Intel(R) Perceptual Computing SDK Demo Application

Remote Control

Table of Contents

1 Legal Disclaimer

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

Unless otherwise agreed in writing by Intel, the Intel products are not designed nor intended for any application in which the failure of the Intel product could create a situation where personal injury or death may occur.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "Reserved" or "Undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's web site http://www.intel.com.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Intel, the Intel logo, and Intel Core are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

1.1 Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

2 Introduction

The Intel® Perceptual Computing SDK is a library of pattern detection and recognition algorithm implementations exposed through standardized interfaces. The library aims at lowering barriers to using these algorithms and shifting the application developers' focus from coding the algorithm details to innovating on the usage of these algorithms for next generation human computer experience.

This document describes the `RemoteControl` demo application.

Document Convention

The SDK document uses the Calibri typeface for normal prose.

With the exception of section headings, captions and the table of contents, all code-related items appear in the Courier New typeface (`pxcStatus`).

Hyperlinks appear underlined in blue, such as pxcStatus.

This is a note that provides additional information to aid your understanding of the procedure or concept.

This is a tip that provides alternate methods or shortcuts.

This is a result statement which indicates what you can expect to see or happen after performing a step.

3 Remote Control

RemoteControl is a sample application that converts perceptual computing events (finger tracking, gesture recognition, face detection, and voice recognition events) to mouse and keyboard events to enable applications that are not perceptual computing aware. With RemoteControl, the user can use those applications as if they are "native" perceptual computing applications.

The event mapping (or binding) can be different for each application. The user can create bindings for a single application or multiple applications. RemoteControl will figure out which window is in focus and send corresponding events (specific to the application) only.

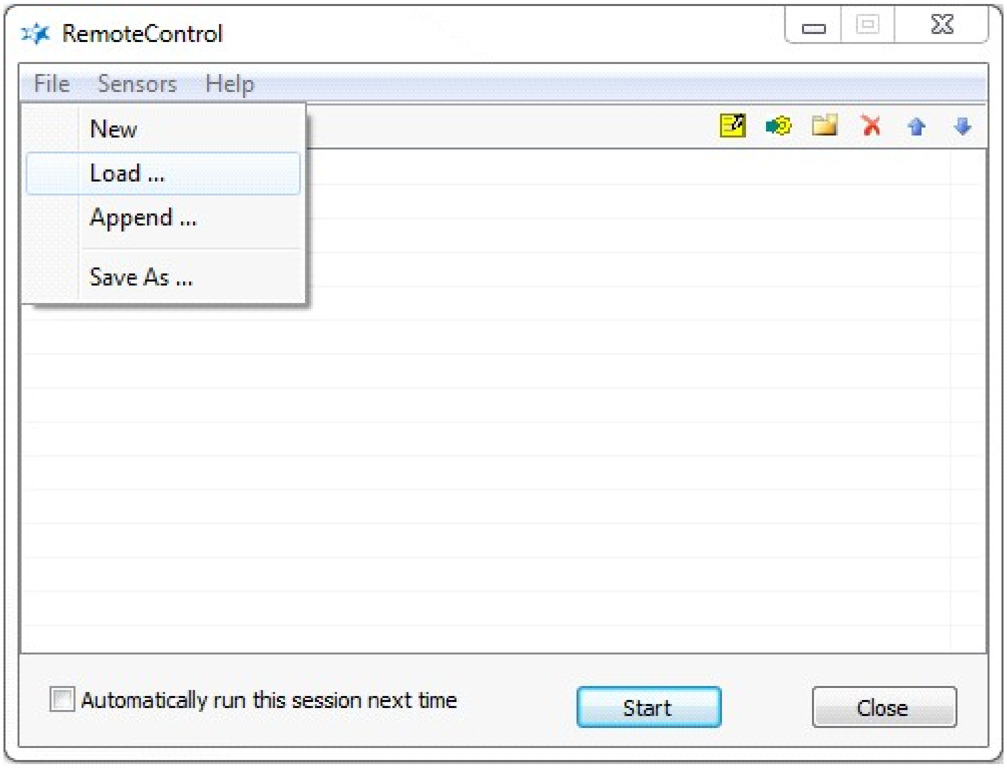
The application directory structure is as follows:

Directory/File	**Description**
AUXlib sensors src	The sample source code.
doc	The directory that contains this document.
RemoteControl.exe	The prebuilt application executable.
calculator_voice.xml notepad.xml notepad_voice.xml RC_presentation.xml	Predefined control scripts for applications such as calculator, notepad, and powerpoint.

3.1 Operation Instructions

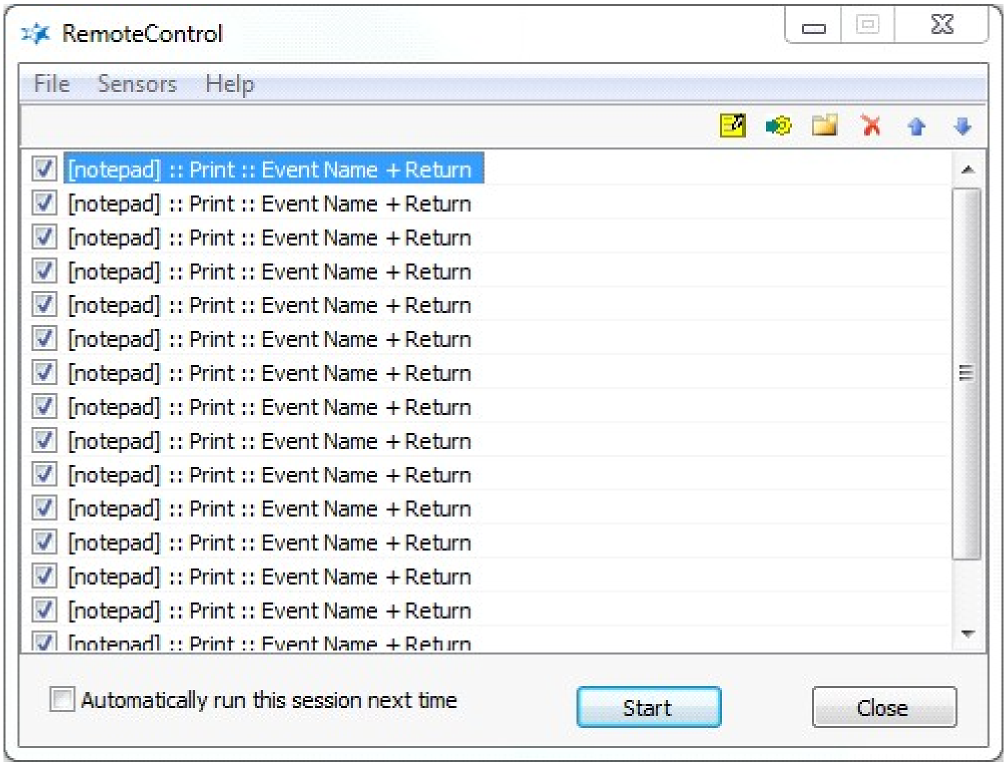
Run the prebuilt executable RemoteControl.exe to start the sample application, which shows the configuration window, as illustrated in Figure 1.

The user may load a pre-defined binding configuration, through File->Load, or start a new binding, through File->New. When the user loads a pre-defined binding configuration, as illustrated in Figure 2, the sample will list all bindings. The user may enable or disable certain binding by right clicking on the items.



□ Automatically run this session next time

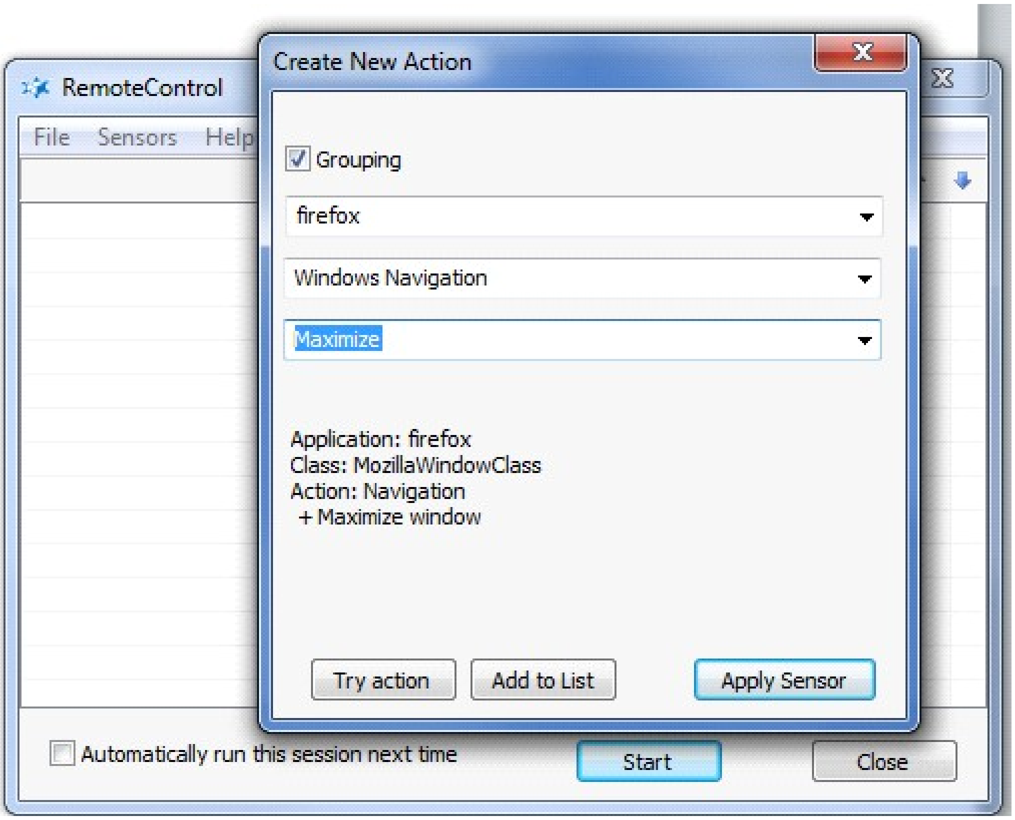
Figure 1: RemoteControl Launch Screen



☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return
☑ [notepad] :: Print :: Event Name + Return

□ Automatically run this session next time

Figure 2: NotePad Predefined Mappings



☑ Grouping

firefox

Windows Navigation

Maximize

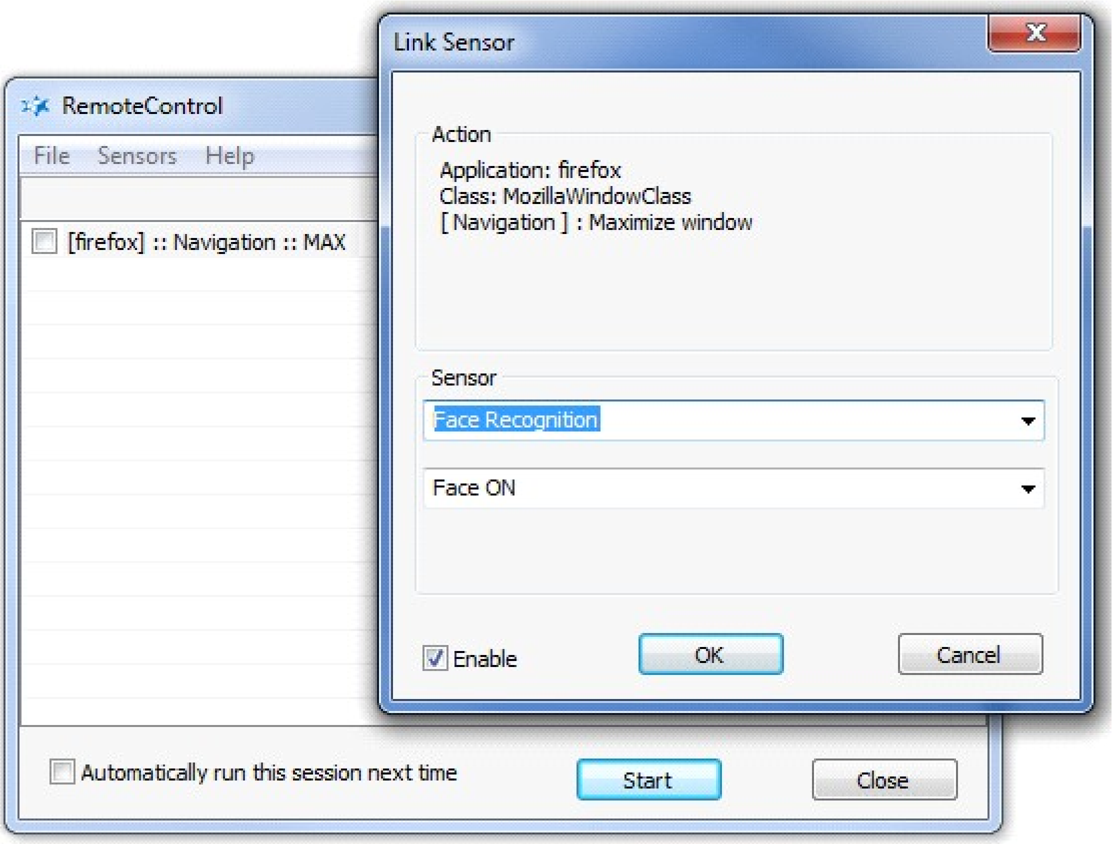
Application: firefox
Class: MozillaWindowClass
Action: Navigation
+Maximize window

□ Automatically run this session next time

Figure 3: The Binding Dialogue

The user can double click on any empty item to add any binding or create a new binding. Double clicking will bring up the binding dialogue (See Figure 3), which defines an application action and its association to any available event. In Figure 3, the action is the maximization operation of the Firefox* browser, and the associated event is any face detection event (See Figure 4.)

The user may use `File->Save As` to save any configuration at any time.



□ [firefox] :: Navigation :: MAX

□ Automatically run this session next time

Action

Application: firefox
Class: MozillaWindowClass
[Navigation] : Maximize window

Sensor

Face Recognition

Face ON

☑ Enable

Figure 4: Map Perceptual Events to Application Actions

After configuration, the user may click the `Start` button to start the service. The `RemoteControl` main window will be minimized. The user can stop the service by clicking the `Close` button at any time.

The user will see a few miniature helper windows on the desktop, as illustrated in Figure 5, with each helper window representing an input modality. The helper windows are there to provide visual feedback of what are recognized and what actions are taken. The user can move their locations and/or close them by right clicking them and selecting the right options.

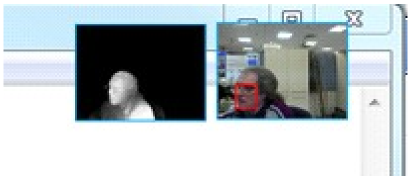
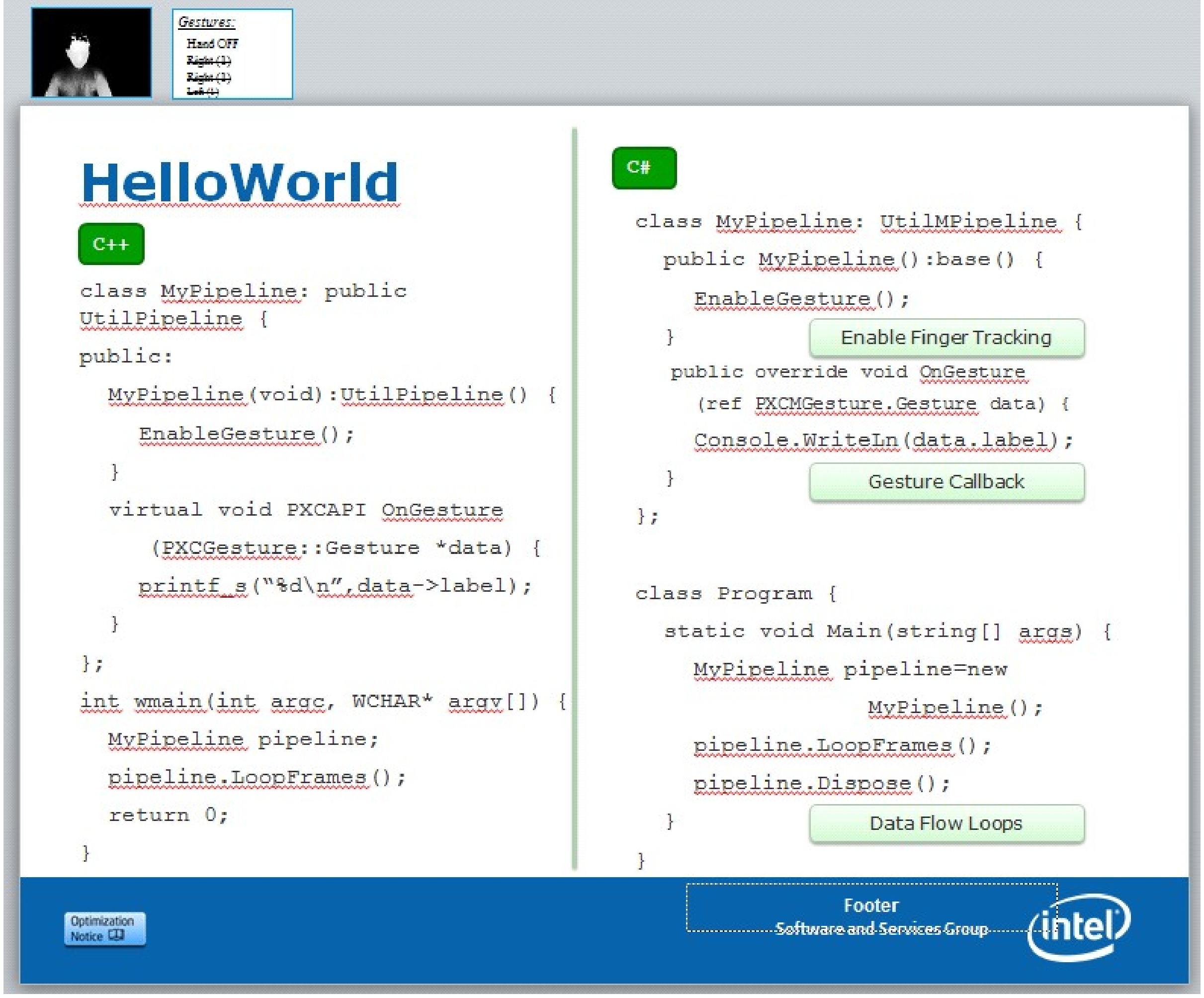


Figure 5: The Video Helper Window

In Figure 6, the user swipes left to turn the presentation forward and swipes right to turn the presentation backward. The helper windows on top show the recognition events and the actions taken.



HelloWorld

C++

```
class MyPipeline: public
UtilPipeline {
public:
  MyPipeline(void):UtilPipeline() {
    EnableGesture();
  }
  virtual void PXCAPI OnGesture
     (PXCGesture::Gesture *data) {
    printf_s("%d\n",data->label);
  }
};
int wmain(int argc, WCHAR* argv[]) {
  MyPipeline pipeline;
  pipeline.LoopFrames();
  return 0;
}
```

C#

```
class MyPipeline: UtilMPipeline {
  public MyPipeline():base() {
    EnableGesture();
  }
  public override void OnGesture
    (ref PXCMGesture.Gesture data) {
    Console.WriteLn(data.label);
  }
};

class Program {
  static void Main(string[] args) {
    MyPipeline pipeline=new
                MyPipeline();
    pipeline.LoopFrames();
    pipeline.Dispose();
  }
}
```

Figure 6: Control Presentation Flow by Swiping

3.2 Build Instructions

The application comes with source code by clicking on the solution file `RemoteControl_vs2010.sln`. The solution file is based on Microsoft* Visual Studio* 2010.

3.3 Design Points

This section describes the application design that is related to Intel® Perceptual Computing SDK and is not meant to be a code workthrough of all application designs.

The modality detection code is located under the `sensors` directory. Take gesture recognition for example, the modality detection code is in `sensors/sensor_gesture_CI.cpp`, which uses the `UtilPipeline` class to engage gesture recognition. When a gesture is recognized, the

`OnGesture` callback is invoked. The code subsequently invokes the associated keyboard or mouse events that are associated with the current window in focus.