs and codes without even installing any software. There are many examples within the same file that includes this platform under the 'examples' folder. Our team went through most of these examples to try to find out how to do basic actions with Chuck, such as creating a sine wave and modifying its characteristics (gain, freq.,), receiving and sending information, taking information from the keyboard, creating a beat, reproducing wav files, using stereo panning, mixing sounds, etc..

As a note for future work, some of the sounds that could have potential for eventual sonification (either using the distance transform or a single tone) are:

- *larry.ck, curly.ck, oft_01.ck, oft_ 02.ck, oft_03.ck, tick.ck,* and all of the files under the 'basic' folder.

Once that we were able to understand better the programming language as well as its basic functions we started developing some basic version helped by Kris' supervision and help. In the meetings with the rest of the GroZi team we decided to create two versions (0.1) of the sonification system. The first one would use the distance transform and the object identified to provide the user with a guidance sound that would indicate the proximity to an object. This idea came up from Tsubasa's iphone app. The second would merely identify the object by reproducing a wav file naming the sound.

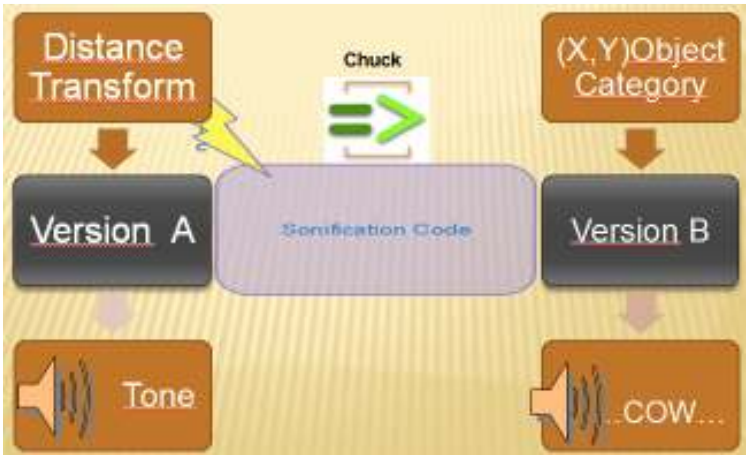
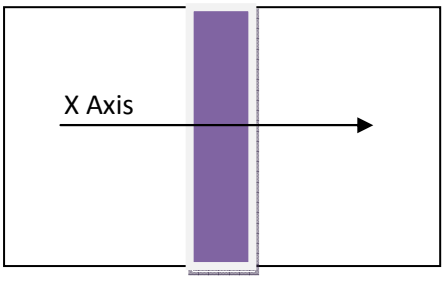


Figure 4. Diagrams of Sonification demos versions A and B

These versions were comprised of two chuck programs each. The first one sent a stream of information through a port using OSC protocol. Both of these programs had sending programs that would simulate the information sent by the image and the interface team. This information was taken from the mouse of the laptop, providing us with a position that would be very similar to the actual input that we would eventually receive from the multitouch screen. In this way we could work and experiment with different sounds and effects without having to wait for the actual information in a real live demo. The second one would receive this information and then perform the specific sonification task.

VERSION A. This version worked by using a 1D distance transform (X axis) defined by us. Once the second program received the position of the mouse in the screen it would reproduce a sine wave with low volume (low gain parameter) that would increase its frequency (higher sound) as it approached a 'virtual object' in the middle of the screen. Once it is in the virtual object the tone increases volume and stays at a constant frequency while the inputs of the mouse position remain in the 'virtual object'. Once the input goes out of the border of the object the sound returns to a low volume tone which frequency goes low as the position goes away from the object in the center. The next figure represents the screen and our 'virtual object'.



VERSION B. This version involves a much more simple process. This just reproduces the wav file containing the voice description of the object. It just receives the object identified and then proceeds to relate it to a specific voice sound. We had to change in this version the sending program so that it would send information every 20 changes in the mouse position because if not the receiving program became too slow by trying to reproduce a sound for every position change in the mouse.

The output of both versions of the sonification system are an audio stream either with a:

- A) Tone with Frequency Indicating Distance from Edge, or a
- B) Spoken Description or Name of Image

This quarter was the first approach to sonification and very important concepts can now be worked on for the next quarter. Some programs were developed using basic Chuck functions. The potential and variety that Chuck provides can be shown in some of the examples included in the MiniAudicle folder of examples. Although the mouse position within the screen works as a good test input for some programs it does not provide a good testing input for others because of the speed with which ther is a change in the position. Keyboard input might be a more appropriate input for tests. Stereo panning should. The implementation of broader and more complex sounds should be one of the main tasks for the next quarter. Multitouch screen provides a lot of options regarding sonification. Whether it is one finger, two fingers or more, the sonification system needs to have more options that perhaps could include stereo panning, sound mixing and sound effects such as delay and reverb. These sounds should also have a feedback by blind users to prove their reliability and quality for guidance. These issues should be looked into in the future, but in general this quarter was a good approach to sonification as a tool to guide a blind person within an image.

Interface Team

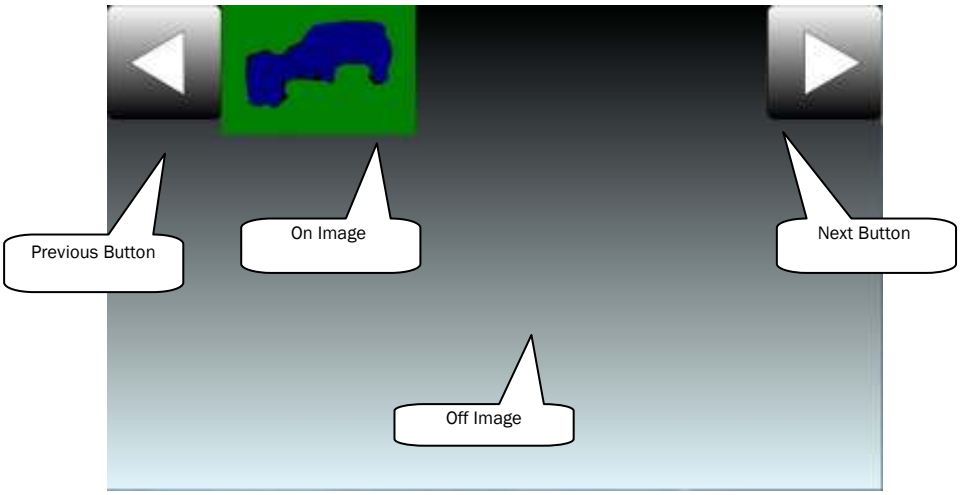
Jeffrey Su, Cankut Guven, Yu-Tseng Chou

Introduction

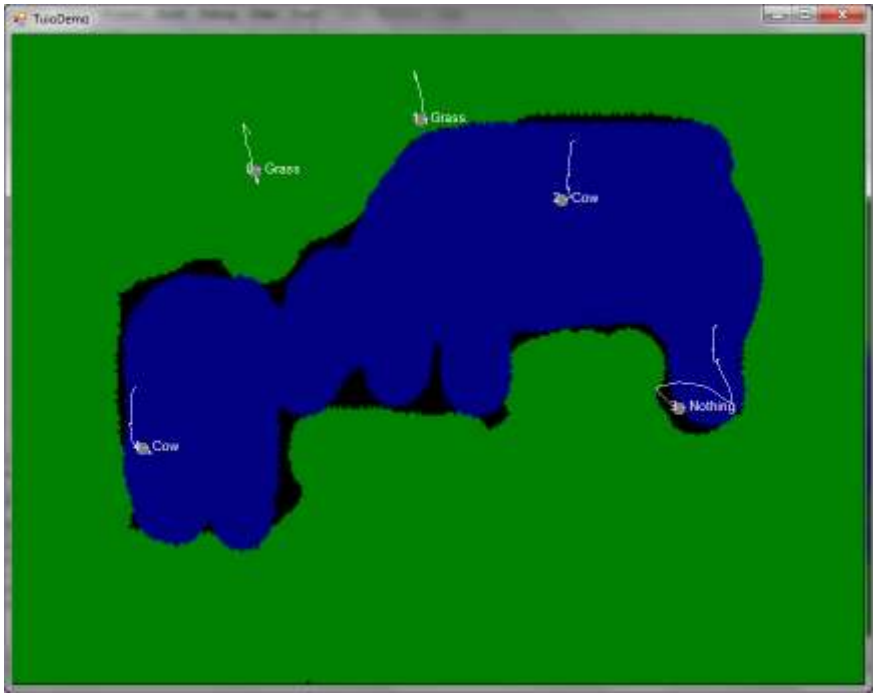
The GroZi project this year focuses on an entirely different goal from the previous years; this year, the project emphasizes the interface aspect of the Grocery Shopping Assistants for the Blinds. In other words, the problem was how to convert visual information to sound information for the visually impaired users. In an effort to solve this problem, the whole team was divided into three sub teams—the image processing team, the sonification team, and the interface team. In particular, the interface team works on the visually impaired user friendly platform that serves as an intermediate between the users and the background system.

Interface Team

The interface team is responsible of taking in the user's input of the finger touches and outputs the sound generated by the sonification team with respect to different types of input. For example in the object identification mode, the interface device detects the coordinates or pixels where the user's fingers touch on and finds the corresponding information regarding its color. Below is an interface platform design with a text-to-speak system playing different sound effects when the user mouseovers. The "Previous" and "Next" buttons were placed at the top two corners for the convenience of the visually impaired user. It was based on the observation that the motion of searching done by hand touching typically begins from the corners or edges of an object.



A mutli-touch system that allows the interface device to detect user's finger touches was also developed. The device is built using a box, a webcam, and a white screen and, through the shadow created by the finger touch, it detects and locates the coordinate of the user's fingers. Below is an example showing the basic design concept.



The multi-touch system is able to detect multiple finger touches and output the objects' names with displayed texts at once. In the future, a goal will be set to enhance this program along with the interface platform design and further modify it with various types of mode.

Gestural Interfaces

Nil Kumar

Within the *User Interface* team, a sub-task for the quarter was to look into gestural interfaces. The literature search on gestural interfaces would ideally help in creating the modes of operation for the multi-touch interface that was created this quarter.

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via programmed computer algorithms using techniques from computer vision and image processing. Gestures can originate from any bodily motion or state but commonly originate from the face or hand (see figure A). This becomes useful in particular for GroZi because our clients are visually impaired and heavily rely upon senses other than sight. Providing them with an option to use gestures to interact with devices such as our multi-touch screen (devices they are already accustomed to) allows for further assistance in areas like grocery shopping that they cannot easily perform.

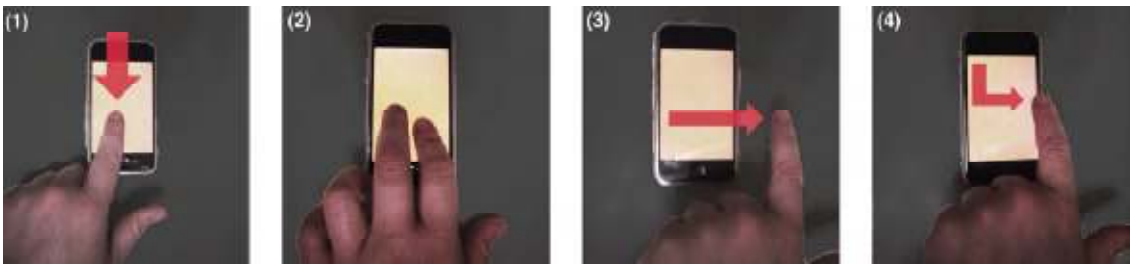


Figure 5: Gestural Interface on iPhone using touch gestures for operation

The primary focus in terms of gestural interfaces pertinent to the GroZi project was to look into specific interfaces that provided auditory feedback. Auditory feedback became a crucial factor when researching various gestural interfaces due to our client needs. We discovered that auditory cues were most effective for visually impaired users trying to perform such activities as grocery shopping. For example, the interface shown in *Figure A* above was created by a team at the University of Washington for use on mobile devices. The application provides auditory feedback upon the user running their fingers across the touch screen.

The recognized gestures for this application include:

(1) A *one-finger scan* is used to browse lists;
(2) A *second-finger tap* is used to select items;
(3) A *flick* gesture is used to flip between pages of items or a currently playing song;
(4) An *L-select* gesture is used to browse the hierarchy of artists and songs in the music player.

Product Research

Another popular mobile application that was discovered through the research was *iPhone VoiceOver*. iPhone VoiceOver is the world's first gesture-based screen reader that allows users to physically interact with the items on the screen (similar to the work done by the U. Washington

team). Prominent features in this application included the ability for the app to speak description of item under the finger, clicking noises to indicate choices made, and additional gestures that included up to three fingers.

Overall, the iPhone VoiceOver feature has a solid set of gestural interfaces for a touch-screen surface. In terms of the GroZi interface, we can definitely model some of our features based on these one, two, and three finger gestures – they seem to have been well received by the blind community.

Figure (5) to the right demonstrates *LookTel Real-Time Object Recognition* software. Pointing cell- phone camera at product reveals what user is looking at through auditory cues.

The other product that was fairly closely related to the GroZi goals was iVisitSolutions' *LookTel Real-Time Object Recognition* software. The LookTel program is a mobile phone app that recognizes objects in real time using the phones camera and provides auditory feedback as to what is being viewed. This is very similar to what we are trying to do with our multi-touch screen as an assistive device in a grocery store.

Although not a touch-screen interface, this mobile app possesses many of the qualities that are needed in the GroZi implementation. Image recognition is the key in this product, and blind users are provided auditory feedback simply by pointing the phone at the product. No gestures are required to detect images; however, similar finger gestures as the iPhone VoiceOver app are needed to navigate various screens of the software.

GroZi Software Ideas

In terms of software implementations for the multi-touch system, the *User Interface* team came up with two different ideas. Version A requires the user to actually identify the object on the screen by using his or her fingers on the touch-screen as guides. With this objective in mind, it made sense to go with gestures similar to those of the iPhone Voiceover application:

- One finger to scan screen searching for object (sounds to indicate distance from objects)
  - Once object is located, double tap one finger to tell user what they have located
- Two fingers on object can provide detailed information (i.e. coordinates)
- Three fingers flick action to switch modes / home screens (optional)

Version B simply deals with boundary detection. In other words, how well can the user trace the boundaries of an object (no detection / identification of the object required) using the touch-screen. The following gestures were created to assist in this process:

- One finger to scan screen searching for object (various pitches/tones to guide user based on distance from boundary)
- Two fingers tap to provide general direction of object (north, south, east, or west) in case user is way off