Kinect in Mac: Quartz Composer and Processing

A brief tutorial

by NIRTeam @ TEI Crete

Contents

- Using Quartz Composer with Kinect
- Using Processing programming language with kinect
- 3D scanning using Kinect

Quartz Composer (QC)

- Node-based visual programming language used mainly for processing and rendering graphics.
- Included in the Xcode developer tools package by Apple
- Quartz Composer uses OpenGL (including GLSL), OpenCL, Core Image, Core Video, JavaScript, and other technologies to build an API and a developer tool

Data types inside QC

Boolean - a boolean value, 0 or 1

Index - a positive integer between 0 and 2147483647

Number - a double precision floating point number

String - a unicode string

Color - an RGBA or CMYK quartet, or a Grayscale value

Image - a 2D image of arbitrary (possibly infinite) dimensions

Structure - a named or ordered collection of objects, including nested structures

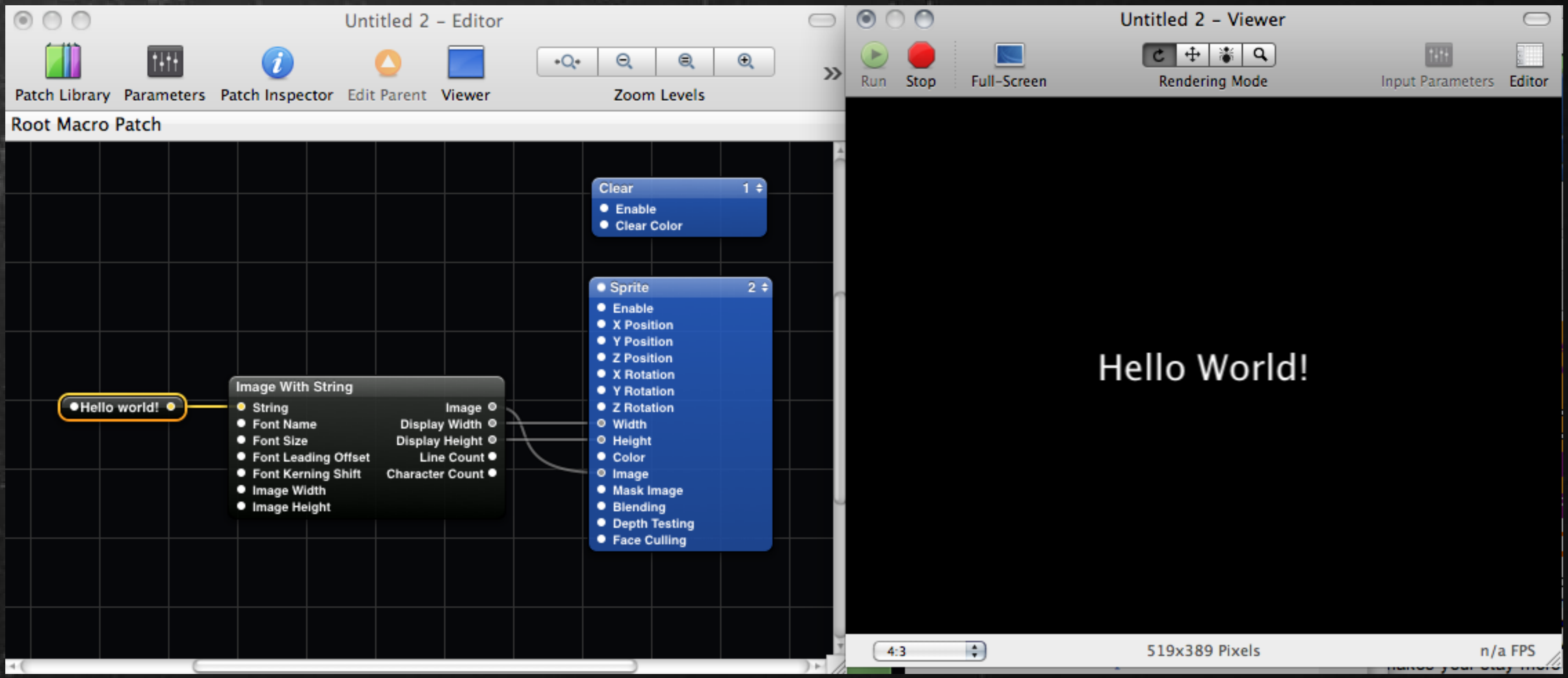
Virtual - any of the above

Mesh - a collection of vertices, and per-vertex normals, texture coordinates, and colors in 3-space.

Interaction - a valueless type used to associate user input with user-interactive elements of the composition.

QC Programming

QC Programming = connecting different nodes (Patches).



QC Patches

The patches are divided into three categories:

1. "Providers" 2. "Processors" 3. "Consumers"

QC resources

- http://kineme.net/
- http://quartzcomposer.com/compositions
- http://www.quartzcompositions.com/
- http://guides.macrumors.com/Quartz_Composer

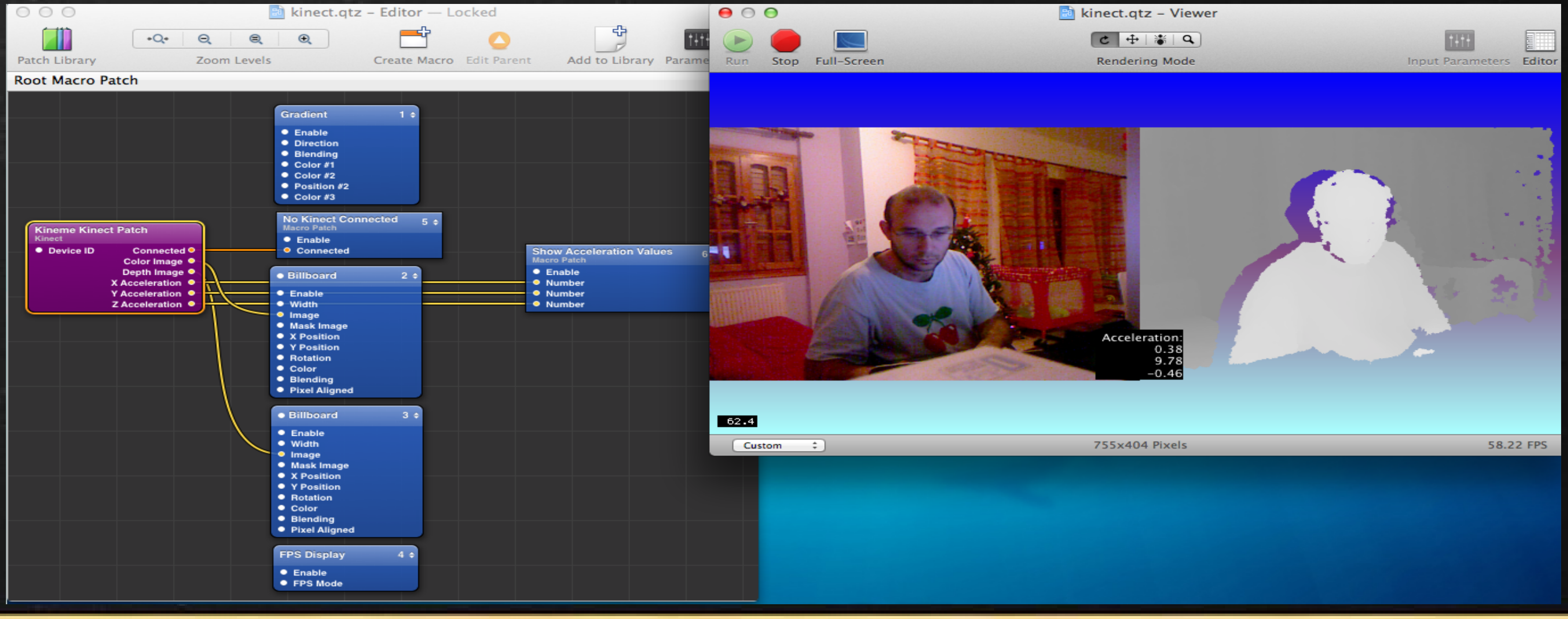
Kinect & QC

Kineme KinectTools plugin
http://kineme.net/KinectTools
Synapse plugin
http://synapsekinect.tumblr.com/
v002 open Kinect example
http://kineme.net/forum/Discussion/Programming/v002OpenKinectBeta

Quartz-Composer-Open-Kinect-Plugin
https://github.com/stoulouse/Quartz-Composer-Open-Kinect-Plugin
UIO Kinect with TUIO client QC Plugin
https://code.google.com/p/tuiokinect/
http://prdownloads.sourceforge.net/reactivision/TUIO_Quartz-1.4.zip?download
Tryplex Toolkit
http://code.google.com/p/tryplex/

Kinect & QC: Kineme KinectTools plugin

- Retrieves color and depth image data from the Kinect.
- Based on libfreenect



Kinect & QC: Synapse

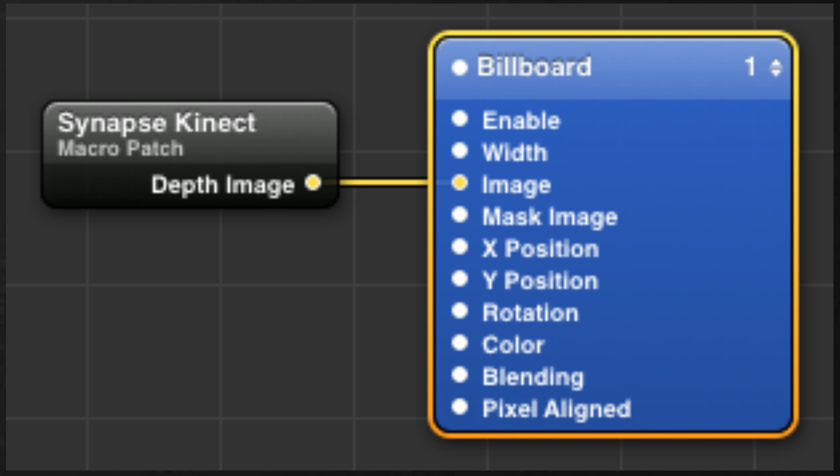
- Based on OpenNI
- Allows Kinect to control applications that can receive OSC events (such as Ableton Live, Quartz Composer, Max/MSP/Jitter)
- Consists of: application that sends joint and event data via OSC, some Max for Live patches to use those messages to control Ableton, along with a plugin for Quartz Composer to read in the depth buffer image.

Kinect & QC: Synapse app

To bind a skeleton to yourself, get in front of the camera, and hold the "psi" pose for several seconds:

Kinect & QC: Synapse plugin

Sends joint positions and hit events via OSC, and also sends the depth image into Quartz Composer



Kinect & QC: Synapse plugin example

http://www.youtube.com/watch?v=mXfd8NmjCJk

Kinect & QC: v002 open Kinect plugin

- libfreenect based
- Provides floating point depth image output, tilt control (+- 30°), color or infra red image, and accelerometer output.

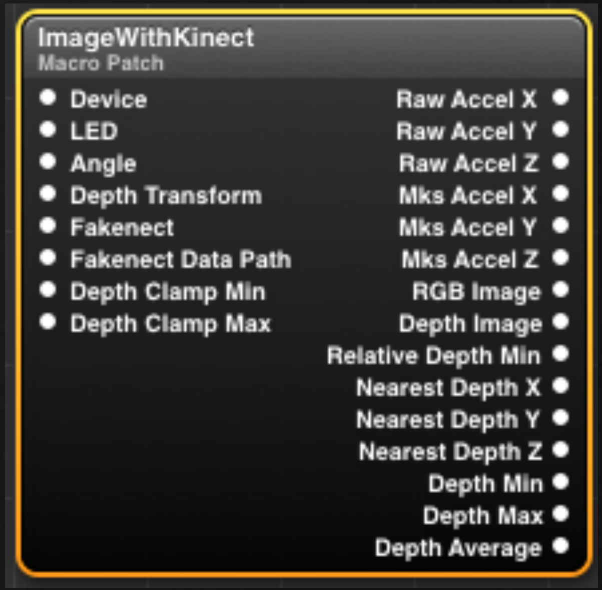
Kinect & QC: v002 open Kinect example

http://www.youtube.com/watch?v=0gNngdDEel4

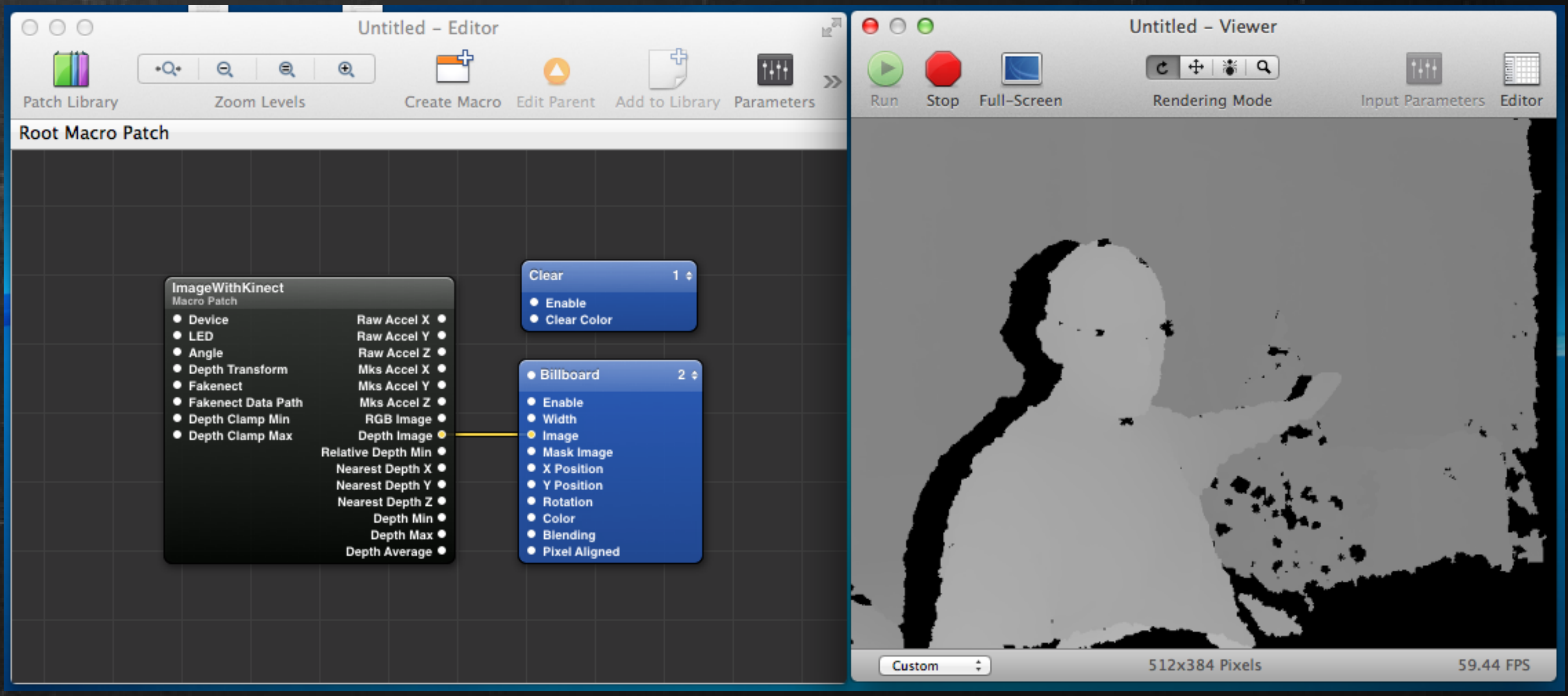
Kinect & QC: Quartz-Composer-Open-Kinect-Plugin

Features:

- kinect and fakenect support
- depth and rgb images
- depth gamma transform
- depth clamping
- depth proximity
- multiple device



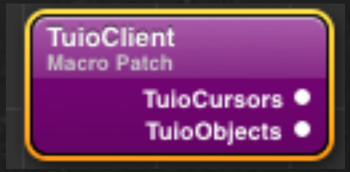
Kinect & QC: Quartz-Composer-Open-Kinect-Plugin example



Kinect & QC: TUIO Kinect with TUIO client QC Plugin

- TUIO Kinect: TuioKinect tracks simple hand gestures using the Kinect controller and sends control data based on the TUIO protocol

- TUIO client QC plugin: receives and interprets TUIO messages



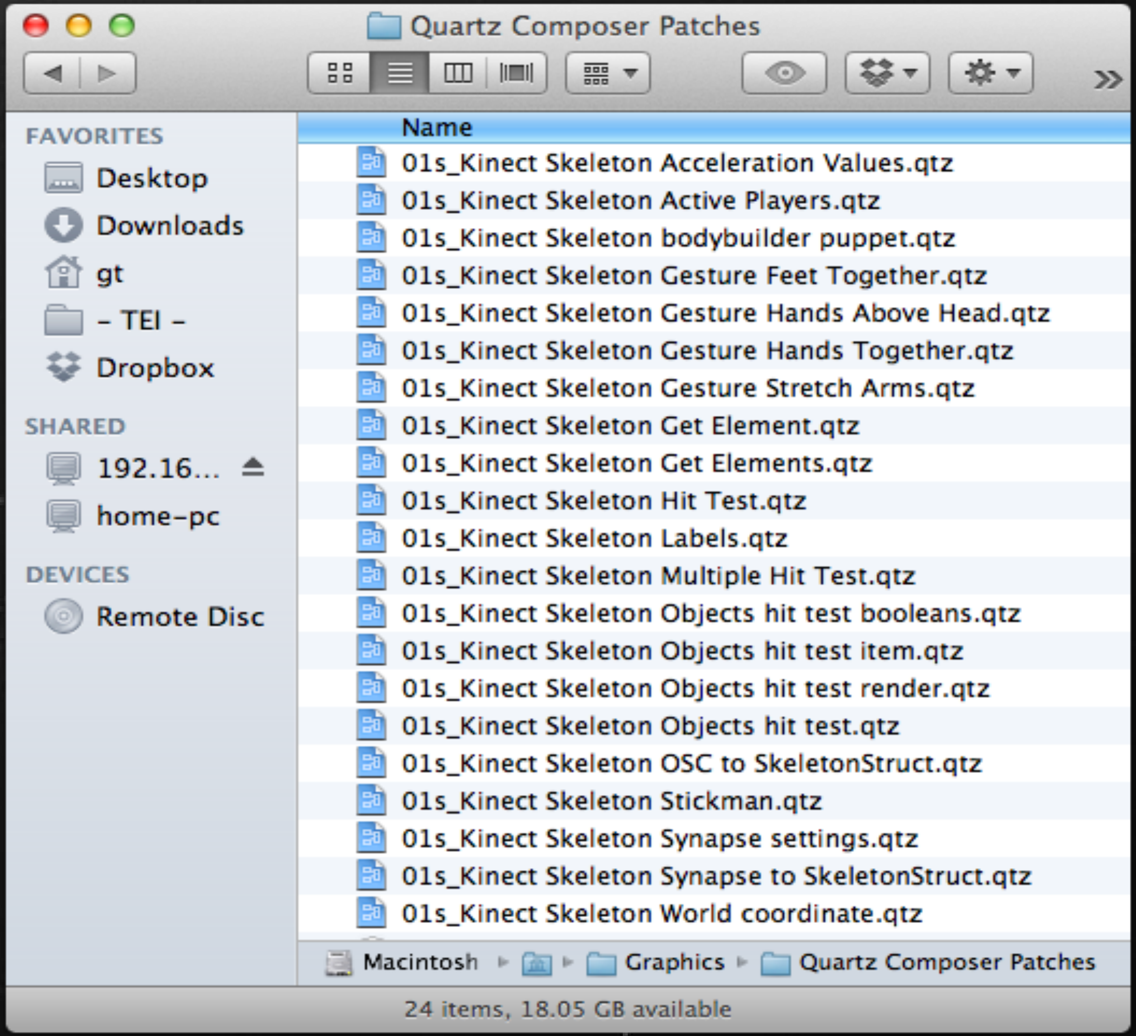
Kinect & QC: TUIO Kinect with TUIO client QC Plugin Example

http://www.youtube.com/watch?v=5camqn9blzc

Kinect & QC: Tryplex Toolkit

- Set of macro patches for Quartz Composer that makes Kinect skeleton tracking more accessible.
- Open source.
- Examples: a puppet-tool and a skeleton recorder.
- Use of Synapse and qcOSC plugins, to show the depth image and track a skeleton.

Kinect & QC: Tryplex Toolkit Patches



Kinect & QC: Tryplex Toolkit Example (skeleton)

http://www.youtube.com/watch?v=q_dyUT20DUo

Kinect & QC: Tryplex Toolkit Example (puppet)

http://www.youtube.com/watch?v=hDtsoSdi5JM

Processing programming language

- Open source programming language (http://processing.org/)

- Initially developed to serve as a software sketchbook and as educational tool

- Used by visual designers and artists

- Create images, animations, and interactions.

- Written in Java. Run as Java programs. For GNU/Linux, Mac OS X, and Windows

- Other approaches: openFrameworks or Cinder

Kinect libraries for Processing

SimpleOpenNI Library for Processing
http://code.google.com/p/simple-openni/

OpenKinect Library for Processing (by D. Shiffman)
http://www.shiffman.net/p5/kinect/

OpenKinect Library for Processing (by P. King-nrocy)
https://github.com/nrocy/processing-openkinect

dLibs_freenect Library for Processing (only for Windows)
http://thomasdiewald.at/processing/libraries/dLibs_freenect/

OpenKinect Library for Processing

1. enableRGB(boolean) — turn on or off the RGB camera image
2. enableIR(boolean) — turn on or off the IR camera image
3. enableDepth(boolean) — turn on or off the depth tracking
4. processDepthImage(boolean) — turn on or off the depth image processing
5. PImage getVideoImage() — grab the RGB or IR video image
6. PImage getDepthImage() — grab the grayscale depth map image
7. int[] getRawDepth() — grab the raw depth data
8. tilt(float) — adjust the camera angle (between 0 and 30 degrees)

OpenKinect Library for Processing Examples

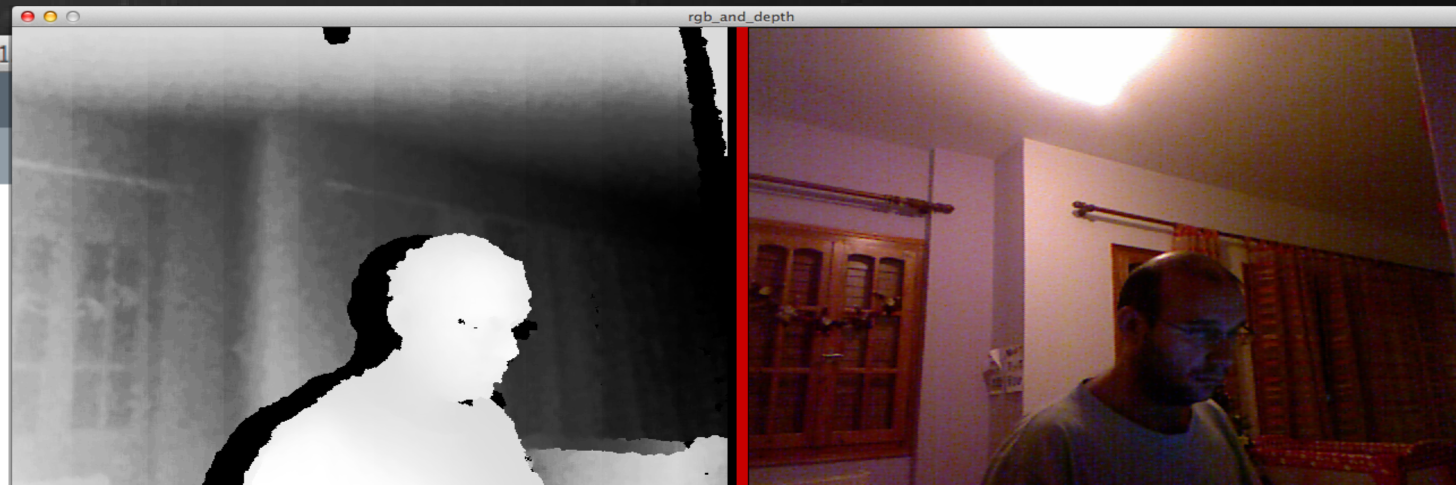
http://www.youtube.com/watch?v=kZBACvkcEsE

SimpleOpenNI Library for Processing

rgb_and_depth | Processing 1

rgb_and_depth

```
import SimpleOpenNI.*;

SimpleOpenNI  context;

void setup()
{
  context = new SimpleOpenNI(this);

  // enable depthMap generation
  context.enableDepth();

  // enable camera image generation
  context.enableRGB();

  background(200,0,0);
  size(context.depthWidth() + context.rgbWidth() + 10, context.rgbHeight());
}

void draw()
{
  // update the cam
  context.update();

  // draw depthImageMap
  image(context.depthImage(),0,0);

  // draw camera
  image(context.rgbImage(),context.depthWidth() + 10,0);
}
```



SimpleOpenNI: Display depthMap and camera

```
import SimpleOpenNI.*;

SimpleOpenNI  context;

void setup()
{
  context = new SimpleOpenNI(this);

  // enable depthMap generation
  context.enableDepth();
```

```
void draw()
{
  // update the cam
  context.update();

  // draw depthImageMap
  image(context.depthImage(),0,0);

  // draw camera
  image(context.rgbImage(),context.depthWidth() + 10,0);
```

SimpleOpenNI Library for Processing Example (hands distance)

http://www.youtube.com/watch?v=BkQsK78xvkU

SimpleOpenNI Library for Processing Example (3d map)

http://www.youtube.com/watch?v=WSqC1g_fTiA

Mesh generation from Kinect Point Cloud using Meshlab

Poisson Reconstruction

http://davidleonardtv.wordpress.com/2011/02/02/turn-kinect-into-3d-scanner-explained-full-tutorial-with-code/