EIDOLON
Manual

Otto Naderer
Ars Electronica Futurelab
otto.naderer@aec.at

February 26, 2014

Abstract

Eidolon is a middleware component that enables the creation of interactive applications that are steered by the user's reflection in a semi-transparent mirror. It therefor calculates how the user would perceive herself in the mirror and passes skeleton data on to some visualization (screen space). Eidolon uses Primesense's NiTE[1] for skeleton tracking.

1 Introduction

Eidolon (*Greek "image, idol, double, apparition, phantom, ghost"*) was developed to enable playful interactions with the mirrored image of a person. The setup consists of a semi-transparent reflective surface that uses either displays or a projection behind/onto this surface to deliver any kind of visualization making use of the data provided by Eidolon.
The component uses an OpenNI-compliant depth-imaging sensor (such as the Asus xTion devices) to detect the user and NiTE to obtain the the user's pose (skeleton). By specifying the mirror's position and dimension, Eidolon can compute the projection of the user's mirrored self with respect to his own perspective. This data is then available for any visualization, already converted in screen space (pixel) coordinates. For that, Eidolon comes with a C++ transmission library that only requires the application developer to implement virtual functions to receive the data and hence abstracts the necessity to implement any network protocol.
In its current form, Eidolon can be compiled on Linux only (Code::Blocks[2]),

[1] primesense.com

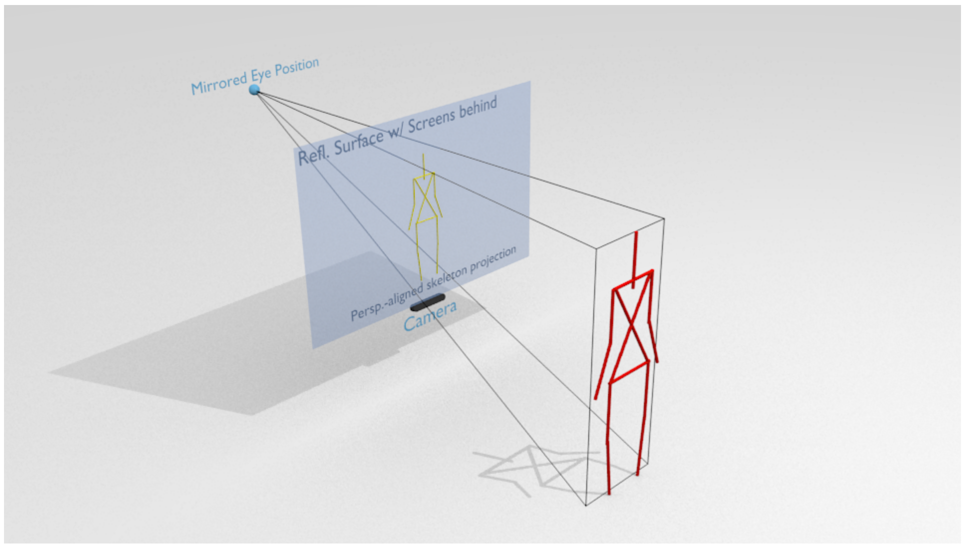

Mirrored Eye Position
Refl. Surface w/ Screens behind
Persp.-aligned skeleton projection
Camera

Figure 1: *Schematic view of Eidolon's intended setup.*

a Microsoft Windows version (including MSVC projects) is on its way. As Eidolon uses cross-platform frameworks only (OpenNI, NiTE, Irrlicht[3] (visualizing system state, geometrical computations)), experienced programmers shouldn't have too much trouble porting it themselves.

2 Setup and Installation

The current version of this document covers the necessary steps for Eidolon to run on GNU/Linux (Ubuntu 12.04 LTS and Arch Linux current). Once Eidolon has been completely ported to other OSs this document will be extended.

2.1 Prerequisites

OpenNI To install the OpenNI SDK, go to the OpenNI download section[4] and get the package suiting your platform (32bit or 64bit). Follow the instructions contained in the compressed file.

[2]codeblocks.org
[3]irrlicht.sourceforge.net
[4]openni.org/openni-sdk

Alternatively, on Arch Linux you can do the following (assuming you have *packer* or *yaourt* installed):

```
packer -S openni2
```

NiTE Primesense's NiTE is a middleware to OpenNI, it sort of acts as a plug-in and provides skeleton tracking. This represents the user's current pose described as 'bone-like' sections. Eidolon will use this data to project the pose onto the mirror plane.
Ubuntu Linux again has no means to directly install NiTE over the package manager so head over to the middleware section[5] and get it there (registration required). Follow the instructions.
Arch Linux features NiTE in its "Arch Linux User Repository (AUR)" so it is available over packer / yaourt:

```
packer -S primesense-nite2
```

Irrlicht Graphics Engine Eidolon uses the Irrlicht engine to render the current state of the application, the positioning of the camera, the interation plane (mirror) and the current pose of the user along with its projection. The engine also provides the functionality for all geometric calculations performed by the application. To install Irrlicht under Ubuntu Linux, type:

```
sudo apt-get install libirrlicht-dev
```

Under Arch Linux, do:

```
sudo pacman -S irrlicht
```

DevEnv Finally we install the CodeBlocks IDE and the GNU Compiler Collection (gcc). To do so, on Ubuntu Linux enter:

```
sudo apt-get install codeblocks gcc g++
```

On Arch Linux, similarly, do:

```
sudo pacman -S install codeblocks gcc
```

We have now installed all prerequisites and are good to go building Eidolon.

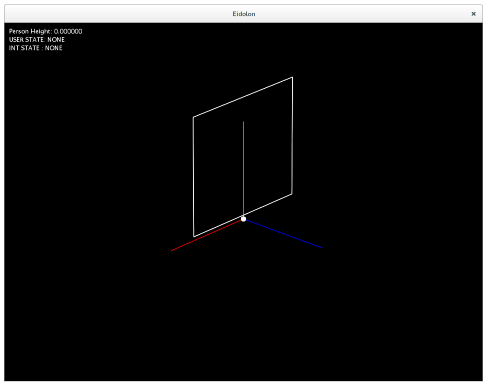
[5]openni.org/files/nite

3 Compiling Eidolon

The framework consists of two separate build projects, one compiles the network library *elTransmissionLib*, the other the main application *elApp*. Both can be opened at once using *Eidolon.workspace* which can be found under */build*. A CodeBlocks workspace resembles MSVC's 'Solution'.
If OpenNI and NiTE are installed in their default locations, there should be no need to customize any paths.
Once the workspace has opened, you see it along with the two projects listed in the left side of CodeBlocks' window. Now just right-click the Workspace and hit *Build workspace*. This automatically starts building the transmission lib and subsequently the application. If for some reason it does not obey the build order, build the lib and the app separately (also by right-clicking the respective project).
Depending on your build target, you should have the following two new files in the folder structure:

```
elApp/bin/elApp
elTransmissionLib/bin/libElTransmission.a
```

If the build target was "Debug", the files will have a 'D' terminating the file name denoting that fact.
Now hit *Run* (greenish 'Play'-Button) or the F9-Key to launch Eidolon. You will be greeted by the following:



```
Person Height: 0.000000
USER STATE: NONE
INT STATE : NONE
```

Figure 2: *Eidolon's main application window.*

Pretty empty, isn't it? It is yet to be filled with interaction and configured. How this is done is described in the next section.

4 Configuring Eidolon

In order to work properly, the application needs to be tought of your current setup. This is what it needs to know:

- *Visualization* – Where and how big your screen / projection is.
- *User Position* – Where to expect the user to stand.
- *Target Machine* – Defines the IP address of the target computer that runs the visualization.

This is it. Now how is this done?

General Assumptions Eidolon's origin of all measurements is the camera position. Everything is measured from where the sensor is placed, in a left-handed coodinate system, z-axis in the camera's view direction, unit: meters. For the novice reader: Take all measurements to the respective positions in meters. If you stand in front of the camera, horizontal measurements are your X values, vertical ones your Y axis and the ones facing towards/away from you the Z value. Just look at fig. 2, the red line is X, the green is Y and the blue one Z. The white sphere is the origin, there's the camera. No worries, this gets easier, just follow on.

4.1 Specifying the Screen

To simplify configuration, Eidolon assumes that the camera sits aligned with the screen/projection/mirror so their depth offset is zero.
In its default configuration, the application has already such a thing defined, this is from now on denoted as *interaction (space)*. Take a look at fig. 2 again, it is drawn as the white rectangle just above the camera.
Now open the configuration file residing in *cfg/config.xml* with an editor of your liking. Find this line there:

```
<interaction ulx="0.69" uly="1.25" lrx="-0.69" lry="0.04"/>
```

You now need to specify the interaction space by filling in the position of the upper-left and lower-right corner of it. Taken the default values, the interaction space's upper-left corner would hence be at (69cm (*ulx*) / 125cm (*uly*)) as measured from the camera, the lower-right corner at (-69cm (*lrx*) / 4cm (*lry*)).
As to be seen in fig. 2 the interaction space hence spans roughly 70cm to the left and right of the camera, is situated slightly above it (4cm) and reaches

1.25m above the sensor.

This document is work-in-progress. This is as far as it gets for the moment. :)