SLIK

Sign-Language Interpreter for Kinect

Phase 3
Software Requirements Specification

Team 48
INDEX

Table of Figures 3

Introduction 3

Objective 3

Assumptions and Limitations 3

Overview of System Architecture & Module Descriptions 4

S.L.I.K. Core Module 5

S.L.I.K. Kinect Interface 7

User Interface Module 8

User Interface Screens 9

Gesture Database 13

Preliminary Integration Thread 14

Change Request Form 15

Cross Reference Listing 16
Table of Figures

Fig. 1 - A functional block overview of the SLIK system.
Fig. 2 - Core Module data flow diagram.
Fig. 3 - Hardware Module / Kinect Interface data flow diagram.
Fig. 4 - User Interface data flow diagram.
Fig. 5 - Gesture Database data flow diagram.
Fig. 6 - The preliminary integration thread for the SLIK system.

Introduction

Sign Language Interpreter for Kinect (S.L.I.K.) is a software sign language interpreter that utilizes the Microsoft Kinect sensor. The purpose is to bridge the gap between people who can only communicate through sign language with people who do not know sign language. The S.L.I.K. is a Windows app that can be deployed on any device that can interface with the Kinect.

Objective

Our objective is to deploy an easy to use system that a speech impaired person can use to communicate with a person who does not know sign language. Sign language is a very different form of communication compared to speech. This makes it extremely hard to learn, especially later in life. The S.L.I.K. will bridge that gap between people who need sign language to communicate and those who do not know sign language and are unable or unwilling to learn. Current systems on the market do offer similar solutions, however those use proprietary software and hardware while costing double or triple the price. The S.L.I.K. is a windows app that will run on any Kinect compatible device, whether it be a tablet or a PC.

Assumptions and Limitations

- The system will run on any windows device with a Kinect interface.
- Users must have enough familiarity with computers to install and setup the system or have someone set it up for them.
- Accuracy of the system will depend on the complexity of gestures and the clarity with which the signer signs.
- The system will require the Kinect sensor to operate, thereby limiting portability to small distances.
- Accuracy will also suffer in case of noisy sensor data, caused by irrelevant motion in front of the sensor.
Overview of System Architecture & Module Descriptions

![Diagram](image)

**Fig. 1** - A functional block overview of the SLIK system.

**S.L.I.K. Core**
Where the bulk of the processing is done. Gesture translation, text and speech generation are all done here. Also the configuration data is saved here.

**Kinect Sensor**
Developed by Microsoft. Records video and infrared data to be used to track and translate signs from user. Video data is also fed directly into the User Interface module to display directly to the user.
Gesture Database

A global database for all S.L.I.K. systems that contains translation data for all sign language dialects.

User Interface

Will have two screens as discussed in the requirements. A main screen with a video feedback and text output. And a configuration screen that is mostly taken care of within the core module.

All of these modules will be explained in greater detail later in this document.

SLIK Core Module
Hardware Module

This is the interface with the Microsoft Kinect sensor. It is explained in greater detail in the next section.

Gesture Interpretation

This module receives movement data from the hardware module, and checks it with the databases. If there is a match the verified gesture info is sent forward.

Text Generation

Generates ASCII text based off the verified gestures, this is more of a formality, as the interpreted text is fetched from the database, this module formats it correctly for the GUI.

Speech Generation

If the toggle from the GUI is enabled, this module uses text to speech to generate the speech that will be sent to the devices output speakers.

Configuration

We decided to put the configuration in the core module because it manipulates a database. So once the user enters the configuration screen in the UI, control is transferred to this module inside the core.
The SLIK system relies on the Microsoft Kinect to appropriately capture user-input. Image data is then captured and processed as a video stream, simultaneously containing 3-dimensional hand and finger coordinates which constitute the gesture data. While the video data is acquired by the Kinect’s video camera, the gesture data is acquired by its infrared image-capturing capabilities. For the SLIK system, the Kinect will be tracking the user’s hand and arm movements, as well as the movements of their fingers. In order to do so, the captured video and gesture data are separated and optimized by image normalizing software (Image Data Normalizer submodule) so that further processing by later modules can be done.
Captured video data is routed to a high frequency motion analyzer that analyzes the motion of the user’s body from the video. Specifically, this component of the Hardware Module targets the user’s arm movements relative to the rest of their body, in order to accurately recognize signs of different context and/or dialect. The gesture data, however, is fed into a submodule (Hand-Finger Geometry Analyzer) that analyzes the motion of the user’s fingers relative to their respective hands. In other words, both of these submodules require only 1 of the 2 input streams, ignoring one another’s. Each of these submodules then sends a descriptor to a processing submodule (within the Gesture Interpretation Module) that queries the Gesture Database for a matching gesture. The Gesture Database (structure is detailed in a later section) returns to the Gesture Interpretation submodule either the correct gesture-equivalent text or a flag which signals that the gesture-equivalent text is user-defined. Once the correct textual representation of the captured gesture is processed, it is sent to the Text Generation module, which simply formats the text so that it can be appropriately displayed on the SLIK User Interface (also detailed in a later section).

User Interface
Fig. 4 above gives a brief layout of the user interface for the SLIK system. Here, the submodules of the User Interface are visible; the list below describes the core functions of each.

**Windows/OS Interface**
This module interfaces with the OS. Utilizing device drivers and Windows features for such things as audio output and user input.

**Screen Display**
This module renders the output screen and refreshes it at a constant rate and upon user input or software interrupt.

**Video Display**
Used to generate the video image that the screen display will use to update the video displayed to the user from the Kinect.

**Event Handler**
Generates an interrupt based off user input from the Windows/OS interface. Though it is not clear in the diagram, this module can call other modules within the user interface.

**Configuration Screen**
Displays the configuration screen to the user and when called by the event handler, it transfers control to the Core Module. Although all the user input is handled on the User Interface end.

**Audio Out**
Outputs audio to the Windows/OS interface to be read aloud. Controlled by the toggle button, i.e. the user.

**Volume Up/Down**
Controls the volume. Called by the event handler.
User Interface Screens

Main Screen
Gesture Recognition On Live Screen

SLIK

Hello!

Configure
Save

Speech - ON
Volume

STOP
Configuration Screen

Configuration

Sign Dialect: ASL
Text Language: US English
Speech Language: US English
Voice: Male/North American

Add Sign

Sign Translation

Please add the desired translation for the above sign here..
Gesture Database

Fig. 5 - Gesture Database data flow diagram.

This concludes our discussion of the main modules and the user interface of the system. Following below is the outline of a preliminary integration thread around which we will design the schedule and allocate a rough timeline for the project. We also have a change request form template that the customer will be required to fill out any time a change to the initial software specifications is to be made.
The diagram above outlines an integration thread for the project. This is a barebones view of the system. Only the components essential for the most basic operation of the product have been included.

The Geometry Analyzer module is essential for the successful recognition of signs and generating the gesture data that the Query Processor will use to search through the database of pre-configured signs. The Query Processor will then generate a text packet which is then formatted and displayed in real time to give feedback to the user.

Note that features such as support for multiple dialects of sign language and multiple languages for speech output, as well as the feature for user-defined gestures and saving generated text to a file have not been included in the integration thread as they are not essential to have a working product. Work on these features can ensue once the pre-configured database is in place and the Geometry Analyzer and Query Processor are working fine, and the work can continue in parallel with the text generation module or in subsequent versions of the product as suggested by the customer.
CHANGE REQUEST FORM

I. Requestor’s Information
   a. Name: ________________________________________________________________
   b. Business Role: _______________________________________________________
   c. E-mail: ____________________________ Contact: ___ - ___ - ___

II. Change Details
   a. Type of Change:
      □ Feature addition   □ Feature removal   □ Feature modification
   b. Description of Change -
      ___________________________________________________________________
      ___________________________________________________________________
      ___________________________________________________________________
   c. Estimated impact on project budget –
      □ High   □ Medium   □ Low
   d. Estimated impact on project schedule –
      □ High   □ Medium   □ Low
   e. Priority –
      □ Mandatory   □ Desirable   □ Non-essential

III. Change Review (to be filled in by project manager)
   a. Module(s) affected: __________________________________________________
   b. Is the change feasible? ______________________________________________
   c. Revised budget: ______________________  Revised schedule: ________________

IV. Change Status
   □ Approved   □ Declined

Customer Representative: __________________________________________________
Project Team Representative: _______________________________________________

Date: ___ / ___ / ___
### CROSS REFERENCE LISTING

<table>
<thead>
<tr>
<th>Item Number</th>
<th>System Spec Location</th>
<th>SRS Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Recognize Gestures and Generate Captions</td>
<td>Functional Requirements 1</td>
<td>S.L.I.K. Core Module – Geometry and Motion Analyzer, Query Processor, Text Validator</td>
</tr>
<tr>
<td>2. Multiple Sign Language Dialect Support</td>
<td>Functional Requirements 1.1</td>
<td>S.L.I.K. Core Module, Gesture Databases</td>
</tr>
<tr>
<td>3. Multiple Languages for Speech Output</td>
<td>Functional Requirements 1.2</td>
<td>S.L.I.K. Core Module – Text Validator</td>
</tr>
<tr>
<td>4. Camera Preview and Speech Output</td>
<td>Functional Requirements 2</td>
<td>S.L.I.K. Core Module, UI Module</td>
</tr>
<tr>
<td>5. Configuration</td>
<td>Functional Requirements 3</td>
<td>S.L.I.K. Core Module, UI Module – Configuration Module</td>
</tr>
<tr>
<td>6. User Interface</td>
<td>Functional Requirements 4</td>
<td>UI Module</td>
</tr>
<tr>
<td>7. Start/Stop Recording</td>
<td>Functional Requirements 4.1</td>
<td>UI Module – Text-to-Speech Module</td>
</tr>
<tr>
<td>8. Speak Aloud</td>
<td>Functional Requirements 4.2</td>
<td>UI Module – Audio Out Module</td>
</tr>
<tr>
<td>9. Volume Bar</td>
<td>Functional Requirements 4.2.1</td>
<td>UI Module – Volume Module</td>
</tr>
<tr>
<td>10. Settings</td>
<td>Functional Requirements 4.3</td>
<td>UI Module – Configuration Screen</td>
</tr>
</tbody>
</table>