GERMAN UNIVERSITY IN CAIRO

Controlling Home Devices with Microsoft Kinect

by

Khaled Osman

A thesis submitted in partial fulfillment for the
degree of Bachelor of Science

in the
Engineering
Computer Science

April 2013

Declaration of Authorship

I, Khaled Salah, declare that this thesis titled, 'Controlling Home Devices with Microsoft Kinect' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ________________________________

Date: ________________________________

"Were data hungry, we want it at our fingertips. At the same time, when it comes to our comfort and convenience, we want it easily and quickly with no distractions."

Geoff Godwin

GERMAN UNIVERSITY IN CAIRO

Abstract

Engineering
Computer Science

Bachelor of Science

by Khaled Osman

A thesis presented in the development of Microsoft Kinect and home automation technology by using Skeletal tracking, Voice recognition and Gesture recognition. . . .

Acknowledgements

...

Contents

Declaration of Authorship **i**

Abstract **iii**

Acknowledgements **iv**

List of Figures **vi**

List of Tables **vii**

Abbreviations **viii**

Physical Constants **ix**

Symbols **x**

1 Introduction **1**
- 1.1 Kinect Features 1
  - 1.1.1 Skeletal Tracking 2
  - 1.1.2 Speech Recognition 2
  - 1.1.3 Gesture Recognition 2
- 1.2 Arduino 4
  - 1.2.1 Kinect-Arduino-HomeDevice physical Connection 4
- 1.3 User Stories 4
- 1.4 Software Architecture 6

A An Appendix **7**

Bibliography **8**

List of Figures

1.1 Kinect sensor 2
1.2 Data stream 2
1.3 Skeleton Joints 3
1.4 Gesture Architecture1 3
1.5 Gesture Architecture2 4
1.6 Arduino Board 4
1.7 Kinect Arduino Connection 5
1.8 Software Architecture 6

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	$rads^{-1}$

Chapter 1

Introduction

Lately, touchless interface devices have appeared in the technology world making our lives easier. Currently the average household pays over 100 dollars a month for electricity according to the U.S. Energy Information Administration. Half of those costs are to keep the motor and compressors in operation in air conditioners and heating systems. If consumers had the ability to control their homes climate control systems remotely, they could save money without sacrificing convenience and comfort. The idea is to extend the Kinect's potential uses from gaming to be used in different regions making more and better use of it's capabilities. This is a handicapped-friendly project that provides an easy way to control home devices using Microsoft Kinect sensor and ARDUINO board by giving certain voice commands or by doing certain gestures or even through a touchless graphical interface.

1.1 Kinect Features

The kinect sensor contains of an infrared projector, a LED light indicator, a color camera, an IR camera, a Microphone array and a tilt motor with a vertical range of -27 to 27 degrees. The camera has an angular field of view of 57 degrees horizontally and 43 degrees vertically. The connection between the arduino uno board and the kinect sensor is done through a USB-Serial communication.

See Figure 1.1 Figure 1.2

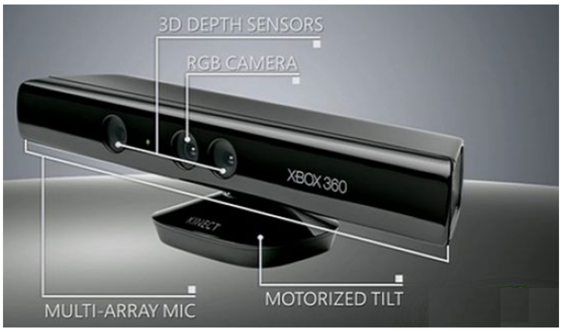

FIGURE 1.1: Kinect sensor

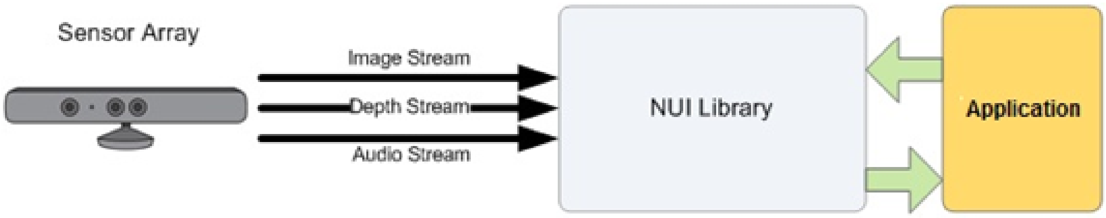

FIGURE 1.2: Data stream

1.1.1 Skeletal Tracking

The Microsoft SDK provides the player's skeleton as an array of joints of a Vector4 value each, which contains each joint's x,y,z and w values; it's x,y,z position values in 3d space and the w value that indicates the quality level (Range between 0-1)) of the position that indicates the center of mass for that skeleton which is only 1 for passive fully tracked players. The kinect sensor can detect up to 6 players at the same time, however it can only track 2 at the same time while it only stores some basic information like position values for the other 4. (See *fig.* 1.3)

1.1.2 Speech Recognition

The microphone array of the kinect sensor is responsible for it's hearing, by integrating it's features and creating grammars the kinect can be used for voice recognition.

1.1.3 Gesture Recognition

There's many ways you could build a gesture recognition system using the kinect SDK for example you can store your gestures as frame images and use image proccessing to compare them, or you can just compare the angle between certain joints but this won't

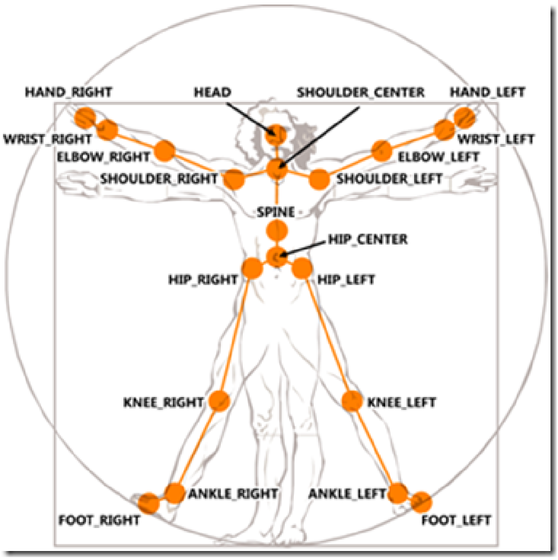

FIGURE 1.3: Skeleton Joints

be as efficient as it may detect other random gestures too, so it's better to compare joint positions relatively to eachothers or draw imagionary vectors between the joints and compare them to the expected directions. You can divide the gesture into smaller gesture parts/segments if needed and check if they succeed in each frame according to their topological order in a certain number of frames then the gesture succeeds. In this project, the gesture recognition system does all of the above except image proccessing as there was no gesture recording software compatible with the microsoft sdk found. See Figure 1.4 Figure 1.5

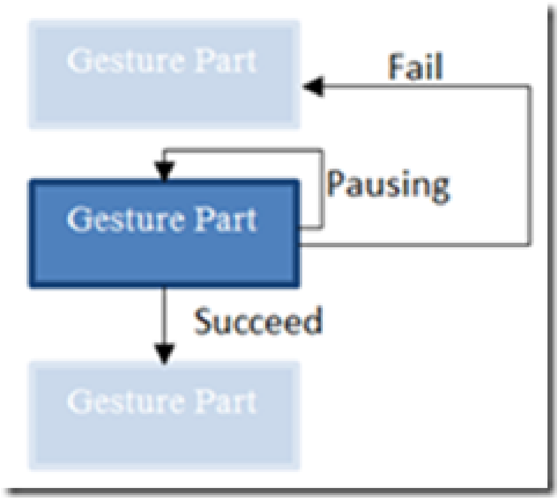

FIGURE 1.4: Gesture Architecture1

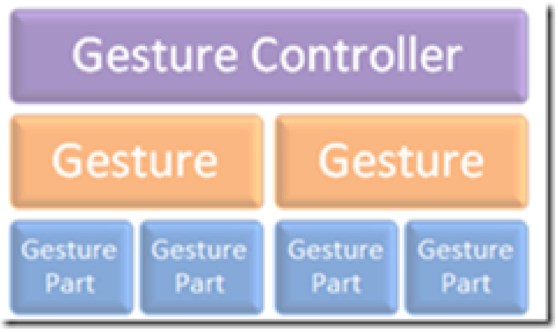

FIGURE 1.5: Gesture Architecture2

1.2 Arduino

The Arduino Uno is a microcontroller board based on the ATmega328 . It has 14 digital input/output pins of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The power pins are as follows: The ATmega328 has a 32 KB memory (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM. See Figure 1.6

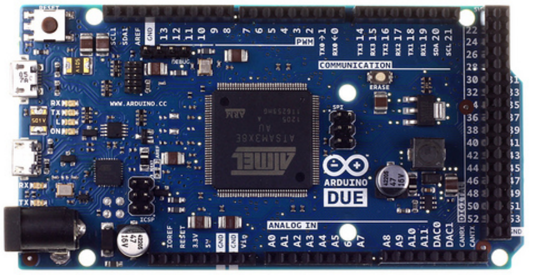

FIGURE 1.6: Arduino Board

1.2.1 Kinect-Arduino-HomeDevice physical Connection

The connection between the kinect and the arduino is handled through a USB to Serial converter and an interface handling the websockets. The connection between the arduino and the home devices is made through an electric circuit using sensors, relays, transistors, resistors, LEDS and an LCD etc.. See Figure 1.7

1.3 User Stories

The user stories and the program features are as follows:

- As a user, I can turn the lights on/off by entering/leaving the room.
- As a system, I should be able to recognize certain gestures and control the device accordingly.
- As a user, I can interact with a touchless interface.
- As a system, I should recognize the user's face and keep track of it.
- As a user, I can give certain voice commands.
- As a system, I should have a circuit connection between the arduino board and the device I want to control.
- As a system, I should be able to control the device and switch it on/off when needed.
- As a system, I should create an interface to allow the communication between the kinect sensor and arduino board.
- As a user, I should see pop-up screens to show feedback from the interface at certain circumstances.
- As a user, I should see an avatar which represents my distance to the kinect.
- As a user, I should see an introductory fading screen to the application.
- As a user, I can play music.
- As a user, I should see a main screen where I can edit settings and enable/disable features like face/depth/voice/skeletal tracking.

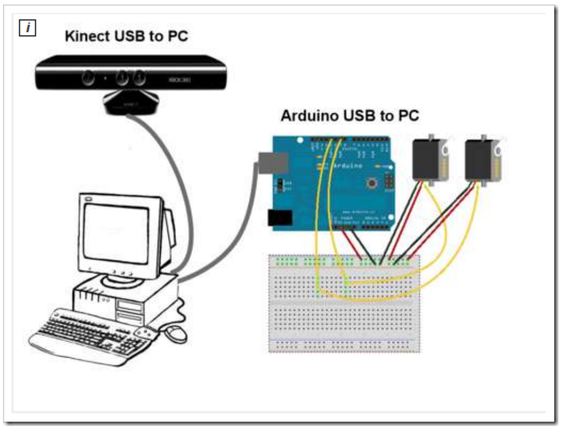

FIGURE 1.7: Kinect Arduino Connection

- As a user, I should see a screen for each controlled device where I can edit it's settings manually.
- As a system I should provide a way for the user to enter a password to be able to manage the application.

1.4 Software Architecture

The software architecture of the project is shown in the figures below See Figure 1.8

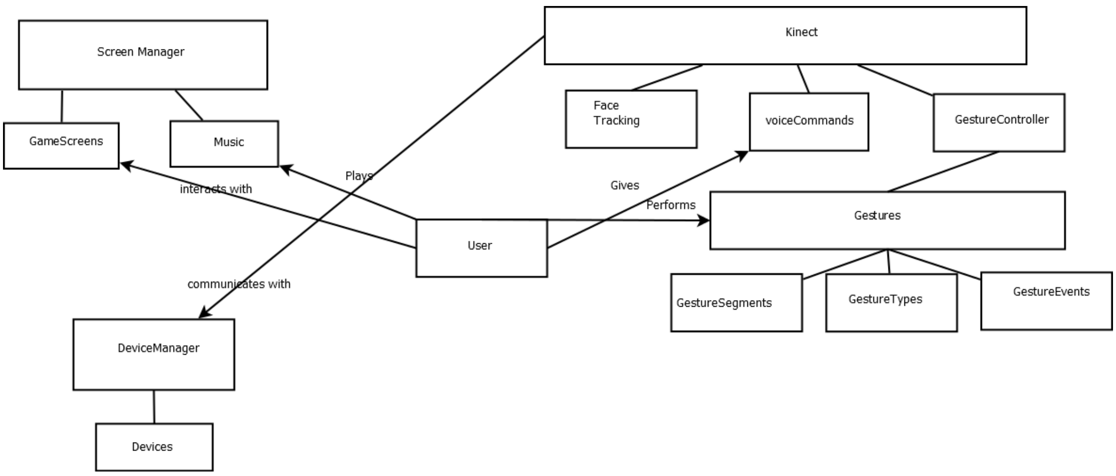

FIGURE 1.8: Software Architecture

Appendix A

An Appendix

http://msdn.microsoft.com/en-us/library/jj131025.aspx http://arduino.cc/en/Main/arduinoBoardUno http://blogs.msdn.com/b/mcsuksoldev/archive/2011/08/08/writing-a-gesture-service-with-the-kinect-for-windows-sdk.aspx Beginning kinect programming with the microsoft kinect sdk book Arduino cookbook

Bibliography