

Unofficial Touchlib Reference

By Laurence Muller

Original source:

Multi-touch displays: design, applications and performance evaluation. (Master's Thesis)

Last updated: 20th June 2008

www.multigesture.net

Touchlib Reference

Currently no official Touchlib documentation is available. Only a few configuration examples are available in the SVN tree. This appendix provides general information about configuring and using Touchlib.

B.1 Project details

Project site: www.touchlib.com

SVN: <http://code.google.com/p/touchlib/>

SVN (repository): <http://touchlib.googlecode.com/svn/multitouch/>

B.2 Touchlib config.xml

The configuration file consist out of the following parts:

1. XML version definition

example: `<?xml version="1.0" ?>`

2. Tracker configuration

example:

```
<blobconfig distanceThreshold="250" minDimension="2" maxDimension="250"
ghostFrames="0" minDisplacementThreshold="2.000000" />
```

Tolerance settings of the blobtracker. The distanceThreshold contains the value of how many pixels a blob can travel. The minDimension and maxDimension variables specify how small or large a contour can be to be detected as a touch. The value of ghostFrames specifies the number of extra frames Touchlib should use for the blobtracker. The minDisplacementThreshold specifies how many pixels a contour needs to be moved before calling the update event.

3. Bounding box

example:

```
<bbox ulX="0.000000" ulY="0.000000" lrX="1.000000" lrY="1.000000" />
```

Specifies in which region blob detection should be applied.

4. Screen calibration points

example: `<screen>...[points]...</screen>`

These values will be filled when running the configapp.

5. The filtergraph

example: `<filtergraph>...[filters]...</filtergraph>`

Filters are explained in the next section.

B.3 Touchlib filters

General note: All filters need to be in between of the filtergraph tags.

`< filtergraph > ... < /filtergraph >`

The first filter in the filtergraph needs to be a video capture filter, the last one must be the rectify filter.

B.3.1 Video capture filters

cvcapture

Description:

This is the default video capture filter used by Touchlib. It uses OpenCV function to capture a video stream. It is also possible to use a video file for testing purposes.

Usage (Video file):

```
<cvcapture label="cvcapture" >
<source value="../tests/rear4.avi" />
</cvcapture>
```

Usage (Camera):

```
<cvcapture label="cvcapture" >
<source value="cam" />
</cvcapture>
```

dsvlcapture

Description:

This is an implementation which uses DirectShow to capture a video stream.

Usage:

```
<dsvlcapture label="dsvlcapture" />
```

cmucapture

Description:

This filter uses the CMU driver to access FireWire video capture devices.

Usage:

```
<cmucapture label="cmucapture" >
<brightness value="-1" />
<exposure value="-1" />
<flipRGB value="false" />
<gain value="87" />
<gamma value="1" />
<mode value="640x480mono" />
<rate value="30fps" />
<saturation value="90" />
<sharpness value="80" />
<whitebalanceH value="0" />
<whitebalanceL value="-1" />
</cmucapture>
```

vwcapture

Description:

This filter uses the VideoWrapper API to access FireWire video capture devices.

Usage:

```
<vwcapture label="capture1" >
<videostring value="pgr: 0 640 30 grey16 1 rgb" />
</vwcapture>
```

The videostring contain the camera parameters which depends on the interface type:

Cameras using DCAM specifications from Videre Design

Parameters: "dcam: camNum width frameRate colorMode scale"

Example: "dcam: 0 640 30 rgb 2"

Cameras from Point Grey Research

Parameters: "pgr: camNum width frameRate colorMode scale outputMode"

Example: "pgr: 0 640 30 grey8 1 rgb"

Cameras using VidCapture (DirectShow)

Parameters: "vc: camNum width frameRate colorMode scale outputMode"

Example: "vc: 0 640 15 rgb 0"

B.3.2 Video processing filters

Mono filter

Description:

Touchlib requires an 8 bit grey scale source image. This filter is only required if the video capture filter is not capable of delivering the right image format.

Usage:

```
<mono label="monofilter" />
```

Background Filter

Description:

This filter removes the background by creating a reference image on initialization and subtracting it from the current active frame.

Usage:

```
<backgroundremove label="backgroundfilter" >  
<threshold value="20" />  
</backgroundremove>
```

The threshold value has a range from 0-255.

Smoothing Filter

Description:

The smoothing filter applies a gaussian blur on the source image.

Usage:

```
<smooth label="smoothfilter" />
```

Invert Filter

Description:

The filter inverts a greyscale image. This filter is only required for FI.

Usage:

```
<invert label="invert" />
```

Scaler Filter

Description:

If the previous used filter gives a weak output the scaler filter is used to amplify the current image.

Usage:

```
<scaler label="scaler" >  
<level value="70" />  
</scaler>
```

Brightness and Contrast Filter

Description:

In particular situation it might be required to adjust the image brightness and contrast. This filter is only used for FTIR

Usage:

```
<brightnesscontrast label="brightnesscontrast4" >  
<brightness value="0.1" />  
<contrast value="0.4" />  
</brightnesscontrast>
```

Highpass Filter

Description:

When using RI or FI, the brightness of the blobs is much weaker compared to FTIR. However if enough contrast is available the HighpassFilter is capable of amplifying the output of these blobs.

Usage:

```
<highpass label="highpass">  
<filter value="6" />  
<scale value="32" />  
</highpass>
```

Highpass Filter (Simple)

Description:

Same purpose as the previous filter, however it uses a simpler algorithm and therefore performance faster than the default HighpassFilter. The filter contains two different methods to amplify the source images, currently it is recommended to set the *noiseMethod* to 1.

Usage:

```
<simplehighpass label="simplehighpass">  
<blur value="13" />  
<noise value="3" />  
<noiseMethod value="1" />  
</simplehighpass>
```

Barrel Distortion Correction Filter

Description:

A wide-angle lens often contains a radial distortion which can not be corrected by Touchlib default correcting algorithm. This filter corrects the barrel distortion based on the lens characteristics. The filter requires a camera.yml file which is create by the barrel distortion correction tool. Because the image get corrected, the result might be missing out important parts. By setting the border size this can be corrected (if not needed, set this to 0).

```
<barreldistortioncorrection label="barreldistortioncorrection1">  
<border_size value="20" />  
</barreldistortioncorrection>
```

Crop Filter

Description:

Allows the user crop the video image.

The values posX and posY define the top left position of the crop area. The height and width define the crop area size.

```
<crop label="crop">  
<posX value="40" />  
<posY value="40" />  
<height value="120" />  
<width value="160" />  
</crop>
```

Rectify Filter

Description:

This is the final filter of the image processing pipeline. The threshold is set to a value in which blobs are visible but noise input is ignored. This image will be used to perform blob detection on.

Usage:

```
<rectify label="rectify">  
<level value="75" />
```

</rectify>

The level value has a range from 0-255.

B.4 Touchlib calibration

In order to calibrate the Touchlib it is required to have a fully functioning multi-touch table (this includes a camera and projector). It is recommended to look at the example xml files to create a config.xml configuration file. Every multi-touch technique requires a different filter chain. When the filter chain has been set up, the configuration application can be used by starting configapp.exe from the Touchlib bin directory (or ./configapp from the /src directory under linux). It now displays the filters you have entered in the config.xml file. Adjust the sliders in the filters until the rectify filter only shows clear blobs when being touched. If you need to recapture the background press 'b'.

When you are satisfied with the result of the filter chain you can start the actual calibration. Press 'enter' to go into full screen mode. The config application shows a black background with 20 green colored reference points, 5 horizontal and 4 vertical. In the upper left corner you will see an example of the camera input. If you do not want to use the full size of your camera input it is possible to adjust the bounding box by pressing 'x'. Please read the instruction on the screen to adjust this box. To start the calibration press 'c'. The current point to select is red, including green arrows rotating around the spot. If your camera is orientated differently (such as flipped on the Y-axis) Touchlib will auto correct this on the first touch. Continue touching the required points. When the calibration is finished you can test the accuracy by touching the surface. It shows a rectangle shaped to the size of the 'press'. To exit the configuration press 'ESC'. When the application quits all values are written to the config.xml. If you need to adjust the calibration, you can adjust the values inside the <screen>...</screen> tag.

B.5 TouchData structure

The range of the X and Y are from 0 to 1 (floating point).

Data type	Variable name	Properties
integer	ID	Unique identifier blob
integer	tagID	Unique identifier fiducial
float	X	Horizontal position
float	Y	Vertical position
float	height	Hight of fiducial
float	width	Width of fiducial
float	angle	Rotation of fiducial
float	area	Size of blob (pressure)
float	dX	Delta movement horizontal axis
float	dY	Delta movement vertical axis

Table B.1: TouchData structure
