Gesture Recognition and Voice Synthesis using
Intel Real Sense

An Experimental Manual for
Design and Development of Mobile Devices

In association with Intel Collaboration Program

Designed by: Zeenat Shareef, PhD
Under the guidance of: Dr. S.R.N Reddy, HOD and Associate Professor, CSE

Department of Computer Science
Indira Gandhi Delhi Technical University for Women
Kashmere Gate, Delhi-110006
# LIST OF EXPERIMENTS

<table>
<thead>
<tr>
<th>EXP. No</th>
<th>Description of Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction to Intel Real Sense Technology</td>
</tr>
<tr>
<td>2.</td>
<td>Introduction to Visual Studio 2013 and write the steps to create a new application.</td>
</tr>
<tr>
<td>3.</td>
<td>Write the steps to control visual studio application screen through gestures using Intel Real Sense technology.</td>
</tr>
<tr>
<td>4.</td>
<td>Write the steps to demonstrate voice synthesis in Visual Studio Application using Intel Real Sense.</td>
</tr>
<tr>
<td>5.</td>
<td>Write the steps to display the different texts by showing various Gestures using Intel Real Sense Camera.</td>
</tr>
<tr>
<td>6.</td>
<td>Write the steps to control the LED on Intel Edison by gestures using Intel Real Sense 3D Camera.</td>
</tr>
<tr>
<td>7.</td>
<td>Write the steps to get the current temperature by the LM35 temperature sensor interfaced on Intel Edison and use voice synthesis capability of Intel Real Sense to speak the current temperature.</td>
</tr>
</tbody>
</table>
EXPERIMENT 1

AIM: Introduction to Intel Real Sense Technology

1.1. Introduction to Intel Real Sense Technology: It is a platform for implementing gesture-based Human-Computer Interaction techniques. It consists of series of consumer grade 3D cameras together with an easy to use machine perception library that simplifies supporting the cameras for third-party software developers.

Three camera models with distinct specifications are:

1. **Intel RealSense 3D Camera (Front F200):** This is a stand-alone camera that can be attached to a desktop or laptop computer. It is intended to be used for natural gesture-based interaction, face recognition, immersive, video conferencing and collaboration, gaming and learning and 3D scanning.

2. **Intel RealSense 3D Camera (Rear R200):** Rear-mounted camera for tablet. This camera is intended for augmented reality applications, content creation, and object scanning.

3. **Intel RealSense Snapshot:** Snapshot is a camera intended to be built into smartphones. Its intended uses include taking photographs and performing after the fact refocusing, distance measurements, and applying motion photo filters.

The Intel RealSense Developer Kit (F200 Camera) is a small, light-weight, USB-powered camera optimized for close-range interactivity. It is designed for ease of setup and portability, it includes a depth sensor and built-in dual-array microphones for capturing and recognizing voice, gestures and images.

![Fig 1: Intel RealSense 3D Camera (F200)](image-url)
1.2. Internal components of Intel Real Sense: Internal components of Intel Real Sense consists of a standard 2D camera for regular photo and video, along with an infrared camera and an infrared laser projector. The three lenses allow the device to infer depth by detecting infrared light that has bounced back from objects in front of it. The infrared parts allow RealSense to see the distance between objects, separating objects from the background layers behind them and allowing for much better object, facial and gesture recognition than a traditional camera. The infrared projector projects a grid onto the scene (in infrared light which is invisible to human eye) and the infrared camera records it to compute depth information. In addition to lenses it has microphone array. The microphone array allows localizing sound sources in space and performing background noise cancellation. The camera is equipped with a 1080p RGB sensor plus an infrared sensor and infrared laser. It uses multiple sensors to add depth to images, allowing a host of applications, from adjusting the focal point of an image to gesture recognition and augmented reality.

![Fig 2: F200 components](image)

1.3 Intel Real Sense SDK

![Fig 3: Intel Real Sense SDK](image)

The base of the components is the SDK core. One of its jobs is to manage the two types of modules: input/output modules and capability modules, representing different input modalities. These modules provide the SDK functionalities to your application. The I/O modules capture input data from your device and send that data to an output device or to the
capability modules. The capability modules include various pattern detection and recognition algorithms, like face tracking and recognition, hand tracking, gesture recognition, and voice recognition and synthesis. Another job the SDK core performs is organizing the execution pipeline. It is possible to have multiple modules contained within the pipeline at the same time, so it is essential that the pipeline have a manager.

1.4 Intel Real Sense SDK capabilities

![Fig 4: Intel Real Sense SDK capabilities](image)

<table>
<thead>
<tr>
<th>Categories of Input</th>
<th>Capabilities</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hands</td>
<td>Hand and Finger tracking</td>
<td>22-point Hand and Finger Tracking</td>
</tr>
<tr>
<td></td>
<td>Gesture Recognition</td>
<td>9 static and dynamic mid-air gestures</td>
</tr>
<tr>
<td>Face</td>
<td>Face Detection and Recognition</td>
<td>Multiple Face Detection and tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78-point Landmark Detection (facial features)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Emotion Recognition (7 emotions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pulse Estimation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Face Recognition</td>
</tr>
<tr>
<td>Speech</td>
<td>Speech Recognition</td>
<td>Command and Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dictation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Text to Speech</td>
</tr>
<tr>
<td>Environment</td>
<td>Segmentation</td>
<td>Background Removal</td>
</tr>
<tr>
<td></td>
<td>3D Scanning</td>
<td>3D Object / Face / Room Scanning</td>
</tr>
<tr>
<td></td>
<td>Segmentation</td>
<td>2D/3D Object Tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scene Perception</td>
</tr>
</tbody>
</table>

Table 1: Intel Real Sense Capabilities

1.5 Intel Real Sense Requirements

<table>
<thead>
<tr>
<th>Required Hardware</th>
<th>4th Generation or later Intel Core processor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 GB free hard disk space</td>
</tr>
<tr>
<td></td>
<td>Intel RealSense 3D Camera</td>
</tr>
<tr>
<td>Required OS</td>
<td>Microsoft Windows 8.1 OS 64 bit</td>
</tr>
<tr>
<td>Supported Languages</td>
<td>C++, Javascript</td>
</tr>
<tr>
<td></td>
<td>C#(.NET Framework) is required</td>
</tr>
<tr>
<td></td>
<td>Java (JDK 1.7.0_11 or later)</td>
</tr>
<tr>
<td></td>
<td>Processing 2.1.2 or later</td>
</tr>
<tr>
<td>Integrated Development Environment to develop applications</td>
<td>Microsoft Visual Studio 2010-2013 with service pack 1 or newer Unity PRO 4.1.0 or later for Unity game development</td>
</tr>
<tr>
<td>Supported browsers for Javascript development</td>
<td>Microsoft Internet Explorer 10.0.13</td>
</tr>
<tr>
<td></td>
<td>Google Chrome 33.0.1750.146</td>
</tr>
<tr>
<td></td>
<td>Mozilla Firefox 27.0.1</td>
</tr>
</tbody>
</table>

Table 2: Intel Real sense requirements
EXPERIMENT 2

AIM: Introduction to Visual Studio 2013 and write the steps to create a new application.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#).

PROCEDURE

a. **Creation of New File**:
   - File->New->New Project or on the front screen, click on new project option.

   ![Image of Visual Studio interface](image1)

Select Visual C#, Select either of WPF or Windows Form depending on application requirements.

Write name of the project

![Image of Project creation in Visual Studio](image2)
The project window will appear as

b. **Solution Explorer**: Solution Explorer allows you to view items and perform item management tasks in a solution or a project. It also allows you to use the Visual Studio editors to work on files outside the context of a solution or project.

c. **Creation of Class**: Project name (Right Click) -> Add -> New Item. Write name of class and click Add.

d. **Application development**: From the toolbox, select button and place it on the main form. Double click on the form and start coding.
Thus we learnt the concept of Visual Studio 2013 and developed a new project.

OUTPUT

Thus we learnt the concept of Visual Studio 2013 and developed a new project.
EXPERIMENT 3

AIM: To control visual studio application screen through gestures using Intel Real Sense technology.


HARDWARE REQUIREMENTS: Intel Real Sense F200 camera

PROCEDURE:


2. Add the Intel Real Sense libraries:
   a. Go to C:\Program Files(x86)\Intel\RSSDK\bin\x64 and you will find two dll files namely, libpxcclrr.cs.dll and libpxccpp2c.dll.
   b. Copy them and paste it in your project directory. Ex: Here, the projects directory is Documents\Visual Studio 2013\Projects\projectname\projectname.
   c. Project Name (Right Click)->Add->Existing Item-> libpxccpp2c.dll and libpxcclrr.cs.dll.
   d. Right click on libpxccpp2c.dll and in properties window in copy to output directory->copy always.
   e. Solution Explorer->References->Add References->Browse->Select the dll file and click OK.
   f. Project Name(Right Click)->Properties->Build->Platform Target->x64
3. To add a class (Project Name(Right Click)->Add new item-> class.

4. Add **MouseInjection.cs** class and write the following code (For Code- Refer Documentation).

5. Write the code in **MainWindow.cs** (Refer Documentation).

6. For this program

   Toolbox->Button->Place it on screen.

   Add->New Item->WPF-><name>.

   Double click on button and write the code

**CODE:** The code is not mentioned here due to its length.

**OUTPUT**

![Image of two screens with a red arrow pointing from the first screen to the second screen.](image)

The first screen will be controlled via gestures. When the user will press on Click Me by pushing the palm forward, then the second screen will open up.
EXPERIMENT 4

AIM: To demonstrate voice synthesis in Visual Studio Application using Intel Real Sense.


HARDWARE REQUIREMENTS: Intel RealSense F200 camera

PROCEDURE


2. Add the Intel Real Sense libraries:

   a. Go to C:\Program Files(x86)\Intel\RSSDK\bin\x64 and you will find two dll files namely, libpxcclr.cs.dll and libpxccpp2c.dll.

   b. Copy them and paste it in your project directory. Ex: Here, the projects directory is Documents\Visual Studio 2013\Projects\projectname\projectname.

   c. Project Name(Right Click)->Add->Existing Item-> libpxccpp2c.dll and libpxcclr.cs.dll.

   d. Right click on libpxccpp2c.dll and in properties window in copy to output directory->copy always.

   e. Solution Explorer->References->Add References->Browse->Select the dll file and click OK.

   f. Project Name(Right Click)->Properties->Build->Platform Target->x64
3. To add a class (Project Name(Right Click)->Add new item-> class.

4. Add **Voice Out.cs** class and write the following code (For Code- Refer Documentation).

5. Add **VoiceSynthesis.cs** class and write the following code (For Code- Refer Documentation).


7. Write the code in **MainWindow.cs** (Refer Documentation).

8. For this program:
   
   ToolBox->Button->Place it on screen.
   
   Add->New Item->WPF-><name>. Double click on button and write the code.

**CODE:** The code is not mentioned here due to its length.

**OUTPUT**

Whenever the user pushes his palm to click the button, another window opens up which speaks “Hello MPC”.
EXPERIMENT 5

AIM: To display the different texts by showing various Gestures using Intel Real Sense Camera.

SOFTWARE REQUIREMENTS- Intel DCM v1.3, Intel SDK r3, Visual Studio 2010-13 for Windows 8.1

HARDWARE REQUIREMENTS- Intel RealSense 3D Camera (F200)

PROCEDURE:

1. File->New Project->WPF Application-> Name of Project-> Ok
2. Add the Intel Real Sense libraries:
   a. Go to C:\Program Files(x86)\Intel\RSSDK\bin\x64 and you will find two dll files namely, libpxcclr.cs.dll and libpxccpp2c.dll.
   b. Copy them and paste it in your project directory. Ex: Here, the projects directory is Documents\Visual Studio 2013\Projects\gesture\gesture.
   c. Project Name (Right Click)->Add->Existing Item-> libpxccpp2c.dll and libpxcclr.cs.dll.
   d. Right click on libpxccpp2c.dll and in properties window in copy to output directory->copy always.
   e. Solution Explorer->References->Add References->Browse->Select the dll file and click OK.
   f. Project Name(Right Click)->Properties->Build->Platform Target->x86
3. Open MainWindow.xaml. Write the code to design the Application.
4. Open MainWindow.xaml.cs and write the code.

To add a class (Project Name (Right Click)->Add new item-> class.
5. Add ConvertBitmap.cs class and write the code.
7. Build your application.
8. Connect Intel Real Sense Camera to the USB 3.0 port.
9. Start the application.
OUTPUT

Input Screen of Gesture App

Waving Hand Output  Fist Gesture Output

SpreadFinger Output  v_sign Output
Thumb_up Output

On showing different gestures different names will be shown on the text box according to the gesture shown.

Full_pinch Output
EXPERIMENT 6

AIM: To control the LED on Intel Edison by gestures using Intel Real Sense 3D Camera.

HARDWARE REQUIRED: Intel Real Sense camera, Intel Edison, LED


PROCEDURE

1. New Project->Windows Form Application-> Name of Project-> Add.

2. Add the Intel Real Sense libraries:
   a. Go to C:\Program Files(x86)\Intel\RSSDK\bin\x64 and you will find two dll files namely, libpxcclr.cs.dll and libpxccpp2c.dll.
   b. Copy them and paste it in your project directory. Ex: Here, the projects directory is Documents\Visual Studio 2013\Projects\ledsensor\ledsense.
   c. Project Name(Right Click)->Add->Existing Item-> libpxccpp2c.dll and libpxcclr.cs.dll.
   d. Right click on libpxccpp2c.dll and in properties window in copy to output directory->copy always.
   e. Solution Explorer->References->Add References->Browse->Select the dll file and click OK.
   f. Project Name(Right Click)->Properties->Build->Platform Target->x64

3. Open Form.Designer.cs. Write the code to design the Form.

4. Open Program.cs and write the following code.

To add a class (Project Name (Right Click)->Add new item-> class.

5. Add timer.cs class and write the following code.

6. Add handrecognition.cs class and write the following code.

7. Open Form1.cs and write the following code.

8. Build your application.

9. Connect Intel Edison(port away from the edge) to the Laptop. Open Device Manager and see the Virtual COM Number.

10. Open Arduino- Intel and write the code.

11. Connect a LED on Edison according to the code.

13. Connect Intel Real Sense Camera to the USB 3.0 port.
14. Start the application.

OUTPUT

App Output using SpreadFinger Gesture   Edison Output: Blinking Led
EXPERIMENT 7

AIM: To get the current temperature by the LM35 temperature sensor interfaced on Intel Edison and use voice synthesis capability of Intel Real Sense to speak the current temperature.

HARDWARE REQUIRED: Intel Real Sense camera, Intel Edison, LM35


PROCEDURE

1. New Project->Windows Form Application-> Name of Project-> Add.

2. Add the Intel Real Sense libraries:
   a. Go to C:\Program Files(x86)\Intel\RSSDK\bin\x64 and you will find two dll files namely, libpxccllr.cs.dll and libpxccpp2c.dll.
   b. Copy them and paste it in your project directory. Ex: Here, the projects directory is Documents\Visual Studio 2013\Projects\ledsense\ledsense.
   c. Project Name(Right Click)->Add->Existing Item-> libpxccpp2c.dll and libpxccllr.cs.dll.
   d. Right click on libpxccpp2c.dll and in properties window in copy to output directory->copy always.
   e. Solution Explorer->References->Add References->Browse->Select the dll file and click OK.
   f. Project Name(Right Click)->Properties->Build->Platform Target->x64

3. Open Program.cs and write the code.

To add a class (Project Name(Right Click)->Add new item-> class.

4. Add VoiceOut.cs class and write the code.

5. Add VoiceSynthesis.cs class and write the code.


7. Open Form1.cs and write the code.

8. Build your application.

9. Connect Intel Edison(port away from the edge) to the Laptop. Open Device Manager and see the Virtual COM Number.

10. Open Arduino- Intel and write the code
11. Connect a LM35 temperature Sensor on Edison according to the code.

13. Connect Intel Real Sense Camera to the USB 3.0 port.

14. Start the application.

**OUTPUT**

![Intel Real Sense Application that would synthesis voice.](image)

![LM35 Temperature Sensor](image)