

Project Plan

FROG Recognizer of Gestures

Team Better Recognize
Version 1.2
November 10, 2009



Revision Sign-off

By signing the following, the team member asserts that he/she has read the entire document and has, to the best of his or 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		

Revision History

The following is a history of revisions of this document.

Document Version	Date Edited	Changes
Version 1.0	10/27/09	Initial Draft
Version 1.1	11/03/09	Standardizing content layout and minor revisions
Version 1.2	11/10/09	Content and schedule revisions

Table of Contents

Revision Sign-off	i
Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope and Objectives	1
1.3 Background	1
1.4 Document Organization	2
2. Definition of Terms.....	3
3. Project Overview	4
3.1 System	4
3.2 Modes.....	5
4. Resource Requirements	7
4.1 System	7
4.2 Hardware	7
4.3 Constraints.....	7
5. Project Management Approach.....	8
5.1 Start Up	8
5.1.1 Resource Acquisition and Setup.....	8
5.1.2 Staff.....	8
5.1.3 Staff Training.....	8
5.2 Work Planning.....	9
5.2.1 Milestones and Deliverables	9
5.2.2 Project Schedule.....	10
5.2.3 Roles & Responsibilities.....	10
5.3 Project Control	11
5.3.1 Project Meetings, Reporting, and Communication.....	11
5.3.2 Schedule Tracking and Review	12
5.3.3 Requirements Control	12
5.3.4 Quality Control	12
5.4 Risk Assessment.....	12

1. Introduction

1.1 Purpose

The purpose of this document is to formally outline the project FROG as well as to define its structure, basic architecture, and development process. In addition, this document will overview the goals of FROG, discuss resource requirements, outline a project management approach, and provide a schedule of project milestones and deliverables.

1.2 Scope and Objectives

The objective of the FROG project is the creation of an independent gesture recognition framework intended for use in motion-based recognition research. The FROG project shall consist of a Hidden Markov Model-based training and recognition framework supported by an intuitive and research-oriented GUI. The framework will be built in a way that allows additional plug-ins for new devices to be created and integrated easily. Frog will include one plug-in at launch, a plug-in for Sun SPOTs.

1.3 Background

Gesture recognition in computer science refers to the identification of hand gestures or facial expression through the use of mathematical algorithms. Gesture recognition can be seen as a bridge between machines and humans, offering a richer interface than traditional input methods such as text interfaces or a GUI. FROG is primarily concerned with gesture recognition through identifying arm motions using 3D accelerometers on mobile devices. Initial references for FROG were the Wiigee gesture recognition project and the LiveMove project from AiLive. The Sun SPOT is the device chosen for FROG as it contains a three-axis accelerometer and IEEE 802.15.4 radio, both easily programmable in Java.

1.4 Document Organization

This document is divided into multiple sections, with each section containing relevant information related to the FROG 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 Control: Defines the staff of the project, their responsibilities, how the project was setup, and how scheduling and tracking was handled.

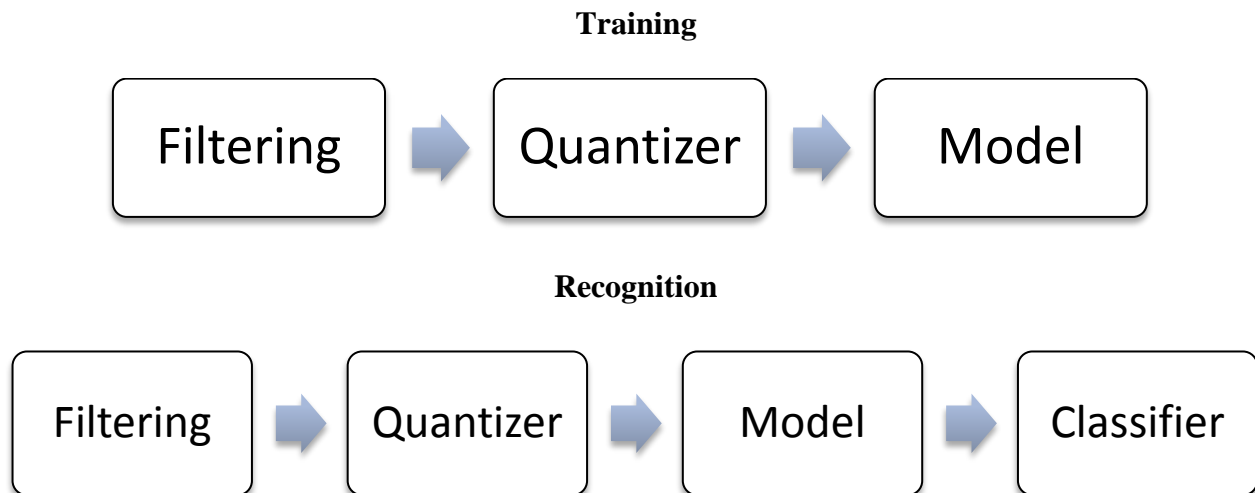
2. Definition of Terms

Accelerometer	An instrument for measuring acceleration. In particular, a 3D accelerometer measures acceleration in three dimensions.
Gesture	A continuous combination of motions made by an individual (usually with the hands) that are related and meaningful as a whole. Gestures are the entities which shall be modeled as well as recognized by the FROG project.
Hidden Markov Model	A doubly stochastic (as opposed to deterministic) math model being used to represent a gesture through the training process. 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. This is a local optimization algorithm.
Sun SPOT	Sun Small Programmable Object Technology: The wireless motes developed by Sun. Contains a 180MHz 32-bit processor, 512K RAM, and a variety of sensors including a three-axis accelerometer. Sun SPOTs communicate using a low-power IEEE 802.15.4 radio.
Training Instance	A training instance is a single motion of the mobile device by a user representing a gesture to be trained.
Training Session	A training session is a collection of training sets created by a user. A session can be saved in a file format in order to perform recognition. A training session can be thought of as a project file containing representations of multiple gestures.
Training Set	A training set is a sequence of training instances created by the user in order to train a gesture. A training set can be thought of as the complete set of data used to create an HMM representation of a gesture.

3. Project Overview

3.1 System

The method used for gesture recognition in the FROG project involves an established process of manipulating data. The system will utilize two pipelines: one for training and one for recognition (as pictured below). Both pipelines employ a filtering stage to reduce the information received from a connected mobile device. Within the training pipeline, user input will be used to construct a quantizer and Hidden Markov Model for each training set. In the recognition pipeline, a filtered input gesture will be passed through the components constructed in the training pipeline. That is, for each training set constructed in the training session, an input gesture will be translated using the corresponding quantizer and then evaluated for recognition probability using a Bayesian classifier and the matching Hidden Markov Model.



The different components of the pipeline will be:

- **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. These vectors can be removed with the application of filters such as the idle state filter and directorial equivalence filter. Idle state is a simple threshold filter that eliminates non-movements while directional equivalence compares adjacent vectors to test for similarity and drops any that change minimally.

- **Quantizer:** The k-means algorithm will be 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 will be clustered with k suitable centers to form a “codebook.” This is done to ensure a workable HMM. In recognition, the incoming gesture will be translated into a series of codebook vectors so as to compare against any given trained HMM (a trained gesture).
- **Model:** During training, codified observations (that is, the codified series of vectors in a training instance) will be used to form a left-to-right Hidden Markov Model (HMM) that represents the gesture. The HMM must be initialized by default and will then be iteratively optimized according to each input training instance. This optimization will be done utilizing a modified Baum-Welch algorithm. In recognition, the constructed model will be used to recognize a gesture in combination with the classifier.
- **Classifier:** A Bayes-Classifer will be 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) will be identified as the performed gesture.

3.2 Modes

The structure and user interface of the project will represent the four modes described below. These stages will allow the user to experiment with, implement, and test changes in parameters. All modes will contain some form of filterable console output as well as a connection framework for user(s). The modes’ unique attributes, however, will be:

- **Training:** This is the gesture training interface, controlled by a single user at a time. The user will train gestures within this mode – training sets – with several training instances. 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. 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 sessions into the system.
- **Recognition:** The recognition interface will allow up to four users to simultaneously validate the accuracy of their training. The interface will show the user(s) the computed probabilities for select steps in computation as well as return the most probable gesture from the training data.

- **Evaluate:** The evaluation mode will have the role of providing a single user with the ability to evaluate the gesture recognition performance of the FROG project. In this interface the user will be able to load a training session with which to test recognition. The user will need to enable which gestures in the session data are to be evaluated and set the number of times for the gesture to be requested. The user will have the option of having the gestures prompted sequentially (i.e., if the user wants ten evaluations, ten squares will be requested, then ten circles, etc.) or in a random fashion (but still limited by the aforementioned evaluation number). The interface will continuously record the number of successes, number of failures to recognize, and number of errors in recognition (recognizing as an incorrect gesture) and display this in a graphical format. Evaluate mode will also display data concerning operation and efficiency of the recognition system.
- **Demo:** Gesture recognition has a multitude of applications, and the demo mode will let up to four users apply the trained gestures to an interactive interface. The purpose of the demo will be to provide a useful but entertaining way for the user(s) to test the system.

4. Resource Requirements

4.1 System

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. The following is a non-exhaustive list of necessary programs.

- Eclipse Java IDE v.3.5 (Galileo)
- Subversion Server and Client
- Netbeans IDE v.6.7.1 (for initial Sun SPOT development)
- Microsoft Office 2007 including Microsoft Project 2007 and Microsoft Visio 2007
- Adobe Creative Suite 4 (for web design and graphics)
- Windows XP or newer
- WIDCOMM Bluetooth stack (for Wiimote and other Bluetooth device development)
- Java Runtime Environment 6 update 17
- Sun SPOT Manager
- Sun SPOT SDK 5.0
- Apache Ant 1.7.1
- Camtasia Studio 5

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

4.3 Constraints

Time

- The team is limited to the academic year ending on May 11, 2010.

Resource Performance

- The computers being used to run and test the software are dated and may need to be upgraded for better performance.

5. Project Management Approach

5.1 Start Up

5.1.1 Resource Acquisition and Setup

A project support environment was created in the lab by installing Eclipse 3.5, a Subversion and web server, Microsoft Project 2007, Adobe Creative Suite 4, Microsoft Visio 2007, NetBeans 6.7.1, and Camtasia Studio 5. Camtasia Studio 5 and Adobe Creative Suite 4 were installed to aid project deliverables but are not necessary for the creation of FROG. Additionally, the Sun SPOT SDK and SPOT Manager were installed to allow for the familiarization of SPOT mote and base devices. The mote is used within the project as a 3-D accelerometer and the base being used to collect the vectors on the mote.

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 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 processes behind gesture recognition as possible. The team looked to related projects for guidance and found the Wiigee project to be similar to what this project is trying to accomplish.

In addition to training related to gesture recognition, the team became familiar with components of the project support environment, specifically Microsoft Project 2007, Microsoft Visio 2007, Adobe Creative Suite 4, and Camtasia Studio 5.

5.2 Work Planning

5.2.1 Milestones and Deliverables

Project Support Environment #1

Date: September 21, 2009

The objective of the first Project Support Environment will be the installation and study of required software for the FROG development environment.

Project Proposal

Date: October 15, 2009

The Project Proposal will be formally documented by this date and presented to the instructor.

Project Plan v1.0

Date: October 27, 2009

The first version of the Project Plan will be completed by this date. The document will be edited throughout the project.

Software Requirements Specification v 1.0

Date: November 3, 2009

The Project Requirements will be solidly defined, allowing the team to move forward with confidence.

Design Documentation v 1.0

Date: November 16, 2009

The design of FROG will be defined by this date in a deliverable design document.

Iteration 1

Date: December 8, 2009

The first iteration will involve the development of a prototype GUI, a filtering and k-means system, and SPOT communication testing.

Iteration 2

Date: February 2, 2010

The second iteration will involve refining components from iteration one as well as a completing training mode and the foundation for recognition mode.

Iteration 3

Date: March 2, 2010

The third iteration will involve refining components from iteration two as well as completing recognition and evaluation modes.

Iteration 4

Date: March 11, 2010

Iteration four will involve refining components from iteration three as well as completing the demo mode.

Developer's Guide

Date: March 16, 2010

A developer's guide will be completed by this date. This guide will be intended for use by developers who wish to modify FROG in any way or write a new plug-in for a previously unsupported device.

User Manual

Date: March 16, 2010

A user manual will be completed by this date. The manual will be meant as a complete guide to FROG, from software required to a walkthrough of user interaction with the system.

Code Complete

Date: March 11, 2010

By this date the team will have completed all iterations and will have a complete working version of FROG.

Code Freeze

Date: March 28, 2010

By this date the team will no longer be editing code. The team will have completed the project by this point and FROG is ready to be delivered.

Final Presentation

Date: April 30, 2010

On this date the team will present the project to the public. In addition to the presentation, the project DVD will be ready for distribution.

5.2.2 Project Schedule

See attached Gantt chart

5.2.3 Roles & Responsibilities

These are the general areas of responsibility for each group member. While each member is finally responsible for his or her area, they are expected to assist in areas outside their own.

Phillip Stromberg: Project Manager

- Primary customer communication
- Assistance with general issues
- Ensure project is on schedule

Josh Alvord: Back-end

- Study and presentation of algorithms
- Back-end coding
- Document editing

Alex Grosso: Quality Assurance

- Testing and quality assurance
- Maintenance of website
- Develop test plans

Jose Marquez: Graphics and User Interface

- Graphical User Interface: design and implementation
- Team graphics and artwork
- Maintenance of document templates

Sneha Popley: Back-end

- Study and presentation of algorithms
- Back-end coding
- Mobile device communication

Ford Wesner: Front-end

- Schedule Planning
- Integration of backend and interface
- General Quality Control

5.3 Project Control

5.3.1 Project Meetings, Reporting, and Communication

Project meetings occur twice a week. At the beginning of the meetings, members take turns reporting what they have accomplished since the last meeting. Tasks to be completed in the coming week are assigned, and it is verified that the whole group understands the current project status and the closeness and urgency of any impending deadlines.

Weekly activity reports are collected at the beginning of meetings and posted to the project website. This allows team members and the team sponsor, Dr. Payne, to monitor individual and project progress.

Each team member is responsible for communicating with the rest of the team, even outside of meetings. If a member is going to change something about the project (perhaps the code or documentation), they should let the others know it by e-mail if at all possible. The Subversion server maintains the versioning and integrity of documents but it is certainly not a replacement for team communication.

5.3.2 Schedule Tracking and Review

The project schedule will be maintained through weekly activity reports and plotted on a Gantt chart. If a task is not being completed, the rest of the team will take steps conducive to productivity. The team shall either motivate the responsible person or move the task responsibility to a different member.

5.3.3 Requirements Control

Ensuring the project is to specification will be accomplished through walkthroughs, demos, requirement reviews, and in-class meetings. In addition, the team will do everything possible to create requirements in a group environment to minimize discrepancies in understanding of said requirements.

5.3.4 Quality Control

Each stage of FROG will be rigorously tested and re-tested to validate requirement functionality and safety. Team members will be appointed (specifically including member(s) not involved with the individual component's development) to oversee testing and ensure deliverability. After strenuous testing deliverables will then be submitted for user acceptance testing to ensure satisfactory results.

5.4 Risk Assessment

Risk	Probability / Severity	Mitigation Strategy
Performance issues on SPOT or computer	Medium/High	Re-factor code for higher performance
Tasks more complicated than initially thought, deadline in risk of being missed	Medium/High	Re-organize group, double efforts
Wiigee Project does not work as expected	Medium/High	Devote extra time to understanding existing project
Team Member sick or excessively absent	Low/High	Early documentation will allow others to pick up the slack
Server failure causes data loss	Low/High	Make frequent backups of server data
Requirements must be changed	Medium/Medium	Early documentation will allow more flexibility later