

The Kinect distance sensor as human-machine-interface in audio-visual art projects

Toningenieur-Projekt

Matthias Kronlachner

June 26, 2012

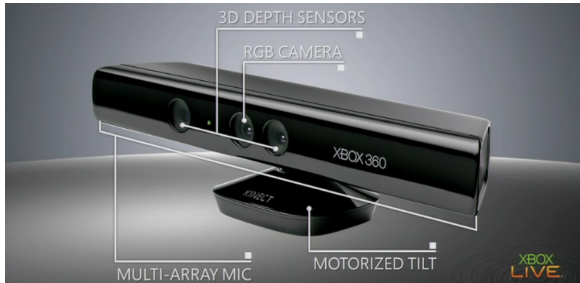
Table of contents

- 1 Goals
- 2 Kinect specifications
 - RGB Camera
 - Depth Sensor
 - Microphone Array
- 3 Access the Kinect
 - libfreenect
 - OpenNI
 - Microsoft Kinect SDK
- 4 PD/Gem externals
- representation of depth data
- pix_freenect, freenect, freenect_audio
- pix_openni
- 5 Application
 - General application examples
 - Concrete projects
- 6 Conclusion
 - Pros and cons
 - Outlook

Goals of the project

- connecting Kinect device and Pure Data
- making high level functions available (Natural Interaction)
- documentation and example patches
- concrete applications in art projects

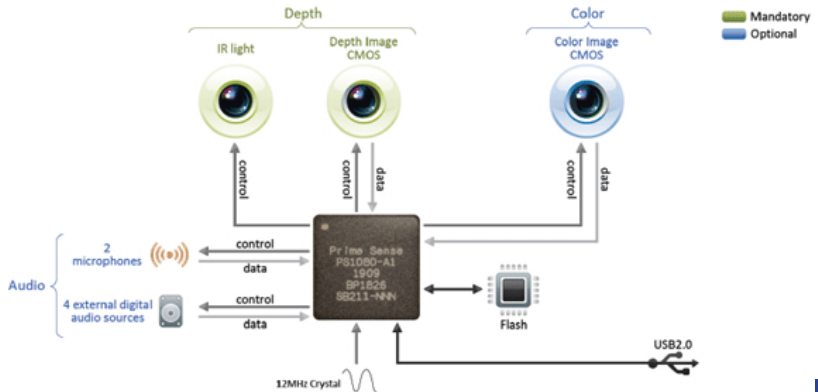
Kinect hardware



Kinect Sensor

- RGB camera
- Depth sensor
- 4 channel microphone array
- tilting head $\pm 27^\circ$
- 3 axis accelerometer
- 3 color LED

PrimeSense - block diagram reference design

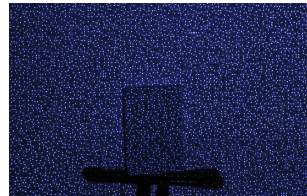
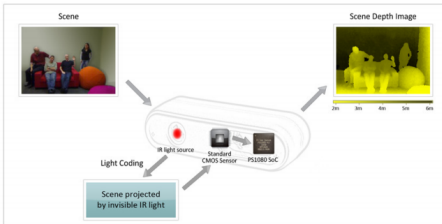


RGB Camera

- 640x480 @ appr. 30 Hz framerate
- max. 1280x1024 @ appr. 15 Hz framerate



Depth Sensor - function



Pattern of infrared dots



Source: <http://www.futurepicture.org>
<http://www.anandtech.com>

Depth Sensor

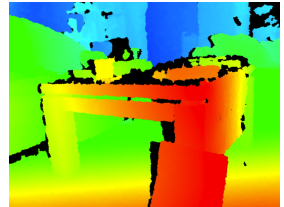
- 640x480 pixels, rightmost 8 columns “no data”
usable: → 632x480
- raw: 11 bit depth value → 2047 if value cannot be estimated
- range: 0.7 to 7 m (optimum range 1.2 to 3.5 m)



rgb image

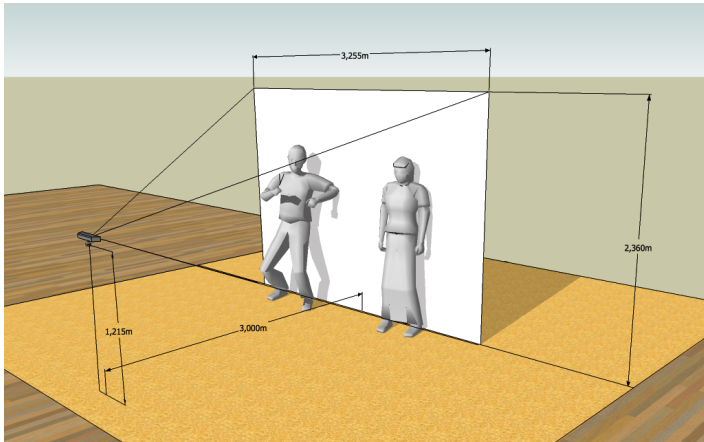


infrared image



colorized depth image

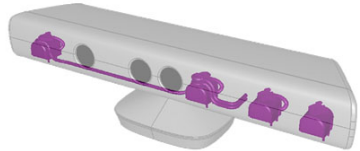
Depth Sensor - Field of view



Field of view 58° H, 45° V, 70° D

Microphone Array

- 4 channels
- 16 bit @ approximately 16 kHz
- 1 mics on the left, 3 mics on the right side
- send 5.1 signal to Kinect for echo cancellation



Access the Kinect

- 1 Goals
- 2 Kinect specifications
 - RGB Camera
 - Depth Sensor
 - Microphone Array
- 3 Access the Kinect
 - libfreenect
 - OpenNI
 - Microsoft Kinect SDK
- 4 PD/Gem externals
- representation of depth data
- pix_freenect, freenect, freenect_audio
- pix_openni
- 5 Application
 - General application examples
 - Concrete projects
- 6 Conclusion
 - Pros and cons
 - Outlook

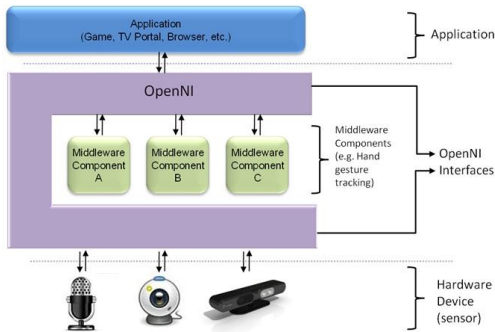
libfreenect

- Open Source (Apache v2/GPL v2 license)
- Windows, OSX, Linux
- ✓ RGB/IR and Depth Image
- ✓ multiple devices by serial number
- ✓ Audio streams
- ✓ LED and Motor tilt control
- ✓ get Accelerometer data
- ✗ no higher level functions (tracking, ...)

→ pix_freenect, freenect is based on libfreenect

OpenNI - Open Natural Interaction

- industry-led, not-for-profit organisation ? (PrimeSense)
"certify and promote the compatibility and interoperability of Natural Interaction (NI) devices"



Source: <http://www.openni.org>

OpenNI - Open Natural Interaction

- ✓ Open Source
- ✓ device independent
- ✓ Record and playback streams
- ✗ Kinect driver without audio, accelerometer, LED and motor tilt access

Primesense NITE Middleware

- ✓ Binaries for Windows, Mac OS, Linux
- ✓ Scene analysis/segmentation (User tracking)
- ✓ Hand and Skeleton tracking
- ✗ high level function just with one device
- ✗ closed source

→ pix_openni is based on OpenNI and NITE middleware

Microsoft Kinect for Windows SDK

- ✓ all functions of Kinect supported
- ✓ commercial license since final version, easy to use
- ✓ Face and Skeleton (standing and sitting) tracking
- ✓ sound source localization, beam forming
- ✓ speech recognition
- ✗ Windows only, closed source!



Source: www.microsoft.com

PD/Gem externals

- 1 Goals
- 2 Kinect specifications
 - RGB Camera
 - Depth Sensor
 - Microphone Array
- 3 Access the Kinect
 - libfreenect
 - OpenNI
 - Microsoft Kinect SDK
- 4 PD/Gem externals
 - representation of depth data
 - pix_freenect, freenect, freenect_audio
 - pix_openni
- 5 Application
 - General application examples
 - Concrete projects
- 6 Conclusion
 - Pros and cons
 - Outlook

representation of depth data

- ✗ no colorspace for depth values in Gem
- ✓ visual representation → RGB or Gray mapping



- ✓ current raw output solutions
 - RGBA
 - YUV422

→ need for pix_expr

R	G	B	A
3/8 msb	8 lsb	0 or userid	255

YUV422 (2 bytes per pixel)
11 bit/16bit depth values

two Kinects, upper pair with Zoom lense

pix_openni

The screenshot displays the **pix_openni** application interface, which is divided into several sections:

- Top Left:** A video feed showing a person's hand and arm. Below the video, there are labels for "hand 2" and "ge".
- Top Right:** A depth map of the scene, showing the person's hand and arm in a different color scheme.
- Bottom Left:** A panel titled "properties for pix_openni" with a "TURN ON/OFF STREAMS" section. It includes checkboxes for "rgb \$1", "depth \$1", "usergen \$1", "skeleton \$1", and "hand \$1". Below these are "s \$0-prop" and "hand \$1" labels.
- Bottom Center:** A panel titled "change video/depth mode of OpenNI" with a "video_mode 640 480 30" and "depth_mode 1280 1024 15" section. It also includes a "bang" button and a "view available modes" label.
- Bottom Right:** A panel titled "skeleton specific" with a "stop skeleton for" section. It includes a "stop_user \$1" button and a "start calibration for all users" button. Below these are "start_user \$1" and "spec. user" labels.
- Top Right (Patched):** A patch window titled "pix_openni-help.pd" showing a Pure Data patch. It includes a "pd properties" object, a "r \$0-prop" object, a "gemhead" object, a "pix_openni 1 1 1 0 0" object, a "translateXYZ -3 2 0" object, a "pix_texture" object, a "loadbang" object, a "rectangle 4 3" object, a "pd pix_info" object, a "tracking_data" object, a "route user" object, a "print" object, a "DEPTH Image" object, a "translateXYZ 3 2 0" object, a "pix_texture" object, a "loadbang" object, a "rectangle 4 3" object, a "pd distance_measurement" object, and a "color 0.5 0.5 0.5" object.

pix_openni with skeleton tracking, hand tracking and usercoloring

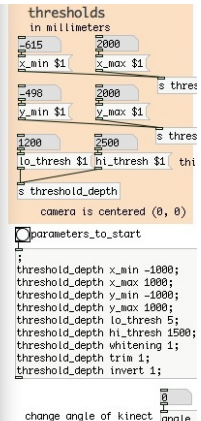
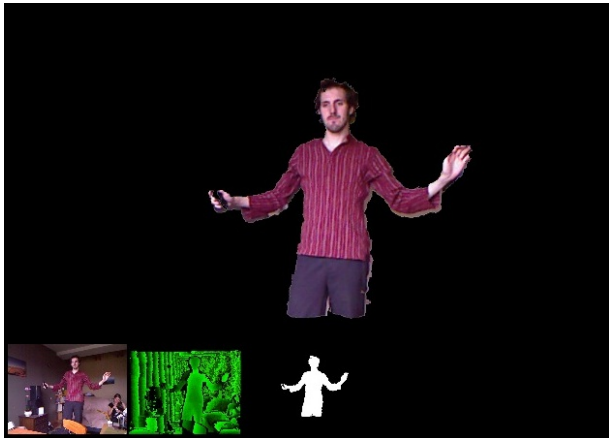
Application

- 1 Goals
- 2 Kinect specifications
 - RGB Camera
 - Depth Sensor
 - Microphone Array
- 3 Access the Kinect
 - libfreenect
 - OpenNI
 - Microsoft Kinect SDK
- 4 PD/Gem externals
- representation of depth data
- pix_freenect, freenect, freenect_audio
- pix_openni
- 5 **Application**
 - General application examples
 - Concrete projects
- 6 Conclusion
 - Pros and cons
 - Outlook

Distance measurement



Background subtraction



User tracking with pix_openni

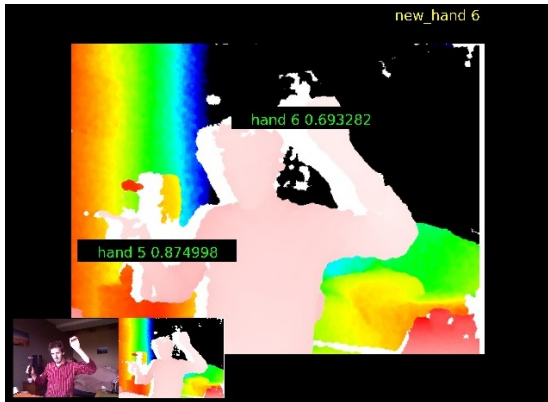
The screenshot displays a Pure Data patch titled "turn_on_off" used for user tracking with the Kinect sensor. The patch includes several components and a message box:

- turn_on_off** (orange box): Contains two checked checkboxes for `nite-userdata` and `blob-detection`.
- Message Box** (white box):

```
r init
delay 2000
[nite-userdata]
[parameters_to_start]
;
/openni usergen 1;
/openni usercoloring 1;
/openni depth_output 1;
/openni real_world_coords 0;
[nite-userdata]
[blob-detection 1];
```
- Angle Control** (bottom right): A slider labeled "change angle of kinect" with a value of 0, connected to an `angle $` message box and a `/free` object.

The main window shows a black background with a white silhouette of a person. Text in the top right corner reads "num_users 1". A green label "user1 z: 2.01774" is positioned near the silhouette. At the bottom left, there are three small video windows: the leftmost shows a person in a room, the middle shows a grayscale depth map, and the rightmost shows a binary blob map.

Hand tracking with pix_openni



```
declare -path ../externals
following File:
>.ini and uncomment:
hands=1
hands=1
```

turn_on_off

☒ nite-userdata

are to start hand tracking!!

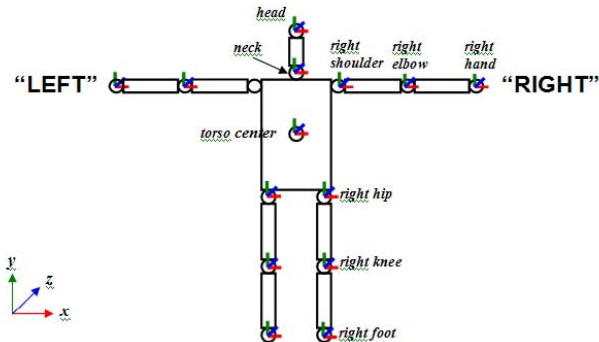
```
create, 1
```

```
y init
delay 2000
parameters_to_start
;
/openni depth_output 0;
/openni real_world_coors 0;
/openni hand 1;
nite-userdata 1;
```

- * depth_output doesn't matter
- * for displaying i use non real world coordinate
- * hand tracking can be turned on by
pix_openni creation argument or message

change angle of kinect

Skeleton tracking with pix_openni

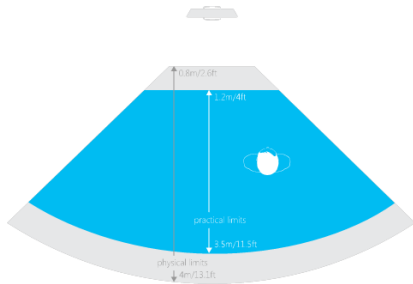


NOTE 1: Skeleton's front side is seen in this figure

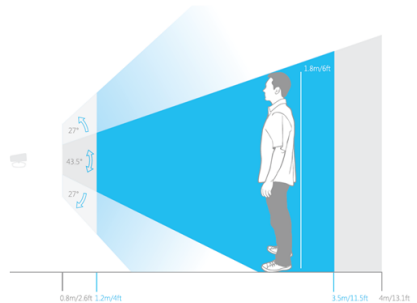
NOTE 2: Upper arm is twisted such that if elbow is flexed the lower arm will bend forwards towards sensor.

Skeleton joints

Skeleton tracking with pix_openni



horizontal field of view



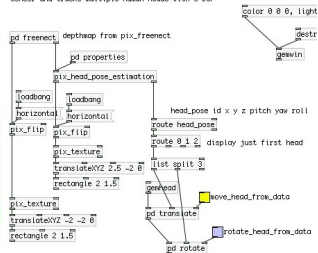
vertical field of view

pix_head_pose_estimation

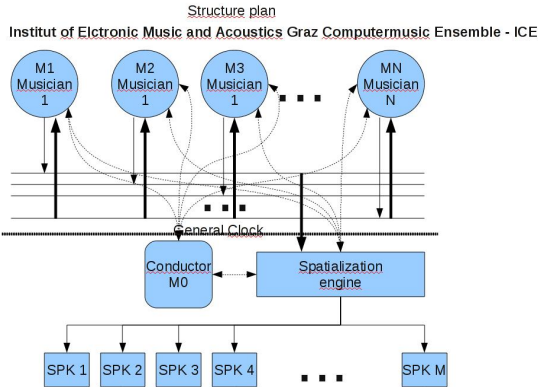
- Standalone application by Gabriele Fanelli, ETH Zurich
- ✓ multiple users
- ✓ depends on OpenCV
- ✗ 6 DOF, can detect angles about $\pm 75^\circ$ yaw, $\pm 60^\circ$ pitch and $\pm 50^\circ$ roll



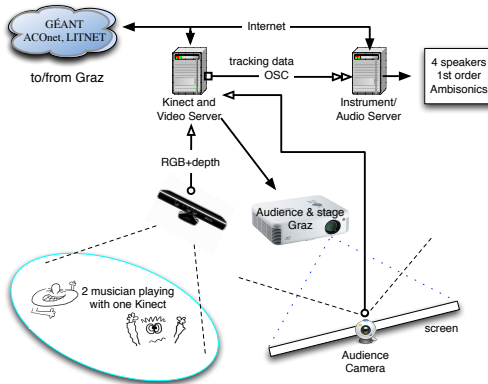
sensor and tracks multiple human heads with 6 DOF



IEM Computermusic Ensemble (ICE) structure

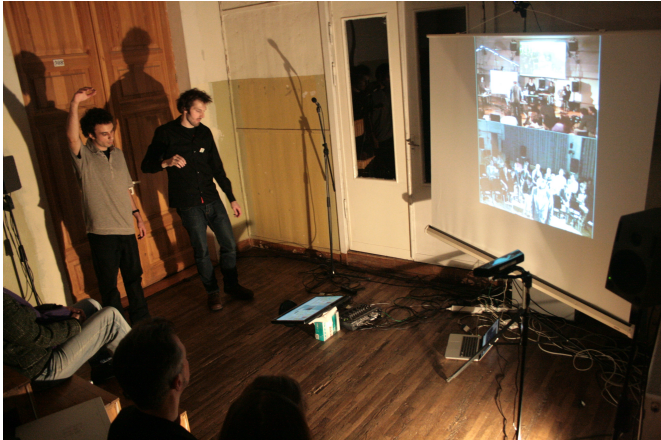


ICE Network concert - Graz - Vilnius -



ICE - Stage Setup Vilnius 20.3.2012

ICE Network concert - Graz - Vilnius



ICE @ Vilnius 20.3.2012

stage setup - vertimas

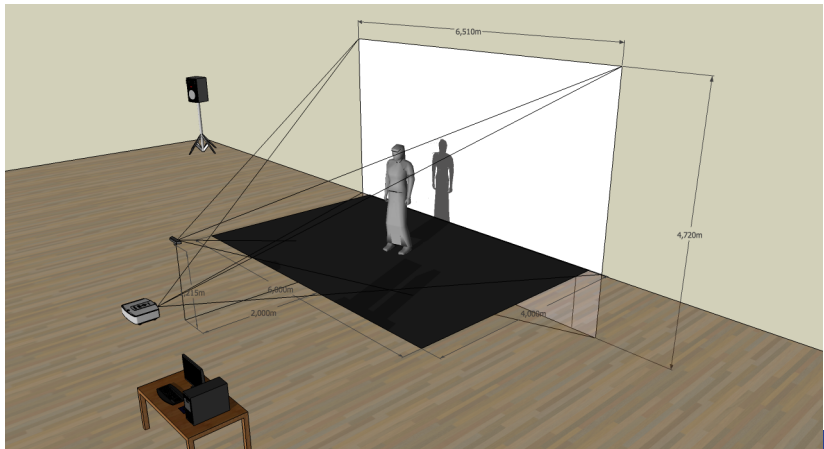
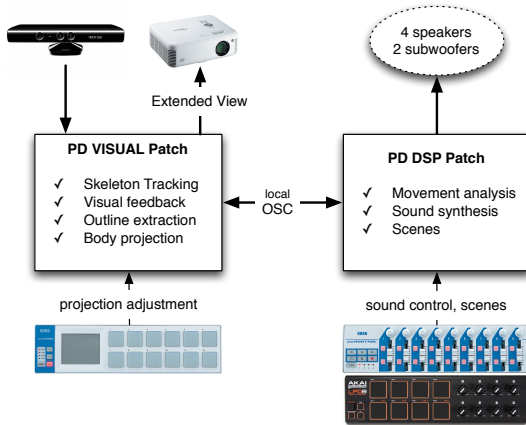


diagram - vertimas



vertimas - übersetzen

demonstration
video - trailer

Pros and cons

- ✓ price, availability
- ✓ ease of use
- ✓ multifunctional
- ✗ range, resolution
- ✗ multiple devices
- ✗ high level functions dependent on big companies

Questions...?

Thank you...