SensoMotoric Instruments

# Eye Tracking HMD Upgrade Package for the Oculus Rift DK2

# Quick Start Guide

28th August 2015
v1.2

# Contents

# 1 Introduction

The *Eye Tracking HMD Upgrade Package for the Oculus Rift DK2* is an Oculus Rift virtual reality (VR) head-mounted display (HMD) with an integrated binocular eyetracker from SensoMotoric Instruments.

The purpose of this document is to give a brief introduction. Along with that, assistance will be offered for creating an application which makes use of eye tracking and thereby giving advice on what data to use for which purpose. This document is not meant as a programming reference (consult `iViewHMDAPI.h` header file), but rather as a best practice quick start.

While most of the explanations below are made with an OpenGL-based application in mind, the principles are equally applicable for any other visualization framework.

# 2 Important Notes

- Read and understand the *Health and Safety Warning* from Oculus Rift.
- Do not leave the Oculus Rift HMD connected to USB and power source when not in use.
- Do not change the distance adjustment as it is fixed. Turning one of the distance adjustment screws on the side of the HMD housing brakes the product.
- Do not exchange lens cups. The product use custom cups, which correspond to Occulus cup set A for moderately nearsighted users or users with normal vision.

# 3 Setting up a new system

1. Download and install the Oculus Windows Runtime 0.5.0 beta from the Oculus Rift website https://developer.oculus.com/downloads/
2. Download and install the latest SMI Eye Tracking HMD installer from the SMI website http://update.smivision.com/iViewNG-HMD.exe
3. Plug the HMD into a power socket using the included power adapter and connect both the HDMI and USB connectors to your computer.
4. The SMI iViewNG HMD SDK supports both, the **Extend Desktop to the HMD** and **Direct HMD Access from Apps** mode for your Oculus. Open the **Oculus Configuration Utility** by right clicking on the Oculus icon in the Windows tray menu and perform the following steps:
   a) If you want to use the Oculus in **Extend Desktop to the HMD** mode:
      - Open in the menu **Tools** > **Rift Display Mode**. Set display mode to **Extend Desktop to the HMD**, apply the changed settings, and close the display mode dialog.
      - Create a new user by using the + button, and click **Advanced**. After reading and understanding the *Health and Safety Warning* section in the quick start guide from the Oculus Rift DK2, enable the checkbox to remove this warning. Confirm the subsequent dialog.
      - Open the **Windows Screen Resolution** configuration dialog.
      - Choose to **extend your desktop** with the Oculus Rift as a second screen and
      - Change the orientation of the Oculus Rift screen to **"Portrait"**. Note that the portrait mode actually looks here like a landscape orientation; this is to be expected.

   b) If you want to use the Oculus in **Direct HMD Access from Apps** mode:
      - Open in the menu **Tools** > **Rift Display Mode**. Set display mode to **Direct HMD Acces from Apps**, apply the changed settings, and close the display mode dialog.
      - Create a new user by using the + button, and click **Advanced**. After reading and understanding the *Health and Safety Warning* section in the quick start guide from the Oculus Rift DK2, enable the checkbox to remove this warning. Confirm the subsequent dialog.
5. Test your installation with the example program `C:/Program Files/SMI/iViewNG-HMD/iViewNG-HMD-API-Example/bin/iViewNG-HMD-API-Example.exe`. The program shows the timestamp and the 2D gaze coordinates of incoming gaze samples if the system has been setup successfully. For further help consult Section 5. Note that on 64-bit Windows systems the SMI software can be found at `C:/Program Files (x86)/...`.

# 4 Creating a new application

## 4.1 Full-screen stimulus presentation window required

It is important for applications that your stimulus presentation window is displayed in a non-windowed, full-screen mode. For instance, you can achieve this with the OpenGL function `glutFullScreen()`. The following code example shows how to maximize and position the window.

```
#include <OVR_CAPI.h>
#include <GL/glut.h>
[...]
ovr_Initialize();
ovrHmd hmd = ovrHmd_Create(0);
[...]
glutInit(...); // adapt this according to your application
glutInitWindowSize(hmd->Resolution.w, hmd->Resolution.h);
glutCreateWindow("My HMD application");
[...]
HWND myWindow = FindWindow(NULL, L"My HMD application");
ShowWindow(myWindow, SW_SHOWMAXIMIZED);
if (hmd->HmdCaps & ovrHmdCap_ExtendDesktop) {
  SetWindowPos(myWindow, HWND_TOP, hmd->WindowsPos.x, hmd->WindowsPos.y,
    hmd->Resolution.w, hmd->Resolution.h, SWP_SHOWWINDOW);
  glutFullScreen();
}
[...]
if (hmd) {
  ovrHmd_Destroy(hmd);
}
ovr_Shutdown();
[...]
```

Listing 1: Example code for the creation of a maximized stimulus presentation window

## 4.2 Gaze interaction: 2D or 3D gaze

The most important step of creating a new application is to decide in what format to process gaze input. The decision boils down to the question of 2D vs 3D: whereas 2D is easier to get started with, 3D data opens the most possibilities for more advanced applications. In the following sections both gaze input data types are explained in detail and some typical use cases are given. Obviously the data types are not limited to those applications.

### 4.2.1 2D gaze input: point of regard (POR)

#### Details

- POR is the gaze vector mapped onto a plane at a specific location within the eyetracker coordinate system.
- The mapping distance can be controlled via the `TrackingParameter` option in `iV_StartStreaming()`
  - A value of `DBL_MAX` will cause the POR to be mapped dynamically to the vergence distance (which is essentially the intersection of left and right gaze rays).
  - An absolute value will result in the POR being mapped to this specific distance.
  - Generally, setting a fixed mapping distance results in better accuracy and precision.
- Monocular PORs for left/right eyes and binocluar POR are mapped to the complete resolution of the HMD screen (1920 x 1080). Realized via virtual camera with a field of view of 87°.

#### When to use POR?

- Monocular POR can be used for interaction applications where the 2D gazes of individual eyes are of importance.
- Binocular POR is a robust entry into simple 2D gaze interaction. It can also be used for reconstructing gaze vectors (e.g. using ray casting) and therefore avoiding transformations.
- Use a fixed mapping distance if you can, e.g. when a scene has only little depth.

#### How to use POR?

- To use the 2D POR coordinates, which are given in pixels, it might be necessary to transform them into relative screen coordinates.
- The following code example can be used, e.g. in combination with the Oculus SDK, to transform pixels (with an origin at the top-left corner of the display screen) into OpenGL coordinates (with an origin at the center of the display). `planeDepth` is the distance in z-direction between the projection plane and the coplanar display.

```
std::pair<double,double> pixelToScreen(double xpixel, double ypixel, double
    height, double width, double planeDepth)
{
  double xgl,ygl = 0.0f;
  xgl = planeDepth * (xpixel/(width/2.0)) -planeDepth;
  ygl = planeDepth  - planeDepth * (ypixel/(height/2.0));
  return std::make_pair(xgl, ygl);
}
```

Listing 2: Transformation of pixels into OpenGL coordinates

### 4.2.2 3D gaze input

#### Details

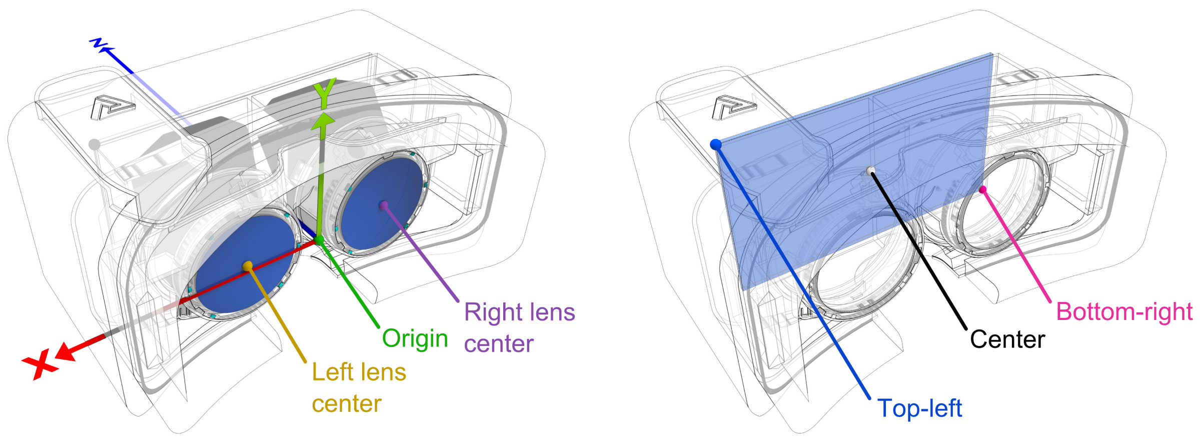
- 3D gaze is provided as a normalized gaze direction vector (`gazeDirection`) and a gaze base point (`gazeBasePoint`).
- Gaze base point is given in mm with respect to the origin of the eyetracker coordinate system.
- "Cyclops" gaze is the averaged gaze direction of both eyes, originating at the virtual center "cyclops" eye of the user.

#### When to use 3D gaze?

- Monocular gaze vectors can be used for vergence research and depth-based selection of objects. The quality of the vergence information varies highly between users and depends highly on ocular motor performance of the wearer. Vergence information works best for viewing distances below 1.5 m.
- Cyclops gaze is optimal in a virtual 3D environment for selecting objects, which the user is gazing at. For instance, select an object where the object's mesh intersects with the ray of the cyclops gaze.
- When displaying the user's gaze it is best to start the ray at around 1 m away from the user in order to avoid discomfort.

#### How to use 3D gaze?

- Using the 3D gaze requires a transformation of vectors from the eyetracker coordinates into user coordinates in the respective scenario.
- As a shared reference between HMD coordinates and eyetracker coordinates Figure 1 and Table 1 define landmarks for the HMD given in eyetracker coordinates.




(a) Origin and lens landmarks

(b) Display landmarks

Figure 1: HMD landmarks in eyetracker coordinates

| Point | Coordinates (x, y, z) [mm] |
|---|---|
| Origin | (0.00, 0.00, 0.00) |
| Left lens center | (31.75, 2.70, 0.24) |
| Right lens center | (-31.75, 2.70, 0.24) |
| Display top-left | (63.00, 39.20, 49.00) |
| Display center | (0.00, 3.70, 49.00) |
| Display bottom-right | (-63.00, -31.80, 49.00) |

Table 1: Coordinates for HMD landmarks in eyetracker coordinates

## 4.3 Display calibration: personal 3D

When a user's physiognomy diverges from made assumptions, a correct projection of a virtual scene to a stereoscopic scene is not possible. Effects can range from a mismatch of object sizes and proportions to unnatural distortions of objects, especially during movement, which may cause nausea and discomfort.

The adaption of the relevant parameters to the user's physiognomy is known as display calibration and gains relevance when presenting close objects. In order to get the best possible results, the virtual cameras (responsible for the left and right eye projections) will have to be placed at the 3D positions of the user's pupils, that is at the position where the light enters the eyes. Keep in mind that it is possible to achieve good results by setting only some of parameters discussed described below while ignoring others.

### 4.3.1 Interocular distance (IOD)

**Details**

- IOD is the distance between the user's left and right eye.
- Can be calculated from the streamed `gazeBasePoint`.
- Constant for each user, independent from `gazeDirection` of each eye.

**When to use IOD**

- For roughly correct proportions and size of virtual objects.
- Approximate replacement for IPD.

**How to use IOD**

- Can be set e.g. in the *Oculus Configuration Utility* or via the Oculus SDK.
- Set as stereo basis for virtual cameras.

### 4.3.2 Interpupillary distance (IPD)

**Details**

- IPD is the distance between the pupil center points of the left and right eye.
- Can be calculated from the streamed `pupilPosition`.
- Changes with gaze direction (gaze angles) and especially vergence.

**When to use IPD**

- Correct position for the virtual cameras.
- Dynamic setting of stereo basis with changing vergence.

**How to use IPD**

- Set as stereo basis for virtual cameras.

#### 4.3.3 gazeBasePoint, pupilPosition, eyeLensDistance, and eyeScreenDistance

**Details**

- `gazeBasePoint` is roughly comparable to the eyeball center.
- `gazeBasePoint` should be used in combination with `eyeLensDistance` and `eyeScreenDistance` in order to position the virtual cameras.
- `pupilPosition` must not be used in combination with `eyeLensDistance` and `eyeScreenDistance`, since they are based on `gazeBasePoint`. Corresponding values would have to be calculated on the fly (check the HMD coordinate system).

**When to use**

- While a good approximation, eye distance is not enough for true stereoscopic projection. For this, 3D positions and orientations have to be set independently for each eye.
- Complete display calibration for user's physiognomy is required in order to achieve best possible rendering quality.

**How to use**

- Place virtual cameras at `gazeBasePoint` or `pupilPosition` (e.g. by transforming the Oculus Rift projection matrix) and orient them using `gazeDirection`. Alternatively use `eyeLensDistance` and `eyeScreenDistance` in combination with IOD or IPD.

### 4.4 User calibration

**Details**

- The 3-point user calibration is the preferred method, however, 0-, 1-, 5- and 9-point user calibrations are available as well.
- Calibration points have to be accepted. Accept them automatically while fixating them, or manually by pressing `space bar`.
- User calibration stays intact after moving the HMD thanks to built-in drift correction.
- Calibration will be visualized by the API and therefore it is essential that the application window and calibration window have the exact same size and resolution (check Listing 1) so they can be aligned correctly.
- User calibration can be saved with a specific name and loaded, to instantly apply the gazecorrection without doing the calibration process.

**User calibration types**

| 0 point | 1 point | 3, 5, 9 point |
|---|---|---|
| • Begins as soon as the eye tracking HMD is put on<br>• Happens automatically<br>• If accuracy is precise enough, you can start immediately | • Improves accuracy over 0 point user calibration<br>• Uses 1 calibration point at 1.5 m distance | • Improves accuracy over 0 and 1 point user calibration<br>• Best calibration compromised between accuracy and speed<br>• Uses calibration points at 1.5 m distance |

**When to use**

- Redo calibration when switching users or when accuracy seems low.

**How to use**

- If necessary, minimize your application before starting the SMI user calibration; due to conflicts with 3rd party applications, the user calibration window cannot be drawn as top-most window
- Launch the calibration (optionally configure it before, e.g. if you want to have a different color set and/or if you want use custom positioned calibration points).
- Each calibration point has to be accepted by fixating on it, this will be automatically detected. In the case that the point is not recognized a manual acception via `space bar` is possible.

## 4.5 Client Side Calibration

### Details

- Create custom calibration visualization
- 1-, 3-, 5-, 9- and 13-point calibration are supported

### When to use

- Calibration in native environment is prefered to the default SMI windowed one

### How to use

- See the code snippet below as an example how to call a client side calibration
- Use the same calibration point order in your visualization as you did for the `smi_CalibrationHMDStruct`
- `smi_calibrate()` is blocking, therefore it's best practice to call your calibration visualization right before in a different thread
- To accept the currently shown calibration point, call `smi_acceptCalibrationPoint()`. The calibration will automatically end all points of your calibration method have been accepted.
- You can also use our fixation detection for your calibration visualization. Therefore call `smi_startDetectingNewF` right before your rendering loop. Then call `smi_smi_checkForNewFixation()` in your loop, and when it's true, call `smi_acceptCalibrationPoint()`. Then call `smi_startDetectingNewFixation()` again.
- To abort the custom calibration, call `smi_abortCalibration()`

```
smi_CalibrationHMDStruct * calibrationHMDStruct;
smi_createCalibrationHMDStruct(&calibrationHMDStruct);
calibrationHMDStruct->type = [...]          // define the type of your struct
calibrationHMDStruct->calibrationPointList[...]   // define the points you
    want to use
calibrationHMDStruct->client_visualization = true;  // set
    client_visualization to true
apiCallRC(smi_setupCalibration(calibrationHMDStruct));
CalibrationThreadHandle = CreateThread(NULL,0,CalibrationThreadFunction,
  NULL,0, &CalibrationThreadID);
apiCallRC(smi_calibrate());
```

Listing 3: Calibration call for client side calibration

```
smi_startDetectingNewFixation (); // initialize the fixation detection
while (runVisualization) {
  if (smi_checkForNewFixation ()) { // check for fixation
    smi_acceptCalibrationPoint ();  // accept calibration point
    smi_startDetectingNewFixation (); // initialize a new fixation detection
  }
  [...]
}
```

Listing 4: Fixation detection calls for client side calibration

## 4.6 Quantitative Validation

The quantitative validation will show a 2x2 grid of points, with only one point shown at a time. As for the calibration, each validation point has to be accepted automatically or manually. After performing the quantitative validation a result window will be shown on both, the user screen (HMD) and the operator screen (this can be turned off programmatically). The result window will show a plot of the fixation per validation point and the average accuracy in degree.

**When to use**

- To measure the accuracy of a calibration.

## 4.7 Recording and evaluation

Currently, recording of gaze data is only possible by doing a client-side file-dump of the streamed data in your preferred format.

### 4.8 Simulator Mode: Use eye tracking API without an HMD device

It is possible to develop and test your eye tracking HMD application using the eye tracking API without having the HMD device attached to your computer. Then the SMI eye tracking API calls your specified callback function at 60 Hz. The exemplary gaze data as shown in Fig. 2 repeats after a while, even though the timestamp is increased further.

The simulator mode can be activated by calling

```
smi_StartStreaming(true);
```

instead of

```
smi_StartStreaming();
```

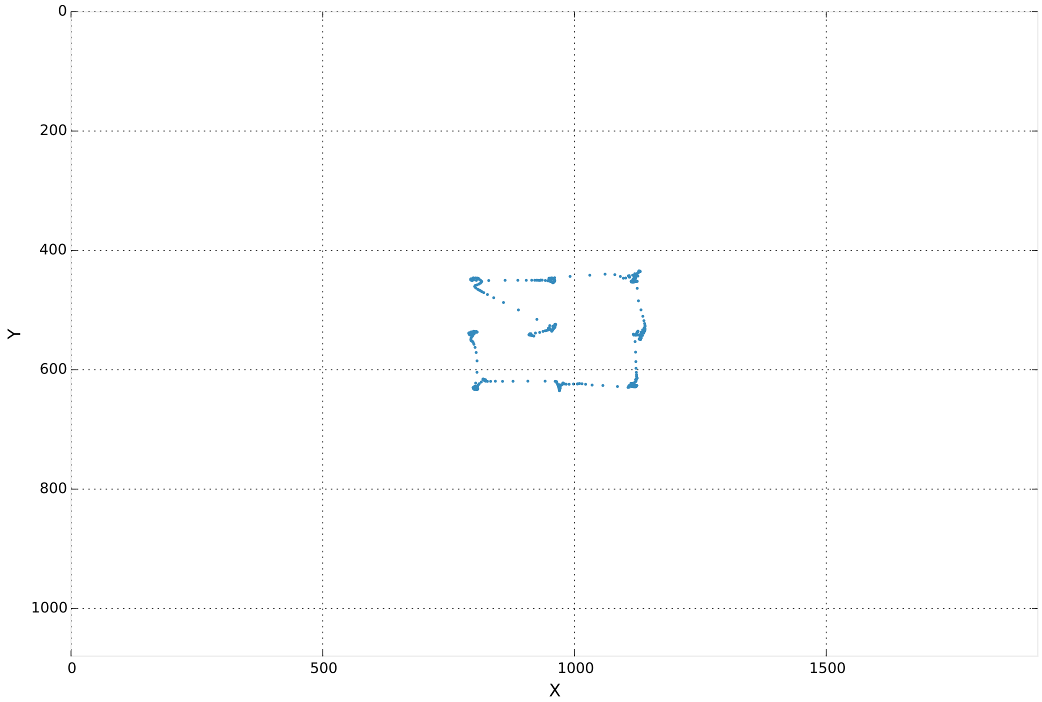



Figure 2: Simulator mode gaze pattern

# 5 Example program for C/C++ SDK API

The *SMI Eye Tracking HMD* provides a pre-built example (`iViewNG-HMD-API-Example.exe`), which can be used to test the device. It is located in the directory:

`C:/Program Files/SMI/iViewNG-HMD/iViewNG-HMD-API-Example/bin/`

Note that the SMI software can be found on 64-bit Windows systems at `C:/Program Files (x86)/...`.

After starting the example, the console output of gaze data shows that the eye tracking is up and running; otherwise error messages will be shown. Note that the console window has to have the input focus in order to control the program via keyboard inputs given in Table 2.

By default, the *SMI Eye Tracking HMD* employs 0-point user calibration for gaze computation. The recommended 3-point user calibration can be started by pressing 3. The calibration points can be accepted automatically or manually. To accept the point automatically, fixate on the point for a short period of time. To accept them manually, press the `space bar`. In this case, please make sure that the user of the HMD is fixating on the current point, e.g. the user signals if point gets fixated.

The new user calibration can be tested in the validation mode, which is started by pressing `v`. A validation pattern (grid of points) will be shown including the POR (2D gaze).

| Keys | Functionality |
|---|---|
| `q` | Quit the program |
| `1` | Start 1-point user calibration |
| `3` | Start 3-point user calibration |
| `5` | Start 5-point user calibration |
| `9` | Start 9-point user calibration |
| `n` | Reset current user calibration to the default 0-point user calibration |
| `h` | Save the current calibration |
| `j` | List all saved calibrations |
| `k` | Load a previously saved calibration |
| `e` | Show/hide eye images |
| `v` | Show validation pattern and POR (2D gaze) overlay |
| `b` | Show quantitative validation pattern and computes the average accuracy |
| `s` | Start eye image streaming: Callback gets called with result types `SMI_EYE_IMAGE_LEFT` and `SMI_EYE_IMAGE_RIGHT` with 5 Hz per eye |
| `u` | Stop eye image streaming |
| `Esc` | Finish current action like user calibration or validation |

Table 2: Keys used in the iViewNG-HMD-API-Example.exe

# 6 Unity integration and example

Unity is a cross-platform game creation tool by Unity Technologies, including a 3D/2D game engine and an integrated development environment. With Unity you can create various applications like games, simulations, or training software.

The SMI *Eye Tracking HMD Upgrade Package for the Oculus Rift DK2* offers an integeration for Unity.

## 6.1 Precompiled Unity integration example

The precompiled Unity integration example can be found in the directory
`C:\Program Files\SMI\iViewNG-HMD\HMD Unity Example\bin`

Different eye tracking user calibration modes are available, see Table 3 for available keys.

| Keys | Functionality |
|---|---|
| `n` | Reset current user calibration to the default 0-point user calibration |
| `1` | Start 1-point user calibration |
| `3` | Start 3-point user calibration |
| `5` | Start 5-point user calibration |
| `9` | Start 9-point user calibration |
| `h` | Save the calibration |
| `j` | List available calibrations |
| `k` | Load Calibration |
| `v` | Show validation pattern and POR (2D gaze) overlay |
| `b` | Start quantitative validation |
| `Esc` | Terminate program |
| `ctrl` | Open the Oculus menu |

Table 3: Keys used in the precompiled Unity integration

## 6.2 Setup

1. If you are using the Oculus in **Extend Desktop to the HMD** mode open the Windows Screen Resolution control panel. Change the order of displays so that the DK2 display is the most right display. Further on, the DK2 display has to be aligned to the top.
2. Import the `SMI_UnityHMD_Integration.unitypackage` (located at `C:\Program Files\SMI\iViewNG-HMD\HMD Unity Example`) into your Unity project. The Oculus SDK version 0.6.0 is already integrated. You can either drag&drop our unity package into your project view or right-click on `Asset`, choose `Import Package` and find our package. Afterwards you have to confirm all the parts being imported.
3. Change Unity settings in the playersettings to API compatibility level ".Net 2.0"

Note: On 64-bit Windows systems the SMI software can be found at `C:/Program Files (x86)/...`.

## 6.3 Notes

- All coordinates are calculated in the classic screen space with [0,0] being top left
- Mapping distance is fixed to 1500 mm

## 6.4 Compatibility

- Use the 32bit editor for Unity 5.0 or Unity 5.1

# 7 General comments and troubleshooting

## 7.1 Usage

- Do not leave the Oculus Rift HMD connected to USB and power source when not in use.
- If using the Oculus in **Extend Desktop to the HMD** use the Oculus Rift only in Portrait mode. Note that the portrait mode of an Oculus Rift display is visualized in the Windows Screen Resolution dialog as a landscape mode.
- Always connect the Oculus Rift to an external power source.
- Do not change the resolution of the Oculus Rift display. Use the native display resolution of the Oculus Rift: 1920 x 1080
- Lenses of the Ouclus Rift have to be handled with absolute care. For cleaning, use a soft damp cloth only.

## 7.2 Update the SDK

SDK updates for the Eye Tracking HMD Upgrade Package for the Oculus Rift DK2 will be made available at http://update.smivision.com/iViewNG-HMD.exe.

For your convenience you can use the Windows Start Menu entry **SMI > Eye Tracking HMD > Check for Updates** to check if a newer version is available. If a newer version is available it will be downloaded and installed.

## 7.3 Error codes

| Error code | Explanation |
|---|---|
| ERROR_NO_CALLBACK_SET | No callback function set. Set a callback function with smi_SetCallback(...). |
| ERROR_CONNECTING_TO_HMD | Unable to find the Oculus HMD or to start the eye tracking server, check the Task Manager and if still running terminate iViewNG-Server.exe |
| ERROR_HMD_NOT_SUPPORTED | HMD device could not be recognized. Download and install an update from http://update.smivision.com/iViewNG-HMD.exe |
| ERROR_INVALID_PARAMETER | Parameter passed to the API is invalid. |
| ERROR_EYECAMERAS_NOT_AVAILABLE | The SMI eye tracking cameras are not available. |
| ERROR_NO_SMI_HARDWARE | The SMI eye tracking system is not available. |
| ERROR_OCULUS_RUNTIME_NOT_SUPPORTED | The Oculus runtime you are using is not supported. Please use 0.5/0.6 runtime. |
| ERROR_FILE_NOT_FOUND | The calibration file could not be found. |
| ERROR_FILE_EMPTY | The calibration file you want to load is empty. |
| ERROR_SDK_NOT_INSTALLED | The SMI HMD SDK is not installed, and our functionality can not be used. |
| ERROR_UNKNOWN | Unknown error occurred. Close program and disconnect the device. Kill iViewNG-Server.exe using the Task Manager. Reconnect the device. |

Table 4: API error codes

## 7.4 Common Problems

**Health and Safety Warning shows up again**
Go to Oculus Configuration Utility, uncheck the *Disable Health and Safety Warning* checkbox. After applying your changes repeat the procedure and check the checkbox again.

**Head mounted display (HMD) flipped vertically on Windows 8.1 systems**
Open the Windows Screen Resolution configuration dialog and change the “Portrait” mode to the “Portrait (flipped)” mode.

**Main Screen is turning black when starting programm**
This might occur when Windows Aero is enabled or the Windows DPI is set above 100%. Disable Windows Aero design or set Windows DPI to 100% as a workaround.

**Oculus Service Crashes**
This might occur randomly when using the Direct To Rift mode and closing an application. Since our calibration visualizer is an own application aswell, it might call this behaviour. Make sure to use a graphical adapter which is recommended by Oculus when you want to use the Direct to Rift mode, otherwise choose Extended Mode.

**HMD is working, but eye tracking isn't.**
The eye tracker uses the USB accessory port on the DK2 headset and hence requires the power supply which has to be connected to the connector box. Please be aware that the order of connecting the cables to the DK2's connector box has to be as follows:

1. Connect the HDMI cable.
2. Connect the USB cable from the connector box to one of the USB ports on the computer.
3. (optional) Positional tracker sync cable: Connect the positional tracker with the cable connector box via the sync cable.
4. (optional) Positional tracker USB cable: Connect the positional tracker with the mini USB cable to one of the USB ports on the computer.
5. Finally, plug the barrel connector of the power cord to the cable connector box.

Retry eye tracking functionality after connecting all cables according to the above sequence of steps.

**Cannot execute example program directly from Visual Studio**
Open the Visual Studio project configration for the example program. Select as configuration “All Configurations” and platform “Active(Win32)”. Go to Configuration Properties > Debugging > Working Directory and set it to `$(OutDir)`.

**Could not open codec file, make sure its located in the same directory**
Although the Oculus Rift DK2 is connected to the system, the execution of the example program `iViewNG-HMD-API-Example.exe` results in the error message

```
smi_StartStreaming:
Could not open codec file, make sure its located in the same directory
```

Solutions:

1. Run `iViewNG-HMD-API-Example.exe` as administrator if executed from a directory with restricted access rights (e.g. `C:\...`), because the program has to copy files to the working directory of the example program.

2. Copy the whole `bin` directory of the example program to another drive where write access is granted and execute `iViewNG-HMD-API-Example.exe` from there.

**Unity 5.1: UnityEditor.PlayerSettings does not contain a definition for 'd3d11FocreExclusiveMode'**
Unlike Unity 5.0, Unity 5.1 does not contain a definition of `d3d11ForceExclusiveMode` as a `PlayerSetting`, depending on which Oculus SDK you are using. Go into the source file and comment the line `PlayerSettings.d3d11ForceExclusiveMode = OVRUnityVersionChecker.hasD3D11ExclusiveModeSupport;`.

# 8 Declaration of Conformity

SMI products are for use in office environments and bear the CE mark to indicate compliance with the health and safety requirements according to European Directives. For individual product declarations please refer to sales@smivision.de.

The equipment has been tested and found to comply with the limits for class A digital devices, pursuant to EMC directive 2004/108/EC, and conforms to the low-voltage directive 2006/95/EC.

The device is meant for use in office environments and as such may generate radio frequency emissions that can cause interference with radio communications. To reduce the potential impact on radio communications, the operator may have to take measures such as increasing the separation between such devices or changing the orientation of the equipment.

The SMI Eye Tracking HMD Upgrade Package for the Oculus Rift DK2 is eye safe according to EN62471:2008.

# 9 3rd Party Licenses

## 9.1 Boost Software License

`http://boost.org/LICENSE_1_0.txt`



## 9.2 GNU Lesser General Public License

**2. Conveying Modified Versions.**
If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:
a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs `whatever`part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

**3. Object Code Incorporating Material from Library Header Files.**
The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

(a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

(b) Accompany the object code with a copy of the GNU GPL and this license document.

**4. Combined Works.**
You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

(a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

(b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

(c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

(d) Do one of the following:

   1) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

   2) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

(e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

**5. Combined Libraries.**
You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

**6. Revised Versions of the GNU Lesser General Public License.**
The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.
If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

## 9.3 GCC Runtime Library Exception

Version 3.1, 31 March 2009

Copyright © 2009 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

**0. Definitions.**
A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and use would permit combination with GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor architecture, in executable form or suitable for input to an assembler, loader, linker and/or execution phase. Notwithstanding that, Target Code does not include data in any format that is used as a compiler intermediate representation, or used for producing a compiler intermediate representation.

The "Compilation Process" transforms code entirely represented in non-intermediate languages designed for human-written code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use of source code generators and preprocessors need not be considered part of the Compilation Process, since the Compilation Process can be understood as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-compatible software, or if it is done without using any work based on GCC. For example, using non-GPL-compatible Software to optimize any GCC intermediate representations would not qualify as an Eligible Compilation Process.

**1. Grant of Additional Permission.**
You have permission to propagate a work of Target Code formed by combining the Runtime Library with Independent Modules, even if such propagation would otherwise violate the terms of GPLv3, provided that all Target Code was generated by Eligible Compilation Processes. You may then convey such a combination under terms of your choice, consistent with the licensing of the Independent Modules.

**2. No Weakening of GCC Copyleft.**
The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements of the license of GCC.

## 9.4 Freeglut License

Freeglut code without an explicit copyright is covered by the following copyright:

Copyright (c) 1999-2000 Pawel W. Olszta. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies or substantial portions of the Software.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL PAWEL W. OLSZTA BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Pawel W. Olszta shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Pawel W. Olszta.

## 9.5 Mozilla Public License Version 2.0

**1. Definitions**

**1.1. “Contributor”** means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

**1.2. “Contributor Version”** means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor’s Contribution.

**1.3. “Contribution”** means Covered Software of a particular Contributor.

**1.4. “Covered Software”** means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

**1.5. “Incompatible With Secondary Licenses”** means

a. that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

b. that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

**1.6. “Executable Form”** means any form of the work other than Source Code Form.

**1.7. “Larger Work”** means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

**1.8. “License”** means this document.

**1.9. “Licensable”** means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

**1.10. “Modifications”** means any of the following:

a. any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software;
or

b. any new file in Source Code Form that contains any Covered Software.

**1.11. “Patent Claims” of a Contributor** means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having made, import, or transfer of either its Contributions or its Contributor Version.

**1.12. “Secondary License”** means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

**1.13. “Source Code Form”** means the form of the work preferred for making modifications.

**1.14. “You” (or “Your”)** means an individual or a legal entity exercising rights under this License. For legal entities, “You” includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50

## 2. License Grants and Conditions

### 2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

a. under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work;
and

b. under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

### 2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

### 2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

a. for any code that a Contributor has removed from Covered Software;
   or

b. for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version);
   or

c. under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

### 2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

### 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

### 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

### 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

1. such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

2. You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

## 4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

## 5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

### 6. Disclaimer of Warranty

Covered Software is provided under this License on an “as is” basis, without warranty of any kind, either expressed, implied, or statutory, including, without limitation, warranties that the Covered Software is free of defects, merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Covered Software is with You. Should any Covered Software prove defective in any respect, You (not any Contributor) assume the cost of any necessary servicing, repair, or correction. This disclaimer of warranty constitutes an essential part of this License. No use of any Covered Software is authorized under this License except under this disclaimer.

### 7. Limitation of Liability

Under no circumstances and under no legal theory, whether tort (including negligence), contract, or otherwise, shall any Contributor, or anyone who distributes Covered Software as permitted above, be liable to You for any direct, indirect, special, incidental, or consequential damages of any character including, without limitation, damages for lost profits, loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses, even if such party shall have been informed of the possibility of such damages. This limitation of liability shall not apply to liability for death or personal injury resulting from such party’s negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

### 8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party’s ability to bring cross-claims or counter-claims.

### 9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

### 10. Versions of the License

#### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

#### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

#### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

### Exhibit A - Source Code Form License Notice

```
This Source Code Form is subject to the
terms of the Mozilla Public License, v. 2.0.
If a copy of the MPL was not distributed with this file,
You can obtain one at http://mozilla.org/MPL/2.0/.
```

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

### Exhibit B - "Incompatible With Secondary Licenses" Notice

```
This Source Code Form is ''Incompatible With Secondary Licenses'',
as defined by the Mozilla Public License, v. 2.0.
```

## 9.6 BSD License

opensource.org/licenses/bsd-license.php

License Agreement

For Open Source Computer Vision Library
Copyright (C) 2000-2008, Intel Corporation, all rights reserved.
Copyright (C) 2008-2010, Willow Garage Inc., all rights reserved.
Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistribution's of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution's in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the copyright holders may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## 9.7 zLib License

www.grinninglizard.com/tinyxmldocs/index.html

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

SensoMotoric Instruments Gesellschaft für innovative Sensorik mbH
Warthestraße 21
D-14513 Teltow
Germany

Phone +49 (3328) 3955 10
Fax +49 (3328) 3955 99
URL http://www.smivision.com
EMail info@smivision.com

Document number: 150109-P-2314-001-000-A