Using Kinect with Robotics Developer Studio

Trevor Taylor, PhD
Senior Program Manager
Microsoft Robotics Group

Agenda

- Overview of Robotics Developer Studio
- Kinect Overview
- Kinect Programming
- Demos
- Summary

Microsoft® Robotics Developer Studio 4 Beta

Runtime	Authoring	Services & Samples	Reference Platform
• Concurrency and Coordination Runtime (CCR) • Decentralized Software Services (DSS) Framework	• Visual Studio (C#) • Visual Simulation Environment • Visual Programming Language	• Samples & Tutorials • Robot services • Robot models • Technology services	• Design Specification • Example Implementation • Robot services • Simulation Entity

A development platform for the robotics community (academia, hobbyist, research, and commercial), supporting a wide variety of hardware and application scenarios

V1 Dec 2006 — Over 550k downloads — Over 20 partners — V4 Dec 2011

RDS Application Stack

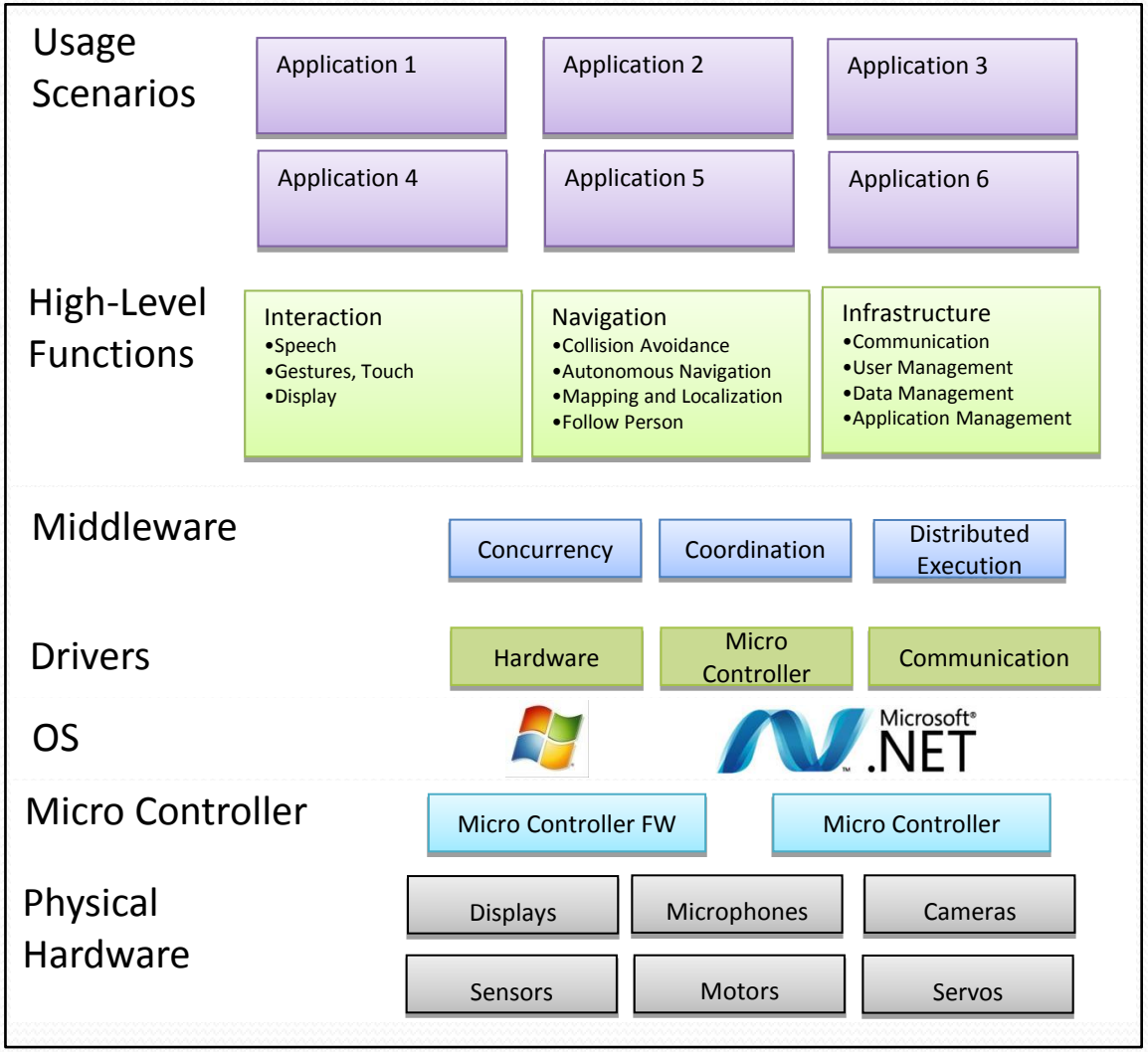
New Opportunities for Developers

Need to Expand and Simplify

Development Tools:

Microsoft® Robotics Developer Studio 4 Beta

Visual Studio 2010

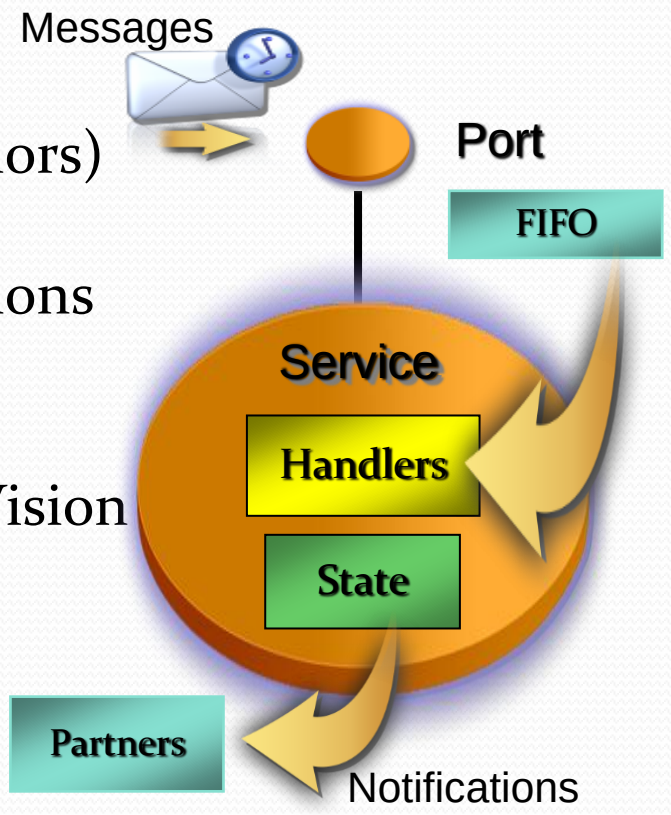


Key RDS Runtime Features

- Concurrency and Coordination Runtime (CCR)
  - Simplifies writing and managing asynchronous processes
  - Avoids the need to understand manual threading, semaphores, etc.
- Decentralized system services (DSS)
  - Service Oriented Architecture (SOA)
  - Makes State observable and easily accessible (RESTful)
  - Provides for reusability and failure tolerance
  - Supports remote/distributed execution
  - Makes the programming model scalable

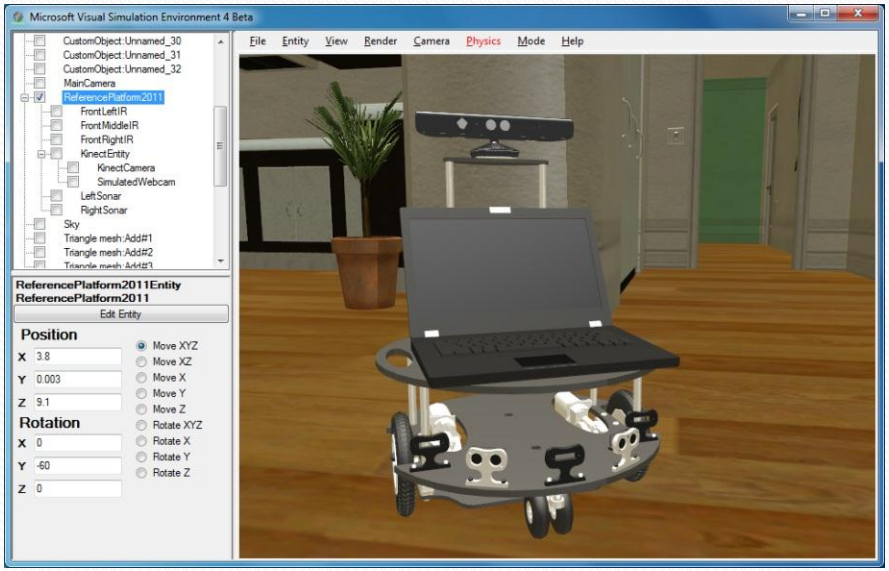
RDS Services – Basic Building Blocks

- Services:
  - Provide abstractions for hardware and software
  - Are inherently distributed and asynchronous
  - Have structured internal state
  - Interact using messages over ports
  - Support handlers (encapsulate behaviors)
  - Can have "partners"
- Provide aggregated, compositional functions
  - Sensor Fusion
  - Drive System
  - High-level functions, e.g. Computer Vision
- Basic operations
  - Create and terminate
  - State retrieval and manipulation
  - Send Notifications



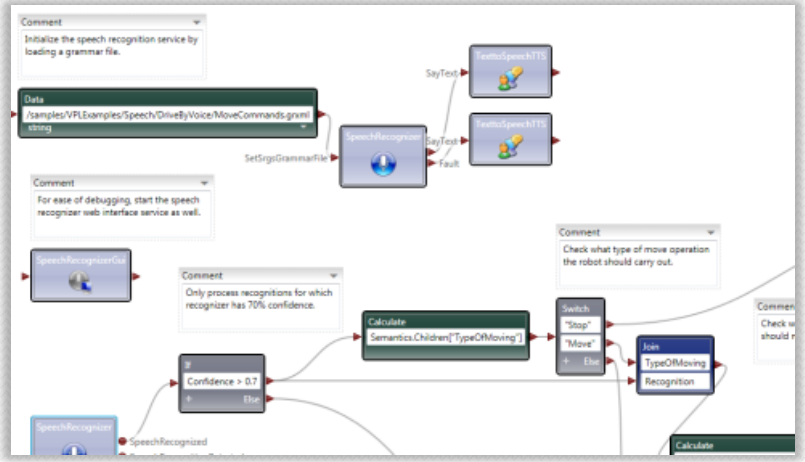
Visual Simulation Environment

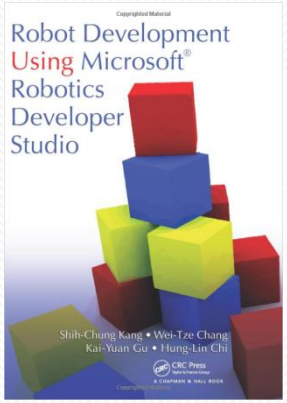
- High resolution 3D rendering
  - Several simulated environments
  - Visual and physics views
- High performance physics engine
  - NVIDIA PhysX™
  - Optional hardware acceleration
- Makes technology accessible without hardware
- Enables fast prototyping and debugging
- Extensible

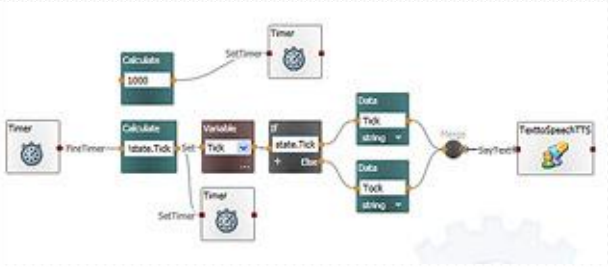


Visual Programming Language

- Dataflow editing
  - Simple connections
  - Building blocks
  - Asynchronous flows
  - Code generation
- Novice to expert
- Useful for:
  - Education
  - Dashboards
  - Prototyping
- Not useful for:
  - Image processing, etc.
  - Heavy-duty services

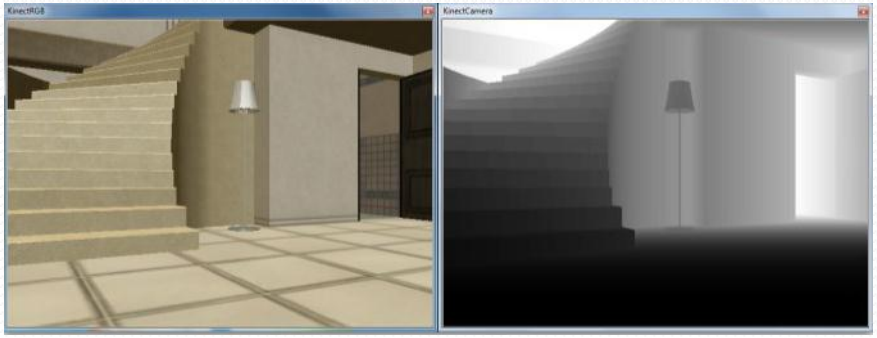






http://www.crcpress.com/product/isbn/9781439821657

Microsoft®
Robotics Developer Studio 4
Beta

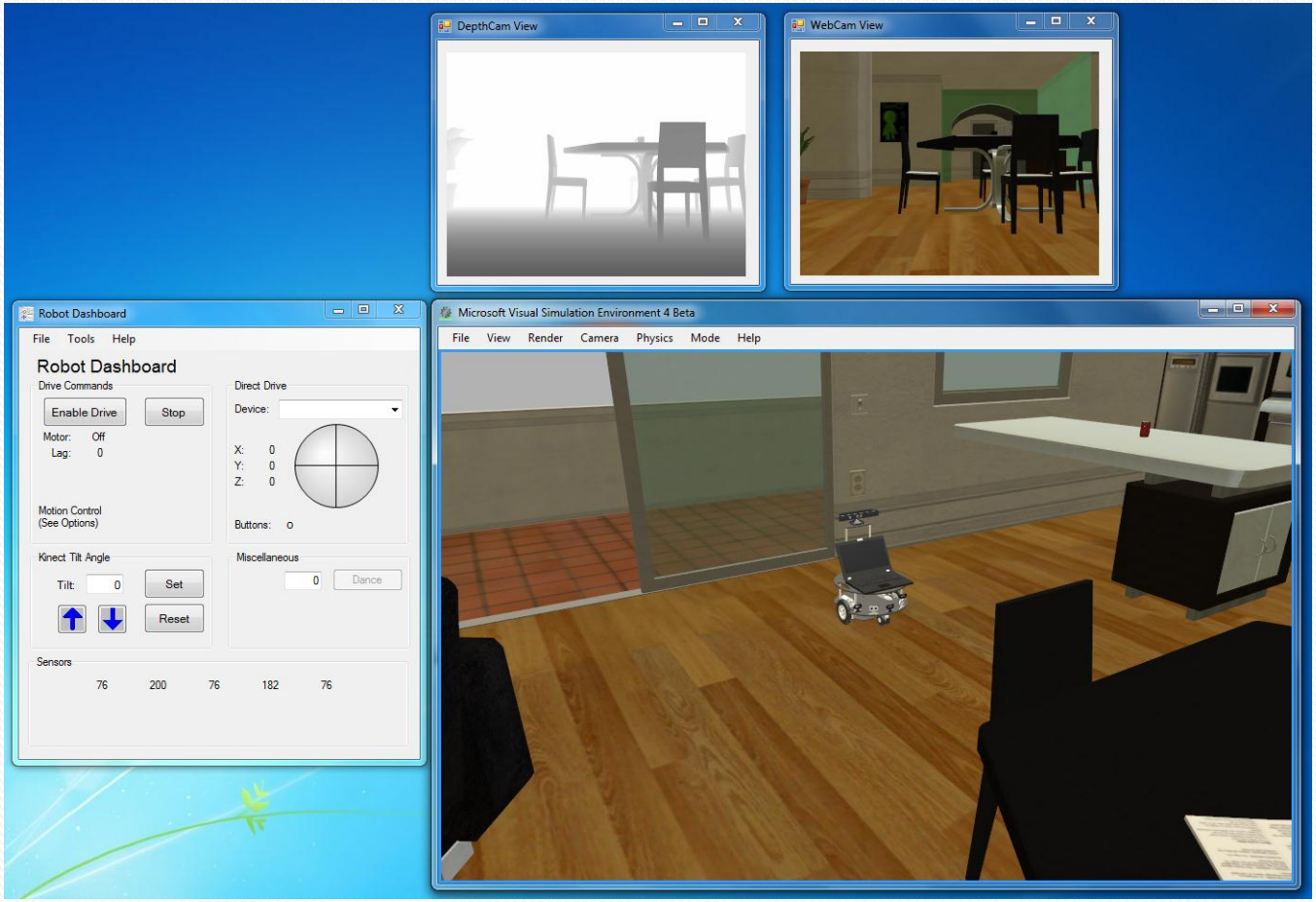


- Kinect Support
  - Uses Kinect for Windows SDK
  - RGB and Depth Image Viewers
  - Simulated Kinect Entity
- New Reference Platform Design
  - Published specification to standardize hardware
  - Differential Drive system
  - Utilizes Kinect as a key sensor
  - Onboard Computer
  - Implemented by multiple vendors
    - Parallax, Inc. (shown)
    - http://www.parallax.com/eddie

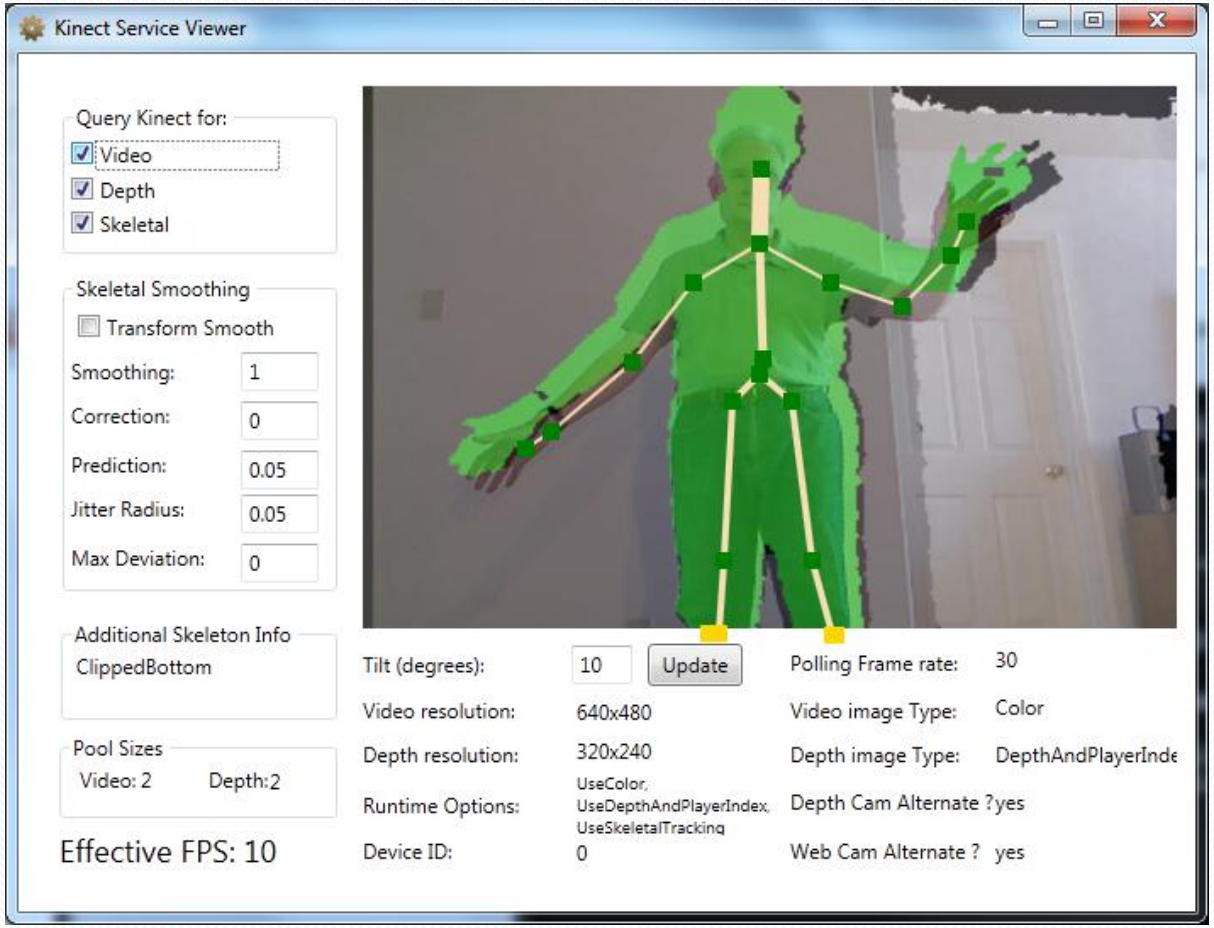
Reference Platform Design

- Sharing software is difficult if robots are not identical
- Reference Platform defines suitable hardware as the basis for a personal robotic companion
- Can support Human Robot Interaction through Gestures and Speech (via Kinect)
- Computer includes WiFi (Cloud) connectivity
- Can be extended over time as appropriate
- Available as turnkey solutions through third-parties
- Obstacle Avoidance Drive sample
- A simulation entity is also provided to allow low-cost development without hardware

Simulated Reference Platform



KinectUI Sample



Kinect Overview

Kinect Hardware

Vision System

RGB camera

IR camera

IR Laser projector

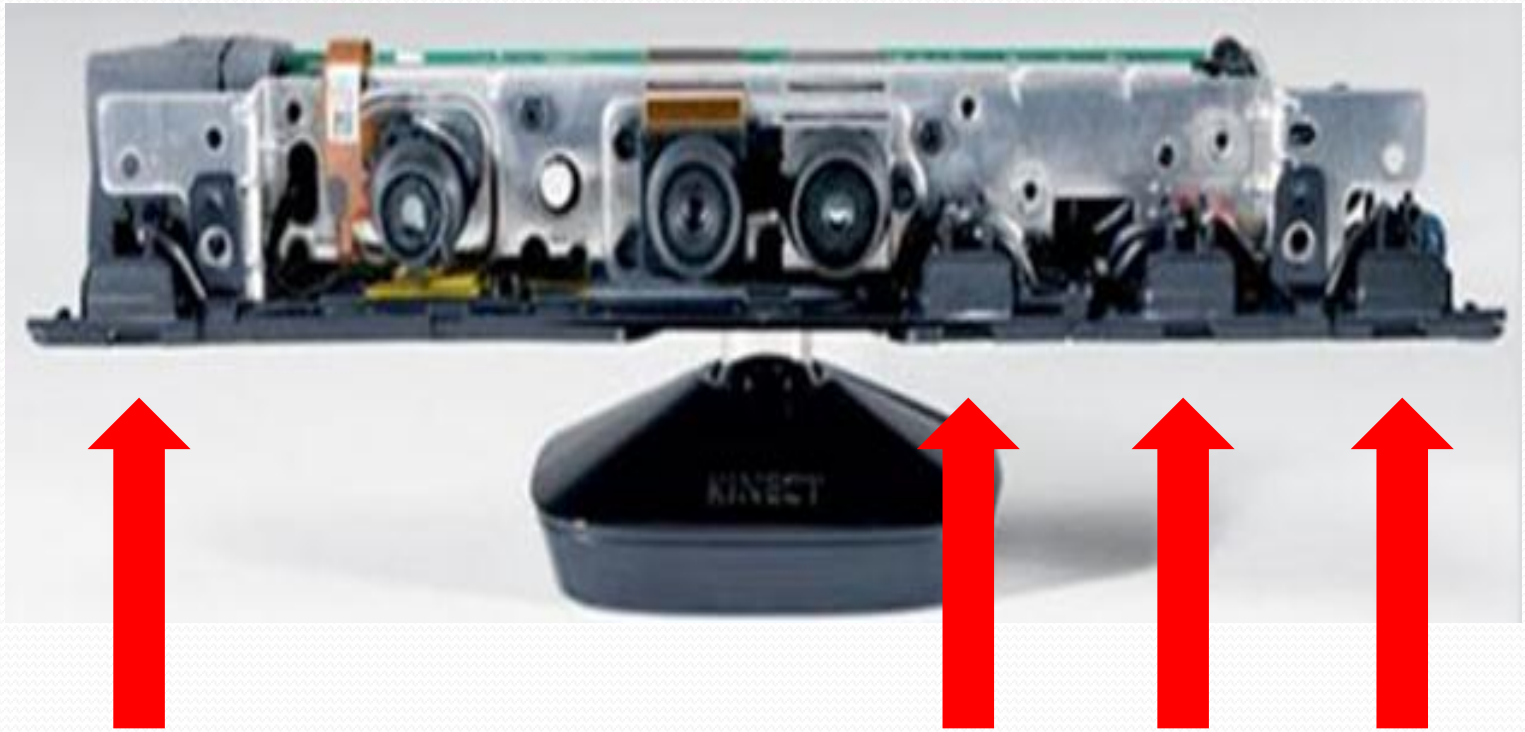
Kinect Video Output

30 Hz Frame Rate; 57 degree Field-Of-View

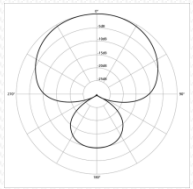
8-bit VGA RGB
640 x 480

11-bit monochrome
320 x 240

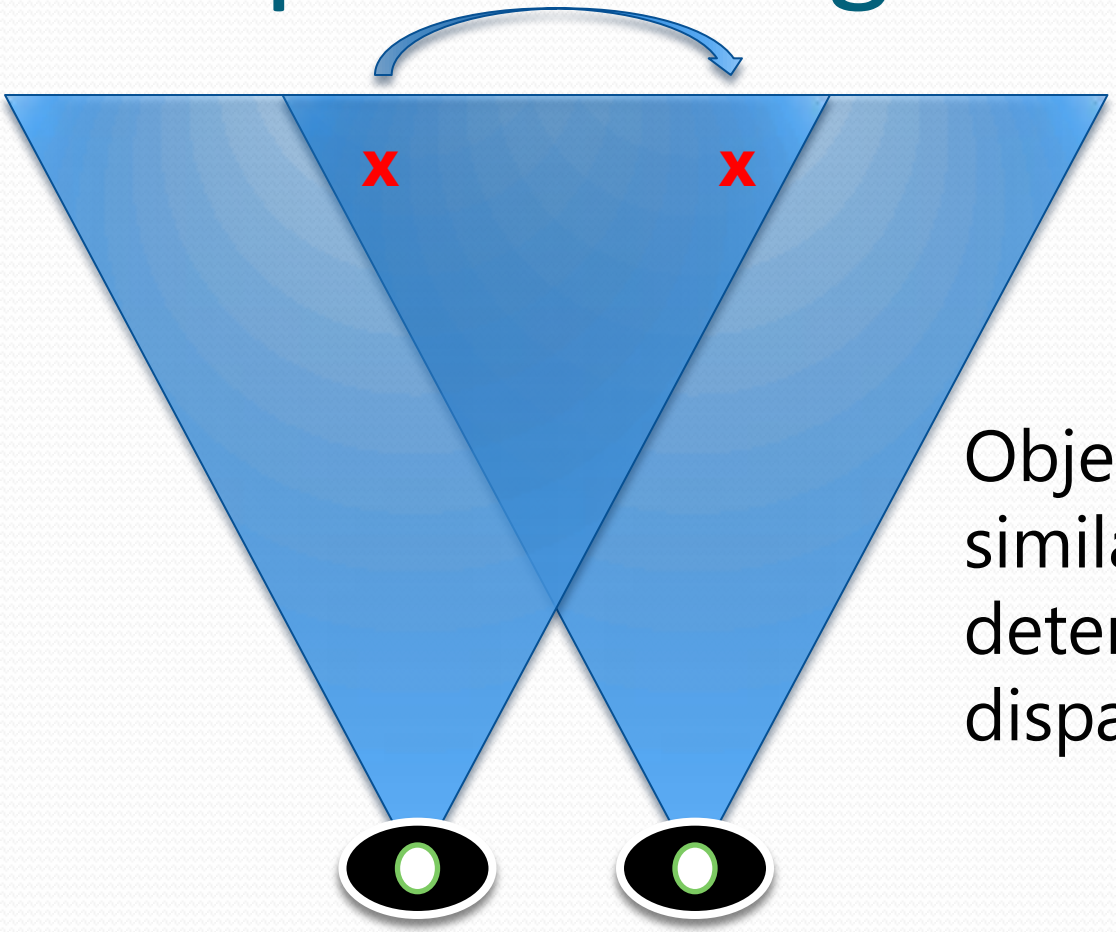
The Audio System



Supercardioid Microphone



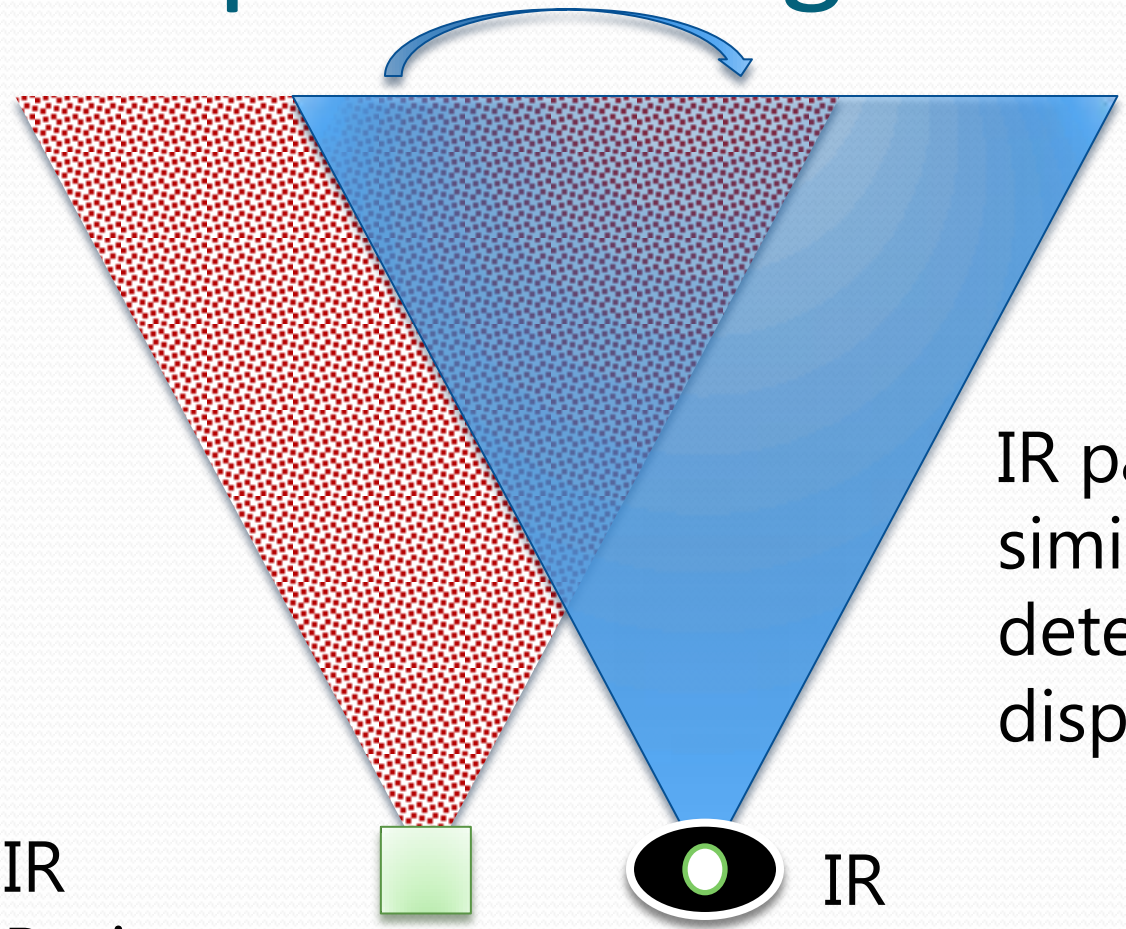
Human Depth Sensing



Object pattern similarity determines disparity

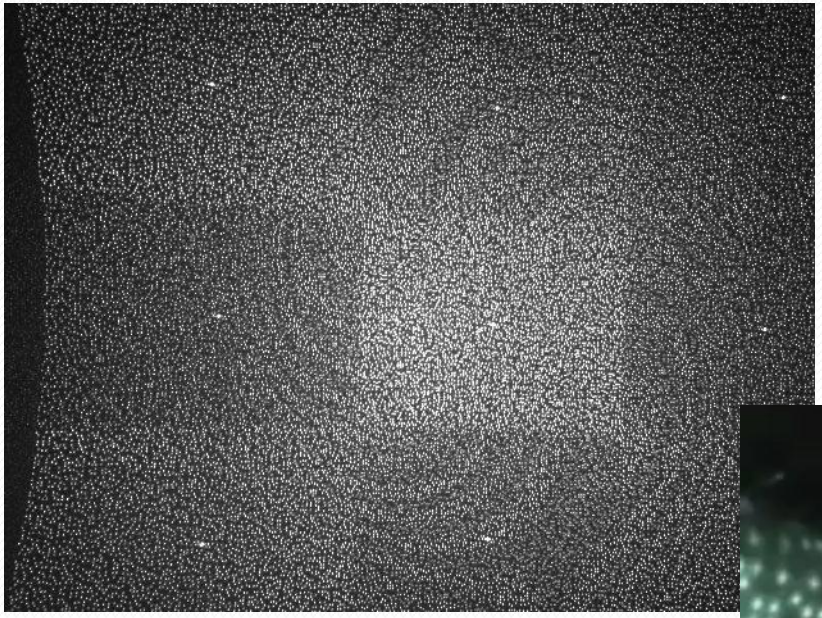
Eyes – Stereo Vision

Kinect Depth Sensing

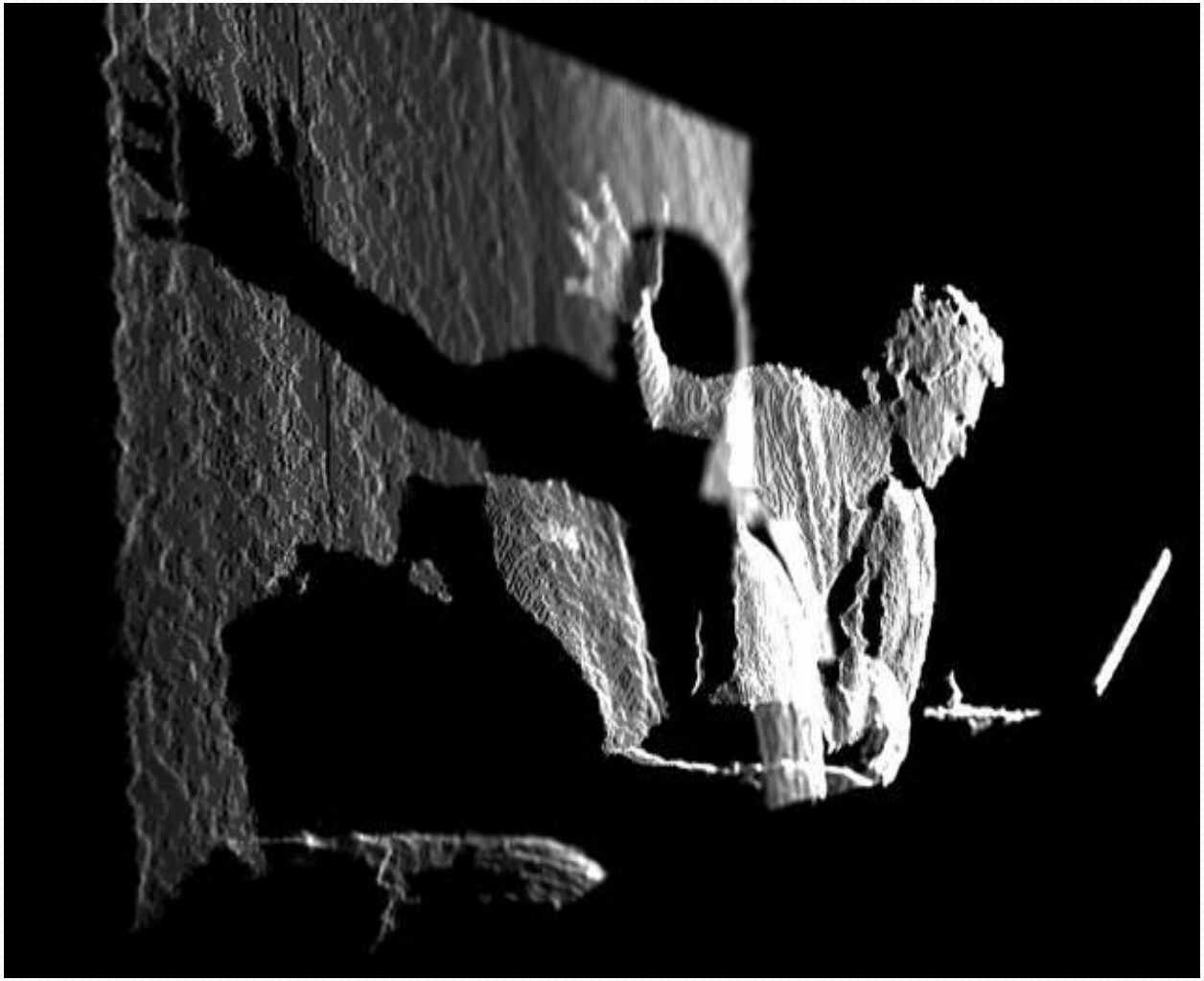


Depth Computation

Projected IR Pseudo-random Pattern

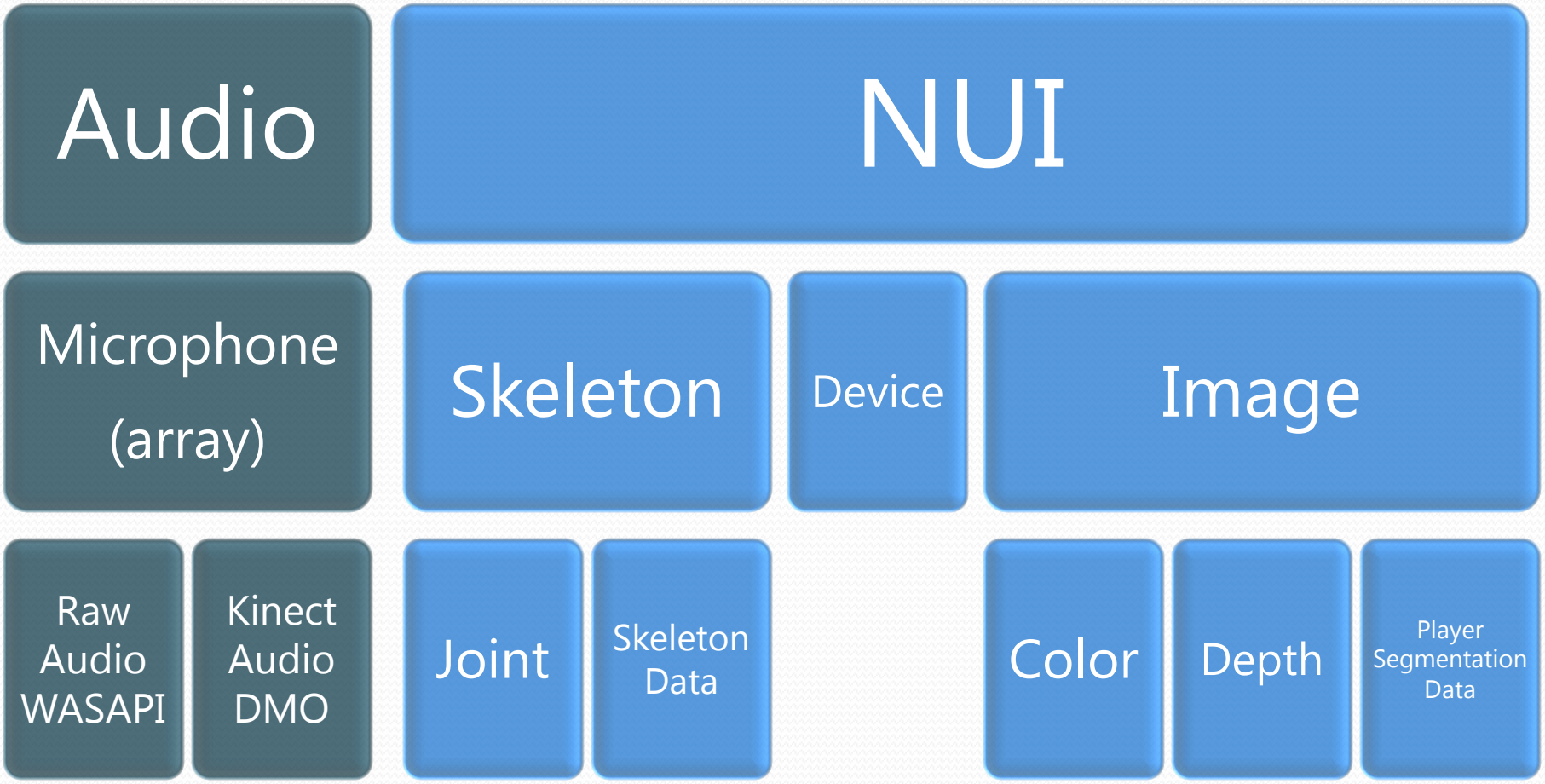


Depth Map in 3D

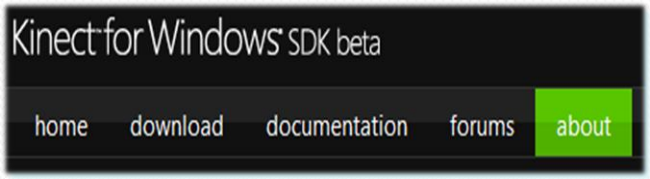


Kinect Programming

Kinect SDK Overview



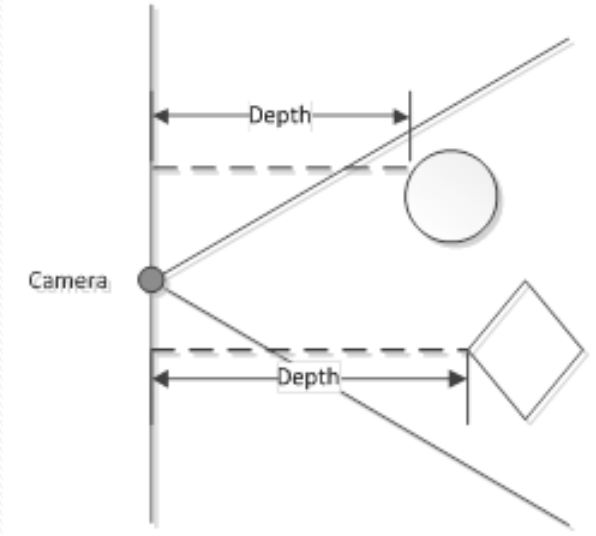
Kinect SDK



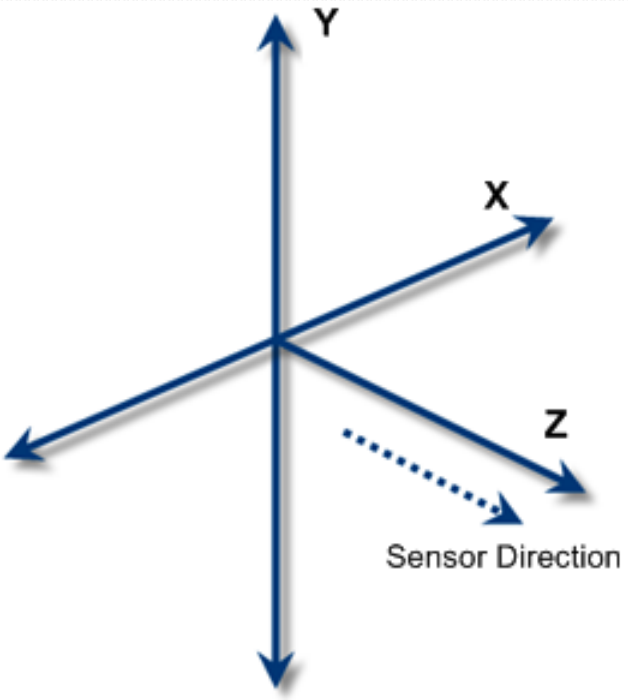
- Raw sensor streams
  - Access to raw data streams from the depth sensor, color camera sensor, and four-element microphone array enables developers to build upon the low-level streams that are generated by the Kinect sensor.
- Skeletal tracking
  - The capability to track the skeleton image of one or two people moving within the Kinect field of view make it easy to create gesture-driven applications.
- Advanced audio capabilities
  - Audio processing capabilities include sophisticated acoustic noise suppression and echo cancellation, beam formation to identify the current sound source, and integration with the Windows speech recognition API.
- Sample code and documentation
  - The SDK includes more than 100 pages of technical documentation. In addition to built-in help files, the documentation includes detailed walkthroughs for most samples provided with the SDK.
  - Assumes some programming experience.
- Easy installation
  - The SDK installs quickly, requires no complex configuration, and the complete installer size is less than 100 MB. Developers can get up and running in just a few minutes with a standard standalone Kinect sensor unit (widely available at retail outlets).
- Designed for non-commercial purposes
  - A commercial version is expected in 2012.
- Windows 7
  - Managed (C#, Visual Basic) and Unmanaged (C++) interfaces using Microsoft Visual Studio 2010.

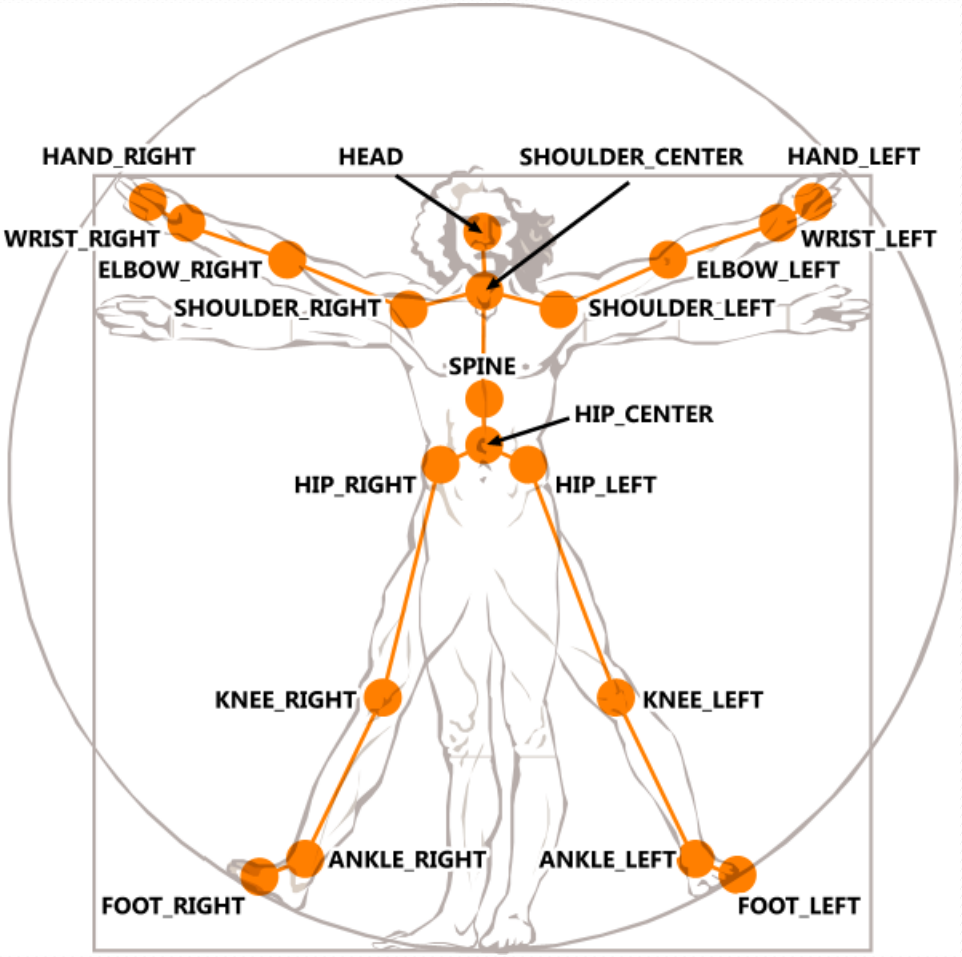
Depth Image Space

- Distance from the Kinect for every pixel in millimeters
- Optionally includes Player Index in low-order 3 bits
- Pixel coordinates do not necessarily align with RGB



Skeleton Space



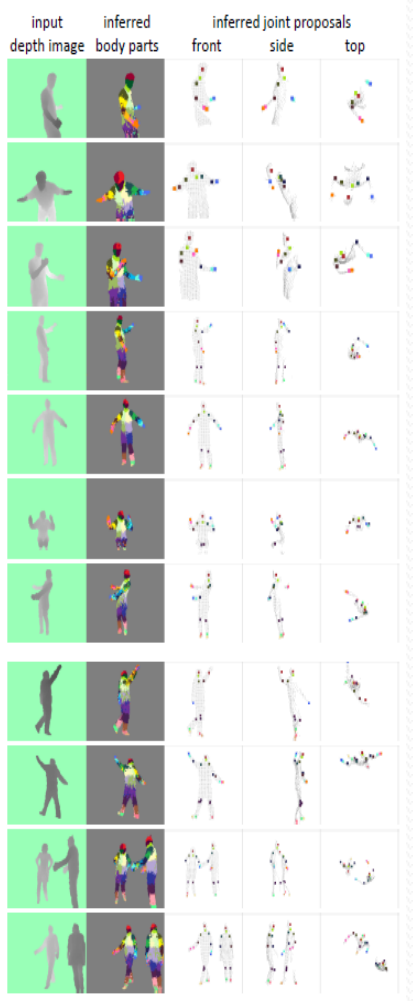


Vision Algorithm

- Quickly and accurately predict 3D positions of body joints
- From a single depth image, using no temporal information

- **Object recognition approach**
- Intermediate body parts representation that maps the difficult pose estimation problem into a simpler **per-pixel classification problem**
- Large and highly varied training dataset allows the classifier to estimate body parts invariant to pose, body shape, clothing, etc.
- Generate confidence-scored 3D proposals of several body joints by re-projecting the classification result and finding local modes

- System runs at 200 frames per second on consumer hardware
- Evaluation shows high accuracy on both synthetic and real test sets
- State of the art accuracy in comparison with related work and improved generalization over exact whole-skeleton nearest neighbor matching



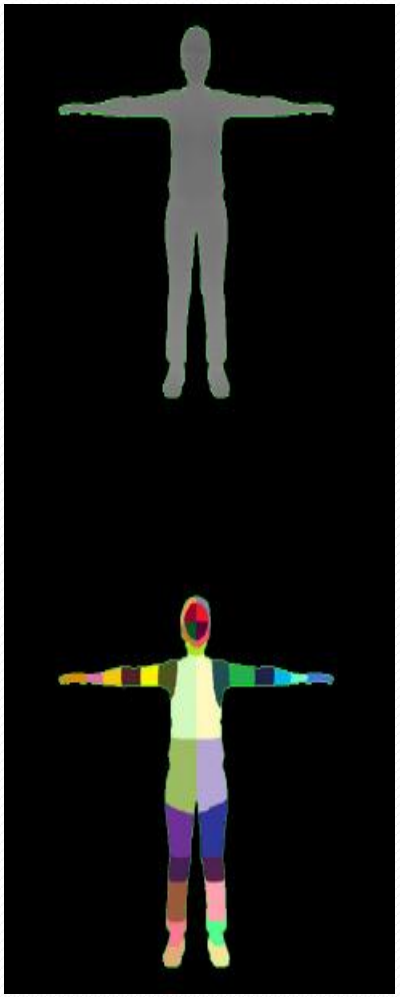
Implementation

- Collect training data – thousands of visits to global households, filming real users, the Hollywood motion capture studio generated billions of images

- Apply state-of-the-art object recognition research
- Apply state-of-the-art real-time semantic segmentation

- Build a training set – classify each pixel's probability of being in any of 32 body segments, determine probabilistic cluster of body configurations consistent with those, present the most probable

- Millions of training images → Millions of classifier parameters
- Hard to parallelize → New algorithm for distributed decision-tree training
- Major use of DryadLINQ (large-scale distributed cluster computing)



Depth Reference

- 16 bits per pixel (but not all used)
- Distance Range: 800 mm to 4000 mm range
- Values are not linearly distributed across the range
- Depth value 0 means unknown
  - Shadows, low reflectivity, and high reflectivity among the few reasons
- Player Index
  - 0 – No player
  - 1 – Skeleton 0
  - 2 – Skeleton 1
  - … (Not yet implemented)

Kinect Frames

- RDS: QueryRawFrameRequest returns a RawKinectFrames object:

```
public ImageFrame RawDepthFrameData { get; set; }
public ImageFrame RawImageFrameData { get; set; }
public SkeletonFrame RawSkeletonFrameData { get; set; }
```

- Depth Frame is 16-bit depth values (more info later)
- Image Frame is 24-bit RGB
- Skeleton Frame contains an array of Joints

Understanding Depth Data

- ImageFrame.Image.Bits

```
public struct PlanarImage
    {
        public byte[] Bits;
        public int BytesPerPixel;
        public int Height;
        public int Width;
    }
```

- Depth data is an array of bytes
- Array details:
  - Starts at top left of image
  - Moves left to right, then top to bottom
  - Represents distance for pixel

Calculating Distance

- 2 bytes per pixel (16 bits)
- Depth – Distance per pixel
  - Bitshift **second byte by 8**
  - Distance (0,0) = `(int)(Bits[i] | Bits[i+1] << 8);`
- DepthAndPlayer Index – Includes Player index
  - Bitshift by **3 first byte** (player index), **5 second byte**
  - Distance (0,0) =`(int)(Bits[i] >> 3 | Bits[i+1] << 5);`

Joint Information

- Maximum two players tracked at once
- Each player has a set of <x, y, z> joints in meters
- Each joint has associated state
  - Tracked, Not tracked, or Inferred
- Inferred - Occluded, clipped, or low confidence joints
- Not Tracked - Rare, but your code must check for this
- RDS: Can convert joints from Skeleton Space back to Depth Image Space using a QueryPixelMappingRequest (one pixel at a time)

Joint Smoothing

- Use to remove joint “noise”
  - Small, high frequency jitter
  - Temporary Spikes
- `nui.SkeletonEngine.TransformSmooth = true;`
- Fine tune using `TransformSmoothParameters`
  - Correction, JitterRadius, MaxDeviationRadius, Prediction, Smoothing
- RDS: Set using UpdateSkeletalSmoothingRequest

Kinect Body Tilt

- ± 27 degrees
- Servo is not very strong and body might tilt if the robot makes sudden movements
- Maximum of 15 requests per 2 minutes
- Tilt angle accounts for direction of gravity using the Accelerometer so that Zero is Horizontal regardless of any tilt of the Kinect base
- RDS: Set tilt angle using UpdateTiltRequest
- RDS: Issue a Get request to find current tilt angle (measured by the Accelerometer)

Audio Processing

- Operates independently of Depth/RGB
- Kinect supports:
  - Multichannel Echo Cancellation
  - Audio Beam Forming (Sound Source Localization)
- Appears as a Windows Audio Device
- RDS: Speech Recognition service can process speech based on a Grammar. Does not provide general recording capability.

Speech Recognition

- Grammar – What to listen for
  - Code – GrammarBuilder, Choices
  - Speech Recognition Grammar Specification (SRGS)
    - C:\Program Files (x86)\Microsoft Speech Platform SDK\Samples\Sample Grammars\
- Note: AutomaticGainControl must be off
- RDS: SpeechRecognizedNotification sent to subscribers contains the Confidence, Text and Semantics
- RDS: BeamDirectionChangedNotification sent to subscribers contains StartTime, Angle and Confidence

Grammar Examples

```
<!-- Confirmation_YesNo._value: string
["Yes", "No"] -->
<rule id="Confirmation_YesNo"
scope="public">
  <example> yes </example>
  <example> no </example>
  <one-of>
    <item>
      <ruleref uri="#Confirmation_Yes" />
    </item>
    <item>
      <ruleref uri="#Confirmation_No" />
    </item>
  </one-of>
  <tag> out = rules.latest() </tag>
</rule>
```

```
<!-- Confirmation_Yes._value: string ["Yes"]
-->
<rule id="Confirmation_Yes" scope="public">
  <example> yes </example>
  <example> yes please </example>
  <one-of>
    <item> yes </item>
    <item> yeah </item>
    <item> yep </item>
    <item> ok </item>
  </one-of>
  <item repeat="0-1"> please </item>
  <tag> out._value = "Yes";</tag>
</rule>
```

Further Information - RDS

- Download Microsoft Robotics Developer Studio 4 Beta
  - http://www.microsoft.com/robotics
- Join the RDS Community
  - http://social.msdn.microsoft.com/Forums/en-US/roboticscommunity
- Watch the Overview Video on Channel 9
  - http://channel9.msdn.com/posts/A-Look-At-Robotics-Developer-Studio-4-Beta
- RDS Reference Platform Specification
  - http://go.microsoft.com/fwlink/?LinkID=228540
- Take part in the Robotics @ Home Contest
  - http://www.roboticsathome.com

Further Information - Kinect

- Download Kinect for Windows SDK Beta
  - http://research.microsoft.com/kinectsdk/
- Join the Kinect Community
  - http://social.msdn.microsoft.com/Forums/en-US/category/kinectsdk
- Quick Starts on Channel 9
  - http://channel9.msdn.com/Series/KinectSDKQuickstarts
- Launch Events
  - http://channel9.msdn.com/Events/KinectSDK/BetaLaunch
- Gallery
  - http://channel9.msdn.com/coding4fun/kinect/