1 About This Document

This document is written to summarise the Most Important Things (e.g. installation procedure) related to the software developed for the gaze tracking glasses. Feel free to update it if you find a bug or change something in the code.

2 Included Libraries and Software

2.1 Libraries

GazeTracker
This is the gaze tracking library

pattern_finder
Part of the gaze tracking library – will be merged to the GazeTracker ⋯ some day

jpeg
Used for uncompressing MJPG data

ResultParser
Parses results from an input stream

QVideo
Used for streaming video streams using OpenCV. This library is used by LEDCalibration

VideoControl
Used for streaming videos from multiple sources. This library uses directly GStreamer

2.2 Software

Following is a description of the software components available. The actual gaze tracker software is named "gazetoworld", under the *src/Ganzheit/TwoCameraTracker/gazetoworld* directory.

LEDCalibration
Used for calibrating the LED positions and camera

LedCalibration/tests/calib_viewer/samples
Used for viewing the LED positions in a 3D world

LedCalibration/calibrate_LED_images/calibrate
Refine the collected LED samples

LedCalibration/calibrate_flipped_samples/calibrate
Read in the samples, apply vertical flip and calibrate

starburst_searcher
Used for testing the 2D pipeline

gazetoworld
This is the actual gaze tracker software. It tracks the gaze vector and draws the point of gaze in the scene camera image. This software supports storing the results in a file.

GazeTracker_simulator
Draws a simulated eye and determines the gaze vector for it. **Deprecated.**

3 Prerequisities

It is advisable to create folder *src/* to your home folder. Most of the software will be installed to this folder.

Download and install following software (the software is known to, but not restricted to, work with the version number in brackets):

- Install OpenCV (2.3.1)
  1. Download the source code from `http://code.opencv.org/` and extract the source to */usr/local/src/OpenCV-2.3.1/*
  2. Install the debug libraries to folder */usr/local/src/OpenCV-2.3.1/build/debug* and the release libraries to */usr/local/src/OpenCV-2.3.1/build/release/*
  3. consider running *Documents/opencv_prerequisities.sh*
  4. Activate the debug libraries by running:

     ```
     sudo sh -c 'echo "/usr/local/src/OpenCV-2.3.1/build/debug/lib" >
     /etc/ld.so.conf.d/opencv.conf' && sudo ldconfig
     ```

     or run

     ```
     export LD_LIBRARY_PATH=$(LD_LIBRARY_PATH)
     :/usr/local/src/OpenCV-<version>/build/debug/lib
     ```

- Eigen (3.X.XX)
  1. Download the source code from `http://eigen.tuxfamily.org/` and extract the source to *yourHomeFolder/src/Eigen3/*. After extracting the source code *Eigen3* folder should contain *Eigen* folder.

- gdb
    1. *sudo aptitude install gdb*
- Gstreamer (0.10)
    1. *sudo aptitude install libgstreamer0.10-dev*
- GNU Scientific Library (1.15)
    1. If you are running on Ubuntu or LMDE, this software should be found from the repository. Simply run: *sudo aptitude install libgsl0-dev libgsl0-dbg gsl-bin*
- TinyXml (2.6.2)
    1. Download the source code from `http://sourceforge.net/projects/tinyxml/` and extract the source to *yourHomeFolder/src/tinyxml/*.
- SDL (1.2)
    1. run: *sudo aptitude install libsdl1.2-dev*
- GLEW
    1. run: *sudo aptitude install libglew-dev*
- guvcview
    1. This software is not mandatory, but extremely useful. It allows setting the camera parameters (exposure, focus, etc.).
    2. If you are running Ubuntu or LMDE, this software should be found from the repository. Simply run: *sudo aptitude install guvcview*

4 Obtaining the Source Code and Compiling

Retrieve the source code from the github repository (`https://github.com/bwrc/gaze_tracker_glasses`).

Providing that you have installed the OpenCV and all other libraries according to these instructions, a software can be compiled by running *make* in the software directory. Compiling LedCalibration requires running *qmake CONFIG+="debug/release"* before running make.

While compiling the software, it is advisable to run make with *-j4* option. This allows compiling multiple files at the same time.

5 Hacking UVC Driver

Microsoft Lifecam Cinema requests always the whole bandwidth of the USB controller. The issue can be solved either by hacking the driver or by using a computer with two USB controllers. The latter requires just finding a computer with two USB hosts. However, most laptops have just one USB controller and hence hacking the driver may be necessary.

WARNING Hacking the driver will affect all cameras using the driver. It is possible that the cameras cannot stream unpacked data anymore or they just do not work. If you experience these issues, remove the hack from the code and reinstall the driver.

Applying the Hack

1. Clone first linuxtv repository:

   ```
   git clone git://git.linuxtv.org/media_build.git
   ```

2. Enter the *media_build* folder and compile the driver (this is the easiest way to get the source code downloaded):

   ```
   ./build
   ```

3. Open file *media_build/linux/drivers/media/video/uvc/uvc_video.c* and search the following section from the code

   ```
   int intfnum = stream->intfnum;

   /* Isochronous endpoint, select the alternate setting. */
   bandwidth = stream->ctrl.dwMaxPayloadTransferSize;

   if (bandwidth == 0) {
   ```

   and modify the code to:

   ```
   int intfnum = stream->intfnum;

   /* Isochronous endpoint, select the alternate setting. */
   bandwidth = 1024; //stream->ctrl.dwMaxPayloadTransferSize;

   if (bandwidth == 0) {
   ```

4. Change directory to *media_build/* and run:

   ```
   make -j4 && sudo make install
   ```