## Revision Sign-off

By signing the following, the team member asserts that he/she has read all changes since the last revision and has, to the best of his/her knowledge, found the information contained herein to be accurate, relevant, and free of typographical error.

| Name | Signature | Date |
|---|---|---|
| Josh Alvord | | |
| Alex Grosso | | |
| Jose Marquez | | |
| Sneha Popley | | |
| Phillip Stromberg | | |
| Ford Wesner | | |

## Table of Contents

# 1. Introduction

## 1.1 Purpose

The purpose of the Appropriate Gesture project is to develop a device-independent, accelerometer-based gesture recognition system. The system will allow for training of new gestures, tuning of the recognition parameters, and verification of recognized gestures. A simple game will be used as a proof-of-concept.

## 1.2 Scope of Objectives

The Appropriate Gesture project will provide a framework for mobile devices to connect into for gesture training and recognition. We will provide one plug-in (for Sun SPOTS) for this framework. Appropriate Gesture will support interaction from up to four users simultaneously, as well as provide consistent and correct recognition based upon the previously trained gestures from these users.

## 1.3 Background

Appropriate Gesture is closely based on a multitude of other projects such as the Wiigee project by Benjamin Poppinga, Thomas Schlomer, Niels Henze, and Susanne Boll at the University of Oldenburg and OFFIS Institute for Information Technology in Germany and Live Motion, a commercial gesture training and recognition software developed by AiLive. We draw from these and other such projects their use of a Hidden Markov Model (HMM), which provides an effective mathematical method for gesture training and recognition. While these projects are useful as references and models for implementation, Appropriate Gesture will be a completely independent framework.

## 1.4 Document Organization

This document is divided into multiple sections, with each section containing relevant information related to the Appropriate Gesture project.

**Section 2** – Definition of Terms: Defines any technical terms or abbreviations used in this document.

**Section 3** – Project Overview: Describes how the system handles data flow and how the user interacts with the system.

**Section 4** – Resource Requirements: Describes any software, hardware, or personnel requirements needed in development and execution of the project.

**Section 5** - Project Management Approach: Defines the staff of the project, their responsibilities, how the project was setup, and how scheduling and tracking was handled.

## 2. Definition of Terms

JSR-82

Java API for Bluetooth communication. There are several implementations of JSR-82. The BlueCove project is an open source and well-developed option for those wanting to use a variety of different Bluetooth stacks and operating systems.

Hidden Markov Model

A doubly stochastic (as opposed to deterministic) math model being used to represent a gesture. Constructed HMMs are then used in recognition. “A statistical model in which the system being modeled is assumed to be a Markov process with unobserved state.”

K-means Clustering

Method of cluster analysis which aims to partition n observations into k clusters in which each observation is clustered with the nearest mean.

Sun SPOT

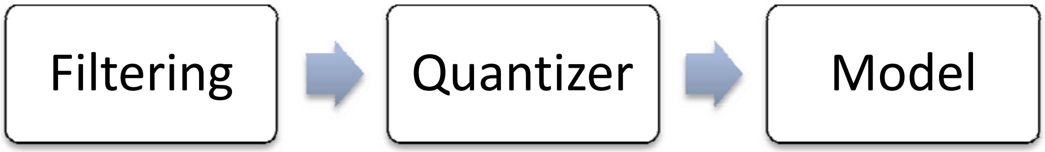
Sun Small Programmable Object Technology:  The wireless motes developed by Sun Microsystems we are using in Appropriate Gesture.  Contains a 180MHz 32-bit processor, 512K RAM, and a variety of sensors including a three-axis accelerometer.

# 3. Project Overview

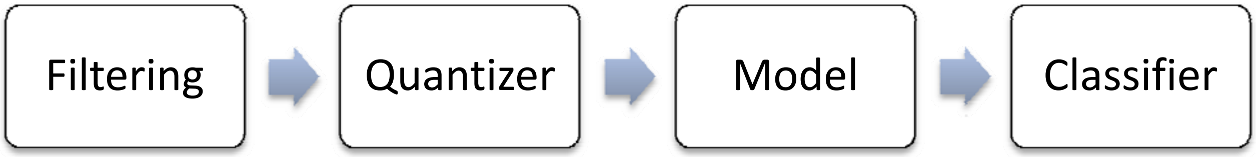
## 3.1 System

Our system is based on the Wiigee interface and takes inspiration from its system pipeline. We utilize two pipelines: one for training and one for recognition.

Training



Recognition



The different components of the pipeline are:

- **Filtering**: The accelerometer data obtained from mobile devices may contain noise in the form of vectors that do not contribute significantly to the gesture or multiple adjacent similar vectors. Such vectors are removed with the application of two filters: the idle state filter and directorial equivalence filter. Idle state is a simple threshold filter while directional equivalence compares adjacent vectors to test for similarity.

- **Quantizer**: The k-means algorithm (with k =14) is used to codify (or quantize) the filtered vectors during training so as to reduce the amount of effective information coming in from accelerometers. The 3-D vectors are clustered with k suitable centers to form a “codebook.” This is done to ensure a workable HMM. In recognition, the incoming gesture is translated into a series of the codebook vectors so as to compare against any given trained HMM (a trained gesture).

- **Model**: During training, codified observations (where observations would be training gestures) are used to form a left-to-right Hidden Markov Model (HMM) that represents the gesture. The HMM must be initialized by default, and it is then iteratively optimized according to each input training gesture. This optimization is done utilizing a modified Baum-Welch algorithm. In recognition, the constructed model is used to recognize a gesture in combination with the classifier.

- **Classifier**: A Bayes-Classifier is used in recognition to compare a test gesture to the trained data. In this case, the most probable trained gesture (above a certain threshold to detect non-matches) is identified as the performed gesture.

Additionally, the users will have the option of saving the trained data in the form of HMMs in a text file. The format of this text file will be one familiar to the program. That is to say, if in a future instance the user wishes to retrieve the data, opening the file in the created framework will be possible.

## 3.2 Modes

The user interface of the project will represent four stages. These stages will allow the user to experiment with, implement, and test changes in parameters. The stages are:

- **Training**: This is the gesture training interface. In this interface, the user will be able to select which filters will be applied (idle state, directorial equivalence, etc). The user will also be able to see the 3-D accelerometer data graphed in the interface. Additionally, the user will be able to modify the parameters for the k-means algorithm. Finally, the user will be able to select the sampling rate from the mobile devices in question (if they support such functions). The interface will also give the user the ability to load/save training sets into the system.

- **Recognition**: The recognition interface allows the user to validate the accuracy of their training. The interface shows the user the computed probabilities for select steps in computation as well as returning the most probable gesture from the training data.

- **Evaluate**: The evaluation mode has the role of enabling the user to monitor the state of the hardware and software system. The hardware system monitoring includes the ability to view the connectivity of mobile devices, whether devices are on or off, as well as if the devices are communicating data at the expected rate, etc. Similarly, the software system monitoring includes the ability to view the status of the framework, the training set being utilized, previous test data, etc. This interface will help the user ensure that the system is functioning normally as well as allow monitoring of the performance and efficiency of the system pipelines (algorithms, classifying, etc…).

- **Demo**: Gesture recognition has a multitude of applications, and the demo stage will let the user apply the trained gestures to an interactive interface. The purpose of the demo is to provide a useful but entertaining way for the user to test the system.

# 4. Resource Requirements

## 4.1 Sofware

Team Better Recognize will need a set of essential software tools in order to create a product that is not only well-designed, but delivered on-time as well. The following list is a non-exhaustive list of necessary programs.

- Eclipse Java IDE
- Subversion Server and Client
- Netbeans IDE (for initial Sun SPOT development)
- Microsoft Office including Project and Visio
- Adobe Creative Suite (for web design and graphics)
- Windows XP or newer
- WIDCOMM Bluetooth stack (for Wiimote and other Bluetooth device development)

## 4.2 Hardware

Team Better Recognize will require the following hardware components to ensure proper development and testing.

- PCs and/or Mac computers
- Bluetooth 2.0+EDR Dongles
- Sun SPOTs including base stations
- eDemoBoard and Projector

## 4.3 Constraints

- Time constraints
- Hardware performance constraints

# 5. Project Management Approach

## 5.1 Start Up

### 5.1.1 Resource Acquisition & Setup

We created a support environment for ourselves in the lab by installing Eclipse, a Subversion and web server, Microsoft Project, Adobe Creative Suite 4, and Camtasia for documenting our process for future software engineering groups looking to get started quickly.

The team started out with Sun SPOTs that had been used by the previous software engineering class. The previous team left documentation on how to get Sun SPOTs installed. We found, however, that the Sun SPOTS presented difficulty in set-up within an environment where students do not have administrator privileges.

### 5.1.2 Staff

The following are senior students at Texas Christian University.

**Phillip Stromberg** is a computer science major.
**Josh Alvord** is a computer science and mathematics major.
**Alex Grosso** is a computer information technology major.
**Jose Marquez** is a computer science & Mathematics major.
**Sneha Popley** is a computer science and mathematics major.
**Ford Wesner** is a computer information technology major.

In addition to the students above, the team sponsor is TCU professor **Dr. Donnell Payne**.

### 5.1.3 Staff Training

The team set out immediately upon the project’s inception to understand as much about the mathematics and code behind gesture recognition as possible. We looked to similar projects for guidance and found the Wiigee project to be very similar to what we were trying to accomplish.

In addition to training relating to gesture recognition, the team became familiar with components of our project support environment, specifically Microsoft Project, Adobe Creative Suite, and Camtasia.

## 5.2 Work Planning

### 5.2.1 Milestones & Deliverables

#### *5.2.1.1 Project Support Environment #1*

Date: September 21, 2009