Augmented Reality for High-throughput Phenotyping

Shanshan Wu and Mitchell L. Neilsen
Department of Computer Science, Kansas State University, Manhattan, KS, USA

Abstract - *Augmented Reality (AR) enables elements of a computer-generated digital world to be integrated with a user's perception of the physical world. Smart glasses, like smart phones, have independent operating systems and they can support a variety of different applications and modes of communication to support augmented reality. This paper details the development of a novel new application that extends a widely-used mobile app for phenotyping and allows agronomists to interact with the app while keeping their hands free to perform field work. The smart glasses accept voice commands from the user and communicate with the mobile phone app via Bluetooth. In addition, changes detected by the mobile phone are displayed to the user on the smart glasses. This enables agronomists to efficiently collect phenotypic data.*

Keywords: Augmented reality, high-throughput phenotyping, real-time data processing, smart glasses, wearable computing.

1 Introduction

An ongoing project at Kansas State University, called PhenoApps, seeks to leverage novel advances in image processing and machine vision to deliver transformative mobile applications through several established breeder networks. Building on the success of Field Book (www.wheatgenetics.org/field-book) [1], several novel user-friendly mobile apps for field-based high-throughput phenotyping (HTP) have been built and deployed [2,3,4].

Augmented Reality (AR) blends information from a digital world together with the physical world [5]. Frequently, agronomists find themselves with too few hands when working in the field while collecting data on characteristics of growing plants, called phenotypes. Field Book was developed to replace pencil and paper and facilitate field data collection. The goal of this project is to extend the functionality of Field Book with AR by developing a new app called Field Book AR. In visual augmented reality, users can use smart glasses to display a combination of real world data with data from a mobile app, such as Field Book. Augmented reality technology includes new technologies and methods such as multimedia, 3D modeling, real-time video display and control, multi-sensor fusion, real-time tracking and scene fusion. Augmented reality provides information that is different from what humans can perceive in general.

Smart glasses, like smart phones, have separate operating systems, and can execute many different kinds of software and games. Smart glasses can be used to add a schedule, map navigation, interact with friends, take photos and videos, and make video calls with friends through voice control. They can support wireless access through communication networks.

Figure 1. Vuzix M300 smart glasses

Bluetooth is a radio technology that supports short-range communication between devices. It can be used to exchange information between devices including mobile phones, wireless headsets, laptops, etc. Bluetooth technology can effectively simplify the communication between mobile devices [6].

In the information age, agricultural technology can be very intriguing and smart. Because agricultural technology is typically not viewed to be as sophisticated as aerospace, developers might conclude that advanced data computing capability is not important to the agricultural sector. However, data and software have become integral in advanced agricultural domains. As smart devices mature, future agricultural technology products will become more lightweight and modular, allowing users to flexibly customize functions according to their needs. A more intelligent knowledge system will provide growers with more convenient and concise technical guidance. In our research group, we have already developed several mobile applications within this domain [2,3,4].

With advances in the computing power of intelligent electronic products, the use of AR is becoming more and more extensive. Augmented reality is already used extensively in visual arts and the medical domain. Augmented reality technology is also gradually being introduced into modern agricultural technology. It is currently used in satellite exploration, crop growth tracking, and GPS guidance systems [7, 8].

Field Book is an Android app used for taking notes on field research plots. Field Book has been developed and released in the Google Play Store and has over 5000 downloads. Both Field Book and this project are open source software projects. The code can be found on Github. They greatly reduce the cost of high-throughput phenotyping to users and provide reliable convenience. In this paper, we combine smart glasses applications with smart phone apps to provide an open source software application to support hands-free support in Field Book. In agricultural fields, users can use Field Book to record and monitor phenotypic data of plants. When data changes in Field Book, those updates are displayed on the smart glasses. This allows users to see the updates immediately. Users can make new corresponding observations and judgments based on the new data. Likewise, when a user wants to control Field Book, but they are busy with both hands, they can use voice commands to the smart glasses. The smart glasses can accept the voice commands. After analysis of the command, the smart glasses send the command to Field Book to take the related action on Field Book. This feature frees the user's hands, and provides more convenience and flexibility.

Section 2 describes extensions to the mobile application Field Book to enable Augmented Reality. The resulting mobile app is called Field Book AR. Then, Section 3 describes the complementary app developed on the smart glasses to communicate with Field Book AR. Section 4 details data flow and analysis of the application. Finally, conclusions and directions for future research are found in Section 5.

2 Field Book AR

The rapid collection and analysis of large amounts of phenotypic data is known as high-throughput phenotyping (HTP). Image-based methods for computing phenotypic measurements provide a promising solution for high-throughput systems in plants. There have been many successful applications of image analysis to plant phenomics including seed measurement, time series growth measurements of roots, plant growth and biomass in relation to abiotic stresses, plant morphology, and disease severity. However, these applications of image-based HTP have largely concentrated on laboratory or controlled environment facilities. While providing highly accurate phenotypes, transferability to harsh field conditions remains limited.

Field-based phenotyping remains the major bottleneck in most breeding and genetics programs, including cassava and wheat. As recognized by the breeders, this bottleneck can be overcome by adopting and/or developing innovative, high-throughput, and accurate hand-held HTP tools that will greatly increase the ability of breeders to generate useful data more cheaply and rapidly. As such, there is considerable willingness and motivation in the breeding community to test and adopt new technologies that can make the breeding program more efficient.

We have developed and deployed several user-friendly mobile apps that have been adopted by hundreds of breeding programs around the world. They are available in the Google Play Store and online at www.wheatgenetics.org/apps. Our apps have been designed to be stand-alone, simple, intuitive, and effective. They are delivering a significant increase in breeding productivity and are the foundation for the current project. These tools focus on simplifying and digitizing data collection by focusing on procedures that are common to all breeding programs. Field Book is an open-source Android app used to take phenotypic notes. The data captures one or multiple traits on each plot. Field Book is easy and fast because of the friendly user interface design. Compared with paper field book, the app can display additional imported data which users can observe more information with one glance. For each field record, users can choose any additional records of interest to display, like entry names, seed sources, etc. It is an ideal replacement for taking notes on paper. Field Book is open source, and developers can customize new features to meet specific data collection requirements [1]. To provide more convenience, Field Book is extended with augmented reality technology. When users wear smart glasses, they can monitor data changing in real-time, and they can control Field Book through voice commands. We start by describing extensions required to Field Book.

2.1 Data monitor

To implement data synchronization, a module, called the Data Monitor, is used to monitor for changes in data. This module implements several functions to get and set data. In the main class, when data changes due to user input or other updates, it calls the functions of the data monitor module. After setting the data that has changed, the data monitor module broadcasts a message to distribute data to the internal module. The message indicates that a data changing event has occurred. It may also include the updated data in the message package.

2.2 Data transfer service

The Data Transfer Service module is a thread that runs in the background without a user interface. The service handles some data processing in the background to avoid blocking the main thread. For example, it waits for and processes incoming and outgoing network transactions.

Data transfer service is started by the main class of Field Book. Its responsibilities are to filter messages, connect the Bluetooth Server to complete data transfer, receive messages to perform related actions. When Data Transfer Service is created, the default action is to register the filtering messages and start a Bluetooth Server. With filtering, it only receives data changing messages, Bluetooth status-change messages, and device-change messages. The Intent Filter is shown below in Figure 2.

```
mIntentFilter = new IntentFilter();
mIntentFilter.addAction("com.fieldbook.tracker.DATA_CHANGE");
mIntentFilter.addAction("com.fieldbook.tracker.BluetoothServer.STATUSCHANGE");
mIntentFilter.addAction("com.fieldbook.tracker.BluetoothServer.DEVICECHANGE");
registerReceiver(mActivityReceiver, mIntentFilter);
```

Figure 2. Intent filter

The messages are processed using the Activity Receiver shown below in Figure 3.

```
private class ActivityReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        Log.i("Receiver", "Broadcast received: " + action);

        String data;
        if (action.equals("com.fieldbook.tracker.DATA_CHANGE")) {
            data = intent.getExtras().getString("data");
            Log.i(TAG, data);
            mBluetoothServer.write(data.getBytes());

        } else if (action.equals("com.fieldbook.tracker.BluetoothServer.STATUSCHANGE")) {
            data = intent.getExtras().getString("message");
            mBluetoothServer.cancelConnection();
            startBluetoothConnection();

        } else if (action.equals("com.fieldbook.tracker.BluetoothServer.DEVICECHANGE")) {
            mBluetoothServer.cancelConnection();
            startBluetoothConnection();

        }


    }
}
```

Figure 3. Broadcast receiver

When it receives a data change message from the data monitor module, it acquires the changed data and calls the Bluetooth server to transfer data. The Bluetooth connection is broken when it receives a status change message. This message comes from the Bluetooth server module. At this time, the data transfer service module tries to restart the Bluetooth server. When users switch to a new device, the device change message is received. This is essentially a request to create a new connection. This message is generated when a user makes a change on the Settings Page in Field Book. After receiving the device change message, the data transfer service closes any current Bluetooth connection, and create a new connection for the new device. To receive messages, it creates a new instance of the Data Transfer Service which is an extension of the BroadcastReceiver class.

2.3 Bluetooth server

Bluetooth server is used to establish a new Bluetooth connection, accept and send data over the connection, and close a Bluetooth connection. After setting the Bluetooth adapter and device information, it starts a new thread called AcceptThread to wait for a client device to connect and accept the connection. The purpose of the new thread is to avoid blocking the main thread. During the waiting period, the main thread still works fine. After a client device sends the connect request, the connection is created successfully, and the AcceptThread is stopped. At the same time, the Bluetooth server creates another new thread, called ConnectedThread, to receive and send data to the client device. The purpose of ConnectedThread is used to send and receive data and avoid blocking the main thread. When the input data stream an interpreted voice command, the thread broadcasts the command with the action information. The main class receives the broadcast. When the ConnectedThread input data stream generates an exception, it broadcasts a status change message to close the connection. Data transfer service receives the status change message and tries to recreate a Bluetooth connection.

3 Smart Glasses App

We selected the Vuzix M300 smart glasses, shown below in Figure 4, for this project. They run the Android operating system, so the development process is similar to development of any Android app.

Figure 4. Vuzix M300 smart glasses

The app in Vuzix M300 is called Data Receiver because the original function on the smart glasses was to only receive the data stream and display them on the user interface. With the perfection of this app, it also sends data to the server device and accepts voice commands now. The user interface shows Bluetooth connection status (connected or disconnected), the current voice command mode (TRAIT SETTING), and the updated data which from Field Book below the command line as shown below in Figure 5. In addition to plant traits (phenotypes), the user might want to see information regarding the current plot; i.e., plot id, location, etc. In which case, the user can just issue the voice command “Plot” to switch from Trait to Plot.

(Disconnected Mode)

(Connected Mode)

Figure 5. Smart glasses user interface

3.1 Bluetooth client

Bluetooth client is used to create a connection with the Bluetooth server running on a handheld mobile device. After setting the Bluetooth adapter and device information, it creates a thread called ConnectThread to connect to the server device. Then, if the server is ready to be connected, a new connection is created and communication can commence. It sends a message to the main class to claim the connection created. Users can displays details regarding the connection through the user interface. If the connection fails, the user is informed on the interface, and the user can update the server device information or check to ensure that Field Book AR is opened and try to re-connect. When a connection is created, ConnectThread is stopped at once. Bluetooth Client starts a new thread call ConnectedThread to send and receive data over the new connection to the server device. Again, the purpose for having a separate thread is to avoid blocking the main thread and ensure that the user can interact with the main thread even when a connection fails. In contrast with the Bluetooth server running on the mobile device, the smart glasses client uses a Message Handler to pass messages. The reason is that Bluetooth clients only need to pass messages to the main class, but the Bluetooth server needs to pass messages to multiple modules. Sending messages by broadcast is more suitable for Bluetooth server.

3.2 Voice control

Speech recognition technology, also known as Automatic Speech Recognition (ASR), aims to convert vocabulary content in human speech into computer readable input such as binary codes or character sequences. In recent years, speech recognition technology has rapidly advanced to provide more convenient and reliable control of mobile devices. Several well-known companies have launched their own speech recognition products, like Amazon's Alexa, Apple's Siri, etc. The Android platform also provides related APIs to support speech recognition development on Android devices. In this project, we use the Vuzix voice recognition SDK to achieve voice control using the Vuzix M300 smart glasses [9]. We could have also used the Android phone to accept voice commands, but it was more convenient and reliable to issue commands directly to the smart glasses.

Figure 6. Field Book app interface

Voice control is an essential function in the project. With this feature, users can easily control devices. In the voice control module, we mainly support four voice commands for now. They are: ”Trait”, ”Plot”, ”Next” and ”Back”. Trait and Plot are the mode names, and they correspond to the modes "trait" and "plot" in Field Book. After setting mode, users can say ”Next” or ”Back” to scroll through the trait or plot dialogs in Field Book. For example, a user would first say the command: ”Trait” or ”Plot”, which would bring up the trait or plot screen, and persist as the current mode in saved mode variable. Then, the user could say ”Next” or ”Back” to move forward or backward in the list of traits or entries. For traits, this would be equivalent to pressing the green arrows of Field Book, and for plots, this would be equivalent to pressing the forward or back black arrows in Field Book as shown in Figure 6.

When the voice control module is created, it calls the functions of the Vuzix speech client to enable a voice recognizer and registers new voice commands. Anytime users say one of the voice commands that have already been registered, they can be recognized. After interpreting the voice command, the command is sent to the phone via the Bluetooth client module.

4 Data Flow and Analysis

The primary function of this project is to monitor data updates in Field Book and show them to users on the smart glasses display. There are several ways to trigger data change events, including user input to Field Book on the mobile device or voice commands to the smart glasses. We define two different types of data flow, active or passive, based on the source of the data. Active data flow is triggered by user input to Field Book. The data flow started with the Field Book user interface and terminated with the Data Receiver user interface on the smart glasses. Passive data flow is triggered by voice commands. It starts with the Data Receiver in the smart glasses where the commands are passed to Field Book. Then, the changes in data displayed are sent back to the Data Receiver for display on the smart glasses. Active data flow is simpler and more intuitive than passive data flow.

4.1 Active data flow

Active data flow means that the data update originated with Field Book on the smart phone. It could be from user input or a dialog switch in the user interface. When data changes, the Bluetooth server sends data out. The Bluetooth client receives it and the update is displayed on the user interface of the smart glasses. The steps involved with active data flow are shown above in Figure 7.

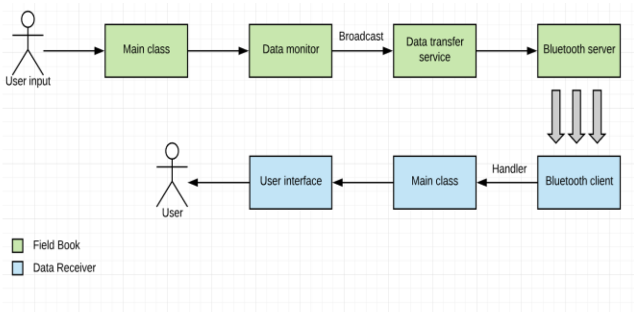

Figure 7. Active data flow

Processing on the smart phone is shown in green, and processing on the smart glasses is shown in blue.

4.2 Passive data flow

Passive data flow means that the data update originated from user input to the smart glasses. A user issues a voice command or presses an input button on the glasses, and the voice command receiver module or event handler, respectively, accept it. After analysis, the application generates a text command and calls the Bluetooth client to send it out. On the smart phone side, the Bluetooth server reads the message, and sends a command to the main class. The main class updates the user interface based on the command received. Because the user interface changes, the data monitor module is called triggering a data change event. After the data transfer service receives the event, it calls the Bluetooth server to send the updated data out.

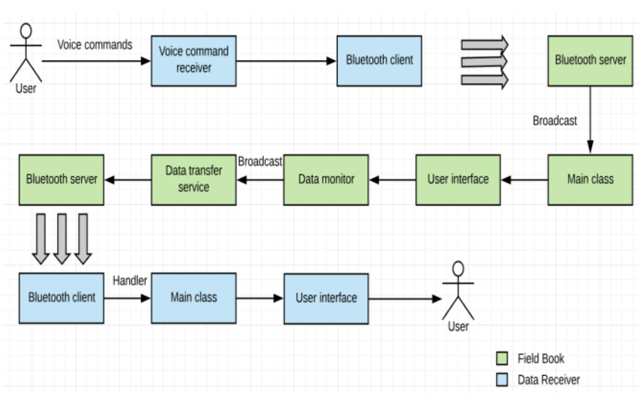

Figure 8. Passive data flow

On the smart glasses side, the Bluetooth client receives the updated data and passes the data to the main class via the data handler, and the main class displays the data on the user interface. The data flow is shown in Figure 8.

4.3 Analysis

Changes to the existing Field Book app imposes very little overhead with respect to processor and memory as shown in Figure 9. As expected, Field Book requires more resources than the Data Receiver on the smart glasses.

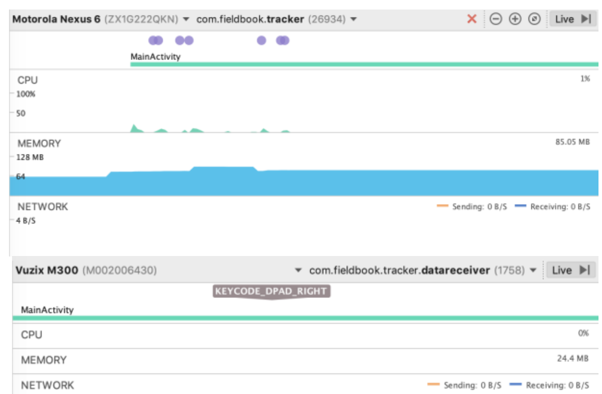

Figure 9. Field Book AR performance

For data updates from Field Book, there is almost no delay. For voice control, 1-2 seconds are needed to analyze the voice commands in the smart glasses before the command is sent to the mobile app. To create a configured Bluetooth connection is very fast as well.

5. Conclusions

This paper described the implementation of Field Book AR, a novel new project at Kansas State University to extend the functionality of the widely-used mobile app Field Book [1]. The new app incorporates the use of smart glasses to support augmented reality in Field Book. Real-time communication between Field Book on the mobile phone and the new smart glasses app for display on the Vuzix M300 smart glasses is enabled by using Bluetooth communication between the phone and glasses. The background threads are conveniently organized to enable seamless service via message passing while imposing very little overhead on the smart phone or smart glasses. All of the applications are open source, and developers are encouraged to download the source code and customize the code to meet their needs. Android is an open source platform, and Android phones are available at a low cost. As smart glasses become more economical, the advancement of similar apps will be more wide spread.

Augmented Reality (AR) enables elements of a computer-generated digital world on Field Book to be integrated with a user's perception of the physical world through smart glasses. Smart glasses have independent operating systems and can support a variety of different applications and modes of communication to support augmented reality. This paper details the development of a novel new application that extends Field Book for phenotyping and allows agronomists to interact with the app. This enables agronomists to efficiently collect phenotypic data.

In the future, we plan to add some additional commands to enable users to issue more commands using voice commands currently implemented for manual input Field Book. In this way, users can focus on more critical observations. We also plan to add a voice notes function so that comment text fields can be filled directly using audible input from the user.

We plan to allow users to capture images using the smart glasses. Currently, users only need to twist their head to let the camera focus and then issue a voice command to take a picture. We also plan to incorporate real-time kinematic (RTK) positioning with the GPS on the cell phone to identify the current plot under analysis without imposing little overhead on the phone or glasses. The volume of data from the captured image to be downloaded via Bluetooth may become a bottleneck.

Overall, the project has been very successful and provides a convenient application for gathering phenotypic data. It fully meets the purpose of liberating a user's hands for other tasks while providing relevant data in a timely fashion.

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under NSF-BREAD Grant No. 1543958. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] T.W. Rife and J. A. Poland, "Field Book: An open-source application for field data collection on Android", In *Crop Science* **54**, 1624-1627, 2014. DOI:10.2135/cropsci2013.08.0579.

[2] Chaney Courtney, Mitchell Neilsen, and Trevor Rife, "Mobile applications for high-throughput phenotyping", in Proc. of the 30th Intl. Conf. on Computer Applications in Industry and Engineering, San Diego, CA, Oct. 2-4, 2017.

[3] Mitchell L. Neilsen, Chaney Courtney, Siddharth Amaravadi, Zhiqiang Xiong, Trevor Rife, and Jesse Poland, "A dynamic, real-time algorithm for seed counting", in Proc. of the 26th Intl. Conf. on Software Engineering and Data Engineering, San Diego, CA, Oct. 2-4, 2017.

[4] Mitchell Neilsen, Shravan D. Gangadhara, and Siddharth Amaravadi, "Extending watershed segmentation algorithms for high-throughput phenotyping on mobile devices", in Proc. of the 30th Intl. Conf. on Computer Applications in Industry and Engineering, San Diego, CA, Oct. 2-4, 2017.

[5] R. Silva, J.C. Oliveira, and G.A. Giraldi, "Introduction to augmented reality", Technical Report: 06/2003, LNCC, Brazil, 2003.

[6] P. Bhagwat, "Bluetooth: technology for short-range wireless apps," *IEEE Internet Computing*, vol. 5, no. 6946016, pp. 96–103, 2001.

[7] J. Santana-Fernandez, J. Gomez-Gil, and L. del Pozo San Cirilo, "Design and implementation of a GPS guidance system for agricultural tractors using augmented reality technology.", in Sensors, Vol. 10, pp. 10435-10447, 2010.

[8] R. Azuma, "A survey of augmented reality", In ACM SIGGRAPH, 1-38, 1997.

[9] Vuzix, "Installing the M300 SDK in Android Studio", Vuzix Developer Center Documentation, https://www.vuzix.com/Developer/, 2017.