1st Kinect Workshop
Consumer Depth Sensors
Natural User Interaction (NUI)

Natural Interaction Research Team (NIRTeam)
Efthimios Syntychakis IDNum: 2243
Eleftherios Manousakis IDNum: 2185
Workshop Objective

• Introduction to Natural User Interface

• Introduction to Technology RGB & Depth of NUI devices (e.g. Kinect)

• Demo Sample Application
Available Depth Sensors

- **Microsoft Kinect Sensor**
  Microsoft Kinect was originally designed for the Xbox 360 console, but soon output decoding algorithms were produced, for computer usage, from developers.

- **Asus Xtion Pro LIVE Sensor**
  Designed by PrimeSense in partnership with Asus, the Xtion aimed at developers.
Microsoft Kinect Sensors

- IR light
- RGB Camera
- Depth Sensor
- Motorized Tilt
- Multi-Array Mic

Additional features:
- Accelerometer
- Depth Sensor Range 0.8 to 3.5 meters
- 4 Microphones
- Viewing angle 43° vertical by 57° horizontal
- External power is needed
Asus Xtion Pro LIVE Sensors

- Depth Sensor Range 0.8 to 3.5 meters
- 2 Microphones
- Viewing angle 45° vertical, 58° horizontal, 70° diagonal
- External power is not needed

Additional features
Explanation of a Depth Image

- The IR (infrared) light produces a point array as illustrated below.
- The Depth Sensor which is a monochrome CMOS (complimentary metal-oxide semiconductor) sensor uses each point’s position to produce a depth image.

For extra information you can see: http://www.primesense.com/en/...
Kinect and Xtion provide three outputs:

1. **Image**,
2. **Depth**,
3. **Audio**.

We use these outputs through the **NUI library** and according to our application.

Five known NUI libraries are:

1. **OpenNI/NITE**
2. **Libfreenect**
3. **CL NUI**
4. **Microsoft Kinect SDK**
5. **Evoluce SDK**
Depth Sensor Available APIs (Application Programming Interfaces)

**OpenNI/NITE**
Supports both Microsoft Kinect and Xtion Pro LIVE. It is also the API which Asus distributes with Xtion Pro LIVE.

**Libfreenect**
The first Kinect Hack API ever made, before Microsoft published Kinect SDK.

**CL NUI**
Widely known in the community of robotic systems.

**Microsoft Kinect SDK (Beta)**
Recently produced by Microsoft, creating a big research community.

**Evoluce SDK**
Kinect API made by Evoluce AG, for use in Windows 7.
Depth Sensor Available APIs

### OpenNI/NITE

**Native programming languages:**
- C/C++, C# (Windows), C/C++ (Linux)

**Available Wrappers:**
- Unity, Java, VB.NET, Python

**Depth camera Drivers:** Yes
**RGB camera Drivers:** Yes
**Audio Drivers:** No
**Accelerometer:** No
**Skeleton Tracking:** Yes
**Calibration Posture:** Needed
**Skeleton Joints:** 15
**Hand Tracking Framework:** Yes
**OS support:** Windows, Linux, Mac OSX
**Game Engine Support:**
- Unity 3D (C#)
- OGRE (C++)

License includes commercial use

---

- Relatively difficult to install.
- Several methods ready to use for Gesture Recognition.
- Very popular, with many applications in various fields.

*Calibrated user skeleton*
Depth Sensor Available APIs (Kinect Only)

**Libfreenect**

- **Native programming languages:** Python, C
- **Available Wrappers:** Actionscript, C++, C#, Java, Javascript, Lisp
- **Depth camera Drivers:** Yes
- **RGB camera Drivers:** Yes
- **Audio Drivers:** No
- **Accelerometer:** No
- **Skeleton Tracking:** No
- **Calibration Posture:** -
- **Skeleton Joints:** -
- **Hand Tracking Framework:** No
- **OS support:** Windows, Linux, Mac OSX
- **Game Engine Support:** -
- **License includes commercial use**

- Very difficult to install for a beginner.
- Several applications.
Depth Sensor Available APIs (Kinect Only)

**CL NUI**

Native programming languages: C/C++, C#

Available Wrappers: None

Depth camera Drivers: Yes
RGB camera Drivers: Yes
Audio Drivers: Yes
Accelerometer: Yes
Skeleton Tracking: No
Calibration Posture: -
Skeleton Joints: -
Hand Tracking Framework: No
OS support: Windows
Game Engine Support: -

License includes commercial use

- CL NUI Accelerometer test
  - Easy to install.
  - Fairly widespread.
  - Poor High level API.
## Microsoft Kinect SDK (Beta)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Native programming languages:</strong></td>
<td>C++, C#, VB.NET</td>
</tr>
<tr>
<td><strong>Available Wrappers:</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Depth camera Drivers:</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>RGB camera Drivers:</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Audio Drivers:</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Accelerometer:</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Skeleton Tracking:</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Calibration Posture:</strong></td>
<td>Not Needed</td>
</tr>
<tr>
<td><strong>Skeleton Joints:</strong></td>
<td>20</td>
</tr>
<tr>
<td><strong>Hand Tracking Framework:</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>OS support:</strong></td>
<td>Windows 7</td>
</tr>
<tr>
<td><strong>Game Engine Support:</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

**License does not include commercial use**

### Skeletal Viewer sample on Microsoft SDK

- Very easy to install.
- Many examples and code to familiarize a novice.
- Documentation with code explanation through samples.
- Active support from the developer community.
# Depth Sensor Available APIs (Kinect Only)

## Evolunce SDK (High level SDK)

<table>
<thead>
<tr>
<th>Native programming languages:</th>
<th>C/C++, C#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Wrappers:</td>
<td>None</td>
</tr>
<tr>
<td>Depth camera Drivers:</td>
<td>Yes</td>
</tr>
<tr>
<td>RGB camera Drivers:</td>
<td>Yes</td>
</tr>
<tr>
<td>Audio Drivers:</td>
<td>No</td>
</tr>
<tr>
<td>Accelerometer:</td>
<td>No</td>
</tr>
<tr>
<td>Skeleton Tracking:</td>
<td>Yes</td>
</tr>
<tr>
<td>Calibration Posture:</td>
<td>Needed</td>
</tr>
<tr>
<td>Skeleton Joints:</td>
<td>15</td>
</tr>
<tr>
<td>Hand Tracking Framework:</td>
<td>Yes</td>
</tr>
<tr>
<td>OS support:</td>
<td>Windows 7</td>
</tr>
<tr>
<td>Game Engine Support:</td>
<td>Unity 3D (C#)</td>
</tr>
<tr>
<td></td>
<td>OGRE (C++)</td>
</tr>
</tbody>
</table>

*License does not include commercial use*

- Easy to install.
- Ready to use methods for Gesture Recognition.
- Based on OpenNI / NITE of Primesense
## Depth Sensor Available APIs

### API versus features supported

<table>
<thead>
<tr>
<th></th>
<th>Depth cam</th>
<th>RGB cam</th>
<th>Audio</th>
<th>Accelerometer</th>
<th>Skeleton Tracking</th>
<th>Calibration Posture</th>
<th>Skeleton Joints</th>
<th>Hand Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenNi/NITE</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>Needed</td>
<td>15</td>
<td>✓</td>
</tr>
<tr>
<td>Libfreenect</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CL NUI</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS Kinect SDK</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>Not needed</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Evoluce SDK</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>Needed</td>
<td>15</td>
<td>✓</td>
</tr>
</tbody>
</table>

### API versus Languages supported

<table>
<thead>
<tr>
<th></th>
<th>C/C++</th>
<th>C#</th>
<th>Java</th>
<th>Python</th>
<th>Lisp</th>
<th>VB.NET</th>
<th>Actionscript</th>
<th>Javascript</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenNi/NITE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Libfreenect</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CL NUI</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS Kinect SDK</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Evoluce SDK</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Characteristics versus Application Type

<table>
<thead>
<tr>
<th>Application Type</th>
<th>Example Application</th>
<th>OpenNI/NITE</th>
<th>Libfreenect</th>
<th>CL NUI</th>
<th>MS Kinect SDK</th>
<th>Evoluce SDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand tracking</td>
<td>Touch-free user interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Body Tracking</td>
<td>Pedestrian Detection</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Tracking</td>
<td>Real Time object tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech recognition</td>
<td>Home automation</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depth Image</td>
<td>PointCloud 3D environment imaging</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sitting position</td>
<td>Control Cursor with hand (while sitting in front of the PC)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game engine support</td>
<td>3D Animated Skeleton for Model Movement</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple sensor support</td>
<td>3D Modeling</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Hand gesture recognition</td>
<td>Deaf language Recognizer</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full body gesture recognition</td>
<td>Motor Disabilities Rehabilitation program</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any-OS compatible application</td>
<td>NUI Game</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commercial Application</td>
<td>Virtual dressing room</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joints rotation</td>
<td>Flight simulator</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Installation Guide for OpenNI/NITE

**Step 1** – Uninstall all Kinect or Xtion drivers our system has.

**Step 2** – Download and install updated Stable or Unstable versions of:
- OpenNI Binaries
- OpenNI Compliant Middleware Binaries
- OpenNI Compliant Hardware Binaries

**Step 3** – Bring Device Manager up to be sure that the above steps were successful. If they were, we have to see something that resembles the picture to your right.

**Step 4** – Bring up `C:\Program Files\OpenNI\Samples\Bin\Release`
Here are the samples of OpenNI and `C:\Program Files\Prime Sense\NITE\Samples\Bin\Release`
the samples of NITE.

**Step 5** – If all the above works out then we successfully installed OpenNI/NITE and we are able to locate the libraries:
*OpenNI: C:\Program Files\OpenNI\Bin*
*NITE: C:\Program Files\Prime Sense\NITE\Bin*
For the libfreenect installation there is a specific guide for each different software in the OpenKinect site:

http://openkinect.org/wiki/Getting_Started

Libfreenect is the hardest to install, on the other hand with the huge developer Community it’s users are provided many opportunities and source code for software development.
In order to install CL NUI we only have to navigate to the page http://codelaboratories.com/nui and download the CL NUI Platform Installer.
A comprehensive installation guide, and also a proposed guide for first time usage of kinect in C#, is Kinect for Windows SDK Quickstarts.

http://channel9.msdn.com/Series/KinectSDKQuickstarts

Following the link you can watch video tutorials, PowerPoint presentations, but also code for the first uses of kinect with C#.
To install Evoluce SDK we only have to navigate to the page
and download the SDK.

Evoluce provides some code examples (samples) with which we can become familiar with the API.
According to the initialization of the Kinect device we choose what information we need. We can choose one or more of the following:

- **UseDepth**
  Turns on the IR light and Depth Sensor and gives us depth information.

- **UseDepthAndPlayerIndex**
  Turns on the IR light and Depth Sensor and gives us depth information and also to whom each pixel belongs.

- **UseColor**
  Turns on the RGB Camera and gives us color image.

- **UseSkeletalTracking**
  Turns on the IR light and Depth Sensor and gives us the users’ skeleton coordinates.

- **Kinect Audio**
  In order to initialize the Microphone Array, a series of commands are needed, which are described in a previous tutorial.
UseDepth and UseDepthAndPlayerIndex

From the depth stream, we get 2 byte of information for each pixel contained in the Cartesian distance in millimeters, from the camera to the nearest object in the x, y coordinates of the sensor’s system axes. We have 2 options:

<table>
<thead>
<tr>
<th>Depth</th>
<th>Depth and Player Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>• When we choose only depth, the 12 less significant bits (0-11) of each pixel which contain the depth data. The other 4 are not used.</td>
<td>• When we choose both depth and player index, the 3 less significant bits (bits 0–2) from each pixel contain the player index and the rest the depth data.</td>
</tr>
</tbody>
</table>
Through the Image stream we receive a normal picture like a webcam. Below are the 2 encoding options of the image:

- **ImageType.Color**
  - 32-bit, linear X8R8G8B8-formatted color bitmaps, in sRGB color space.

- **ImageType.ColorYuv**
  - 16-bit, gamma-corrected linear UYVY-formatted color bitmaps.
MS Kinect SDK gives us 20 joints for which we have the following information:

- **ID**: Name of specific joint
- **Position**: Vector4 variable where:
  - X – Joint’s position in x axes
  - Y – Joint’s position in y axes
  - Z – Joint’s position in z axes
  - W – The Tracking Confidence Kinect has for a specific Joint (takes values 0 - 1.0)
- **Tracking State**: At what state is the detection of the Joint, which has 3 possible outcomes:
  - Tracked, PositionOnly, NotTracked
Kinect Audio

Capabilities of Audio API of Kinect SDK:
- Raw Audio Capture (C#, C++)
- Echo Cancellation (C++)
- Record Audio (C#)
- Speech Recognition (C#)
- Detect Sound Source and Beam (C#, C++)

Example application
Detect Sound Source and Beam
## Kinect SDK Requirements

### Hardware:
- 32 bit (x86) or 64 bit (x64) processor
- Dual-core 2.66-GHz or faster processor
- Dedicated USB 2.0 bus
- 2 GB RAM
- A retail [Kinect for Xbox 360® sensor](https://www.xbox.com/en-US/accessories/microsoft-kine) which includes special USB/power cabling

### Software:
- [.NET Framework 4.0](https://dotnet.microsoft.com/download)
Indicative Research Areas with use of NUI devices

The NUI devices are new devices that use standard technology of materials. Looking back at bibliography we find an increasing trend of implementation of classical and non-classical systems using NUI devices. These outputs derive from different research areas such as:

- **Augmented Reality**
- **Ambient Intelligent Environments**
- **Computational Vision and Robotics**
- **Disabilities**
- **Gesture Recognition**
- **Data Visualization**
NUI exemplar applications: **HoloDesk**

### Application details

**Research Area:** Augmented Reality  
**Project Name:** HoloDesk  
**Developed by:** Microsoft Research Labs  
NUI exemplar applications: *Interactive Light Display*

**Application details**

- **Research Area:** Ambient Intelligent Environments
- **Project Name:** Interactive Light Display
- **Developed by:** Jared St. Jean
Application details

Research Area: Computational Vision and Robotics
Project Name: StarMac
Developed by: Hybrid Systems Lab
Reference: http://hybrid.eecs.berkeley.edu/
NUI exemplar applications: NAVI

Application details
Research Area: Disabilities
Project Name: NAVI (Navigational Aids for the Visually Impaired)
Developed by: University Of Konstanz, Germany
Reference: http://hci.uni-konstanz.de/blog/2011/03/15/navi/?lang=en
NUI exemplar applications: *Tedcas*

**Application details**

- **Research Area:** Gesture Recognition in Hospitals
- **Project Name:** Tedcas
- **Developed by:** Tedesys
NUI exemplar applications: **SimplyCube & Kinect**

**Application details**

- **Research Area:** Motion Capture
- **Project Name:** SimplyCube & Kinect
- **Developed by:** SimplySim
- **Reference:** [http://www.simplysim.net/blog/2011/01/11/](http://www.simplysim.net/blog/2011/01/11/)
NUI exemplar applications: *Interactive Hopscotch*

**Application details**

- **Research Area:** Data Visualization
- **Project Name:** Interactive Hopscotch
- **Developed by:** MediaLab Helsinki
- **Reference:** [http://kinect.dashhacks.com/kinect/]
Natural Interaction Research Team
Application(s)

Application details
Research Area: Head Pose Estimation
Project Name: Window View
Developed by: NIRTeam
Application details

Research Area: Gesture Recognition
Project Name: Puzzle Image
Developed by: Efthimios Syntychakis (NIRTeam)
Natural Interaction Research Team
Application(s)

Application details
Research Area: Gesture Recognition
Project Name: Presentation Helper
Developed by: Efthimios Syntychakis (NIRTeam)
• http://kinectforwindows.org/

• http://openkinect.org/wiki/Main_Page

• http://openni.org/

• http://codelaboratories.com/home/

• http://channel9.msdn.com/Niners/Code4Fun

• http://channel9.msdn.com/coding4fun/kinect/Kinect-Audio-Positioning

• http://www.evoluce.com/
Thank you for your attention.

We welcome your questions, suggestions, comments!