Intel® RealSense™ SDK Code Samples

Samples Implemented By *Felipe Pedroso* and *João Pedro Nardari*

Abstract

This set of code samples was created to be used during the Brazilian Intel RealSense Hands-on Labs to make it easier for the participants to understand how to use the Intel® RealSense™ SDK. The 12 samples use the C# SDK wrapper and provide simple, console-based apps that print the information available from the RealSense modalities including face and hand tracking and speech recognition. Also, there are 2 WPF apps showing how to display the camera streams and how to achieve background subtraction.

Introduction

As part of preparation for the Hands-On Labs Brazil, we created 12 code samples with instructions to show how to leverage Intel RealSense voice and camera capabilities with simple examples. The code is commented (in English) and can be freely shared with the worldwide developers' community. The samples were implemented using C# and are basically simple console applications that show how to use the RealSense SDK functionalities. The code has been tested with the Intel RealSense SDK R2 (RSSDK).

We hope you enjoy our contribution and if you have any questions or need help, please use the comments section below.

Pre-Requisites to Run the Samples

- Intel® RealSense™ SDK R2
- Intel® RealSense™ 3D camera (F200)
- Microsoft* Visual Studio* 2010 or later
- 4th generation Intel® Core™ processor or later
- Windows* 8.1 or later, 64 bit with August update

Important Intel RealSense Documentation Links

- SDK Architecture
- Programming Guide
- Session Class
- SenseManager Class

Available Samples

Camera Calibration Library

Camera Calibration is a library project that receives a device and a modality and makes the proper calibration to improve the quality of the camera recognition for that specific mode, for example hand tracking.

Reference Links: Depth Settings for different modalities

Device

The Device sample makes device enumeration possible. Select a device and get the available streams and set device configurations. Note: this sample uses the Camera Calibration Library to configure the devices.

Reference Links:

- I/O Devices Operations
- Enumerating Devices
- Enumerating Streams
- Enumerating Streams Configurations

Emotion

The Emotion sample lists emotions using SenseManager with a procedural implementation. This sample finds all the emotion data and prints each one along with its intensity value.

Reference Links:

- Emotion Detection
- Detection via SenseManager
- Intensity and Evidence
- PXCMEmotion

Emotion with Callback

The Emotion with Callback sample has the same functionality as the Emotion sample, but with a different implementation. It shows how to use handlers in the RSSDK to get module data. It uses the Emotion module, but can be implementated with other modules.

Reference Links: Emotion Detection with Callback

Face

Face is a sample that implements some of the various functionalities of the Face module. It uses the PXCMFaceData object and processes information separately as listed in the functionalities below:

Reference Links:

- Face Tracking via SenseManager
- Configuration and Data
- Detection - Prints X, Y, width, and height from a detected face; Location Documentation
- Expressions - Prints all detected expressions (one-by-one) with their intensity from a detected face; Expressions Documentation
- FaceExpression Enumerator
- Landmarks - Prints all (max 78) landmarks from a detected face; Landmark Documentation
- Pose - Prints X, Y, Z Euler Angles from a detected head. Pose Documentation

Face Recognition

The Face Recognition sample detects a face and checks if the user is already registered. When the program detects a face that is not registered, the user can press the space bar to register their face in the database (in memory).. After registration, the sample prints the unique identifier of the recognized face.

Reference Links: Face Recognition Documentation

Hands

The Hands sample tracks hands, fingers and gestures. The sample prints how many hands are detected and their positions (image and world), body sides, joints and detected gestures.

Reference Links:

- Gesture Recognition Data
- Gesture Interaction Guide
- Hand Tracking via SenseManager
- Hand Tracking Data
- Joint Type
- Joint Data

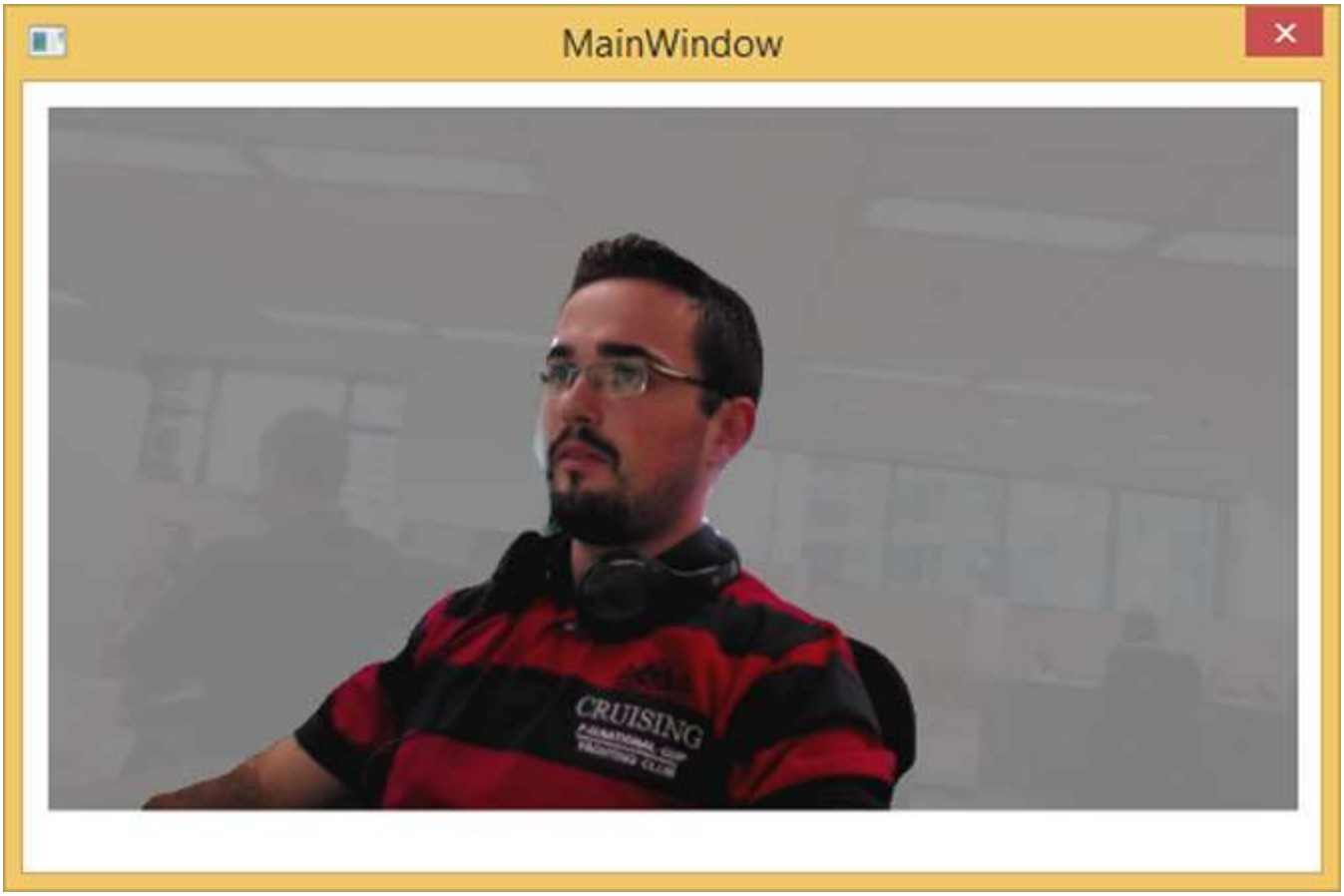
Object Tracking

The Object Tracking sample detects a 2D object using Intel RealSense™ SDK. It uses JPEG/PNG markers, and tracks X, Y, and Z positions as the object is tracked by the camera. Note: This sample requires that the camera is calibrated with a specific tag.

Reference Links:

- Object Tracking Documentation
- Object Tracking via SenseManager
- The Metaio Toolbox Calibration(Instructions)

Segmentation



The Segmentation sample uses the WPF structure to display the camera stream on a WPF form and uses the Segmentation feature to remove the image background.

Reference Links:

- User Segmentation
- Accessing Image and Audio Data
- ImageData to Bitmap

Speech Recognition

The Speech Recognition sample shows how to use both Speech Recognition modes: DICTATION or COMMAND. In dictation mode, it recognizes all words that users are saying

and prints them on the screen. In Command mode, the program sets a standard dictionary and when the user says one of added commands, it prints it on the screen.

Reference Links:

- Speech Recognition Documentation
- Command Control and Dictation
- Handling Recognition Events
- RecognitionData Object

Speech Synthesis

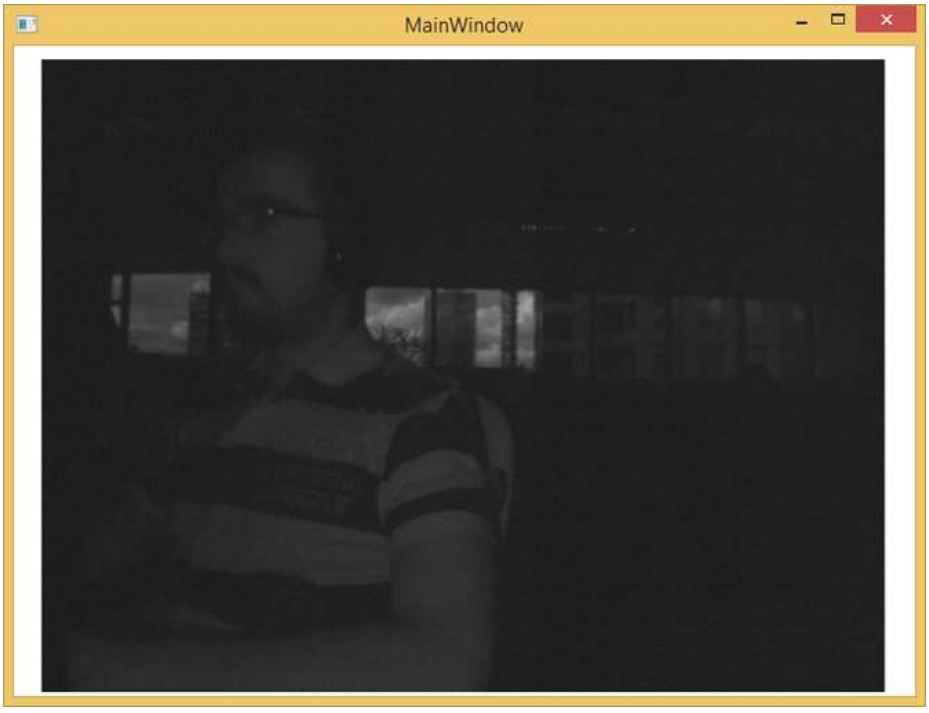
The Speech Synthesis sample is an implementation of the text-to-speech capability of the Intel RealSense SDK. When a sentence is made available in the profile, it converts the sentence to audio and plays it.

Reference Links:

- Speech Synthesis Documentation
- BuildSentence
- Acquiring Audio File

Streams

IR stream view (notice the effect from the outside lighting in the background.)



The Streams samples uses the RSSDK to display the Creative Camera streams (Color, Depth and Infrared) into a WPF form. The sample selects a stream by its type and shows a window with the selected camera stream, updating the image frame-by-frame in the selected FPS configuration.

Reference Links:

- Enabling Streams
- Capturing Individual Color or Depth Stream
- Accessing Image and Audio Data
- ImageData to Bitmap

Download the Samples

To experiment with these samples and learn more about how use the Intel RealSense SDK, please download the code from https://software.intel.com/sites/default/files/managed/48/85/RealSense_BrazilLab_CodeSamples_1.3.zip

About Intel® RealSense™ Technology

To get started and learn more about the Intel RealSense SDK for Windows, go to https://software.intel.com/en-us/realsense/intel-realsense-sdk-for-windows.

About the Authors

João is a Software Analyst Intern in the Developers Relations Division Brazil. He is studying Information Systems at University of São Paulo and is a Software Developer working mainly with mobile platforms, web applications and RealSense.

Felipe is an Intel RealSense Technical Evangelist in the Developers Relations Division Brazil. He studied Computer Engineering and worked with different technologies, platforms and programming languages during his career. His main interests are game development, mobile platforms and HTML5.