TacTile Software Documentation

Version: May 19, 2009

http://www.evl.uic.edu/cavern/tactile/

Introduction

TacTile is a High-definition Multi-touch LCD display designed for the visual exploration of scientific datasets. The system can be built using off-the-shelf components. TacTile uses a LCD display as opposed to a projector, which means a higher picture quality, lower cost, and longer life span. It is greatly suited for both scientific applications as well as museum and informal education applications.

This document gives an overview of the architecture of TacTile and follows this with a practical step-by-step guide to configure and calibrate the system. The software needed to run TacTile can be downloaded from this web page: http://www.evl.uic.edu/cavern/tactile/software.html

You can also download the latest version of this document from the web page above.

I. Architecture

TacTile uses a client-server architecture to provide abstraction to the computer vision software so that multi-touch applications can be developed rapidly without the need to directly access the tracking cameras.

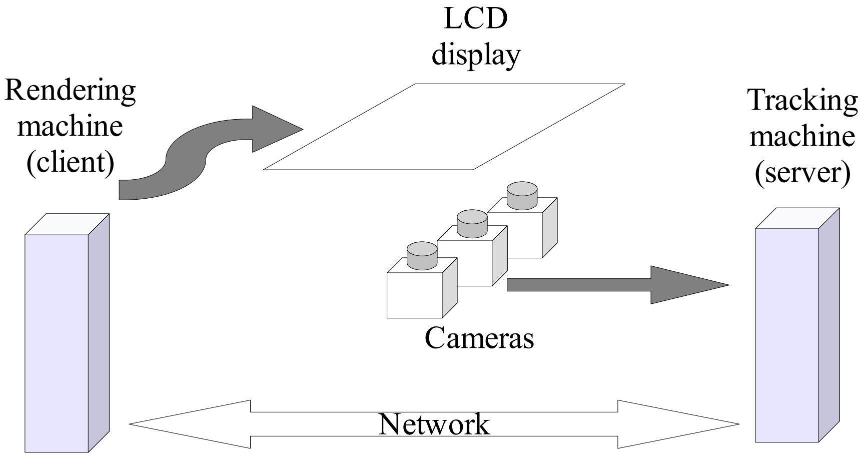


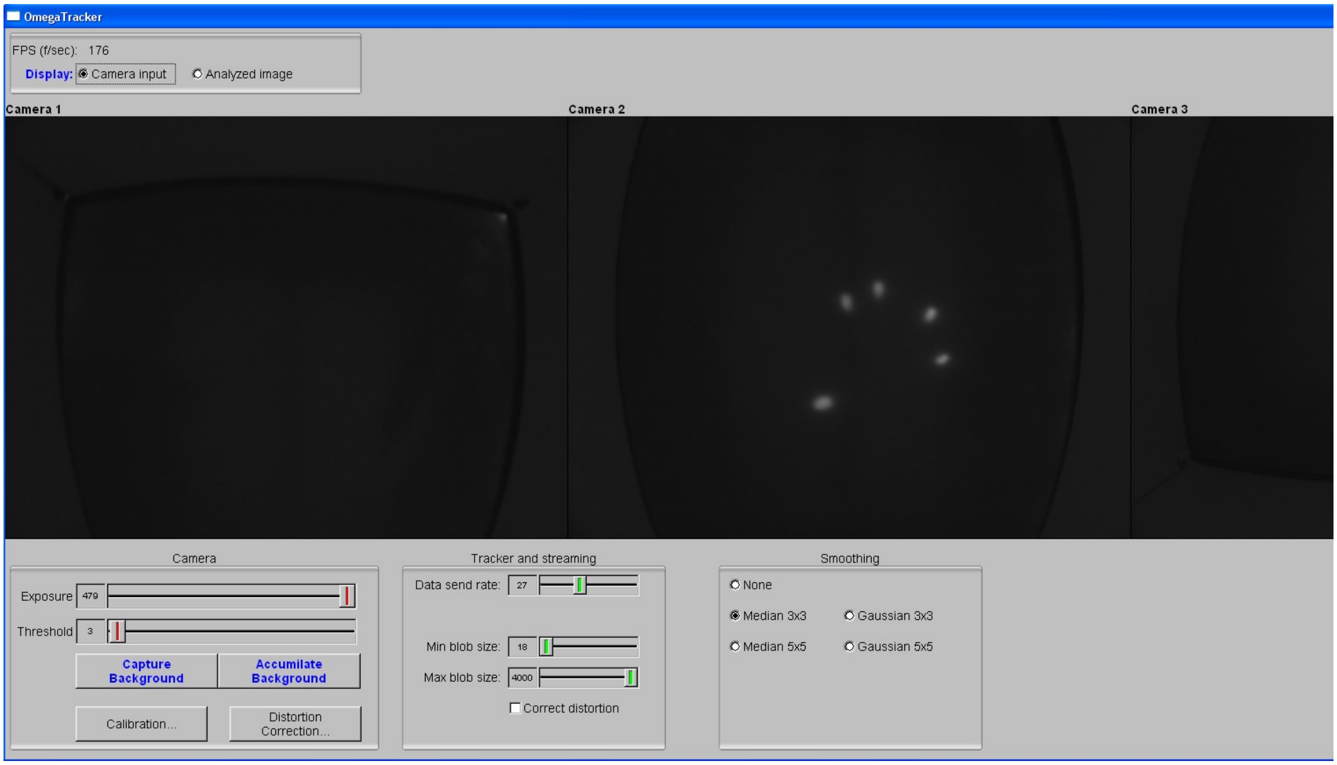
Figure 1: architecture of TacTile

The server program (the tracker program) implements all the computer vision code required to track fingers. Client applications connect to the server program. The server program will stream data containing information about the location and movement of fingers the client application in real-time. The server and client can run on a single machine, or they can be split over two machines. In this case the two machines need to be networked and client applications will need to specify the network address of the server machine.

II. Configuring the Tracker program

The tracker program should be run on the computer connected to the cameras. Currently the tracker runs only under Windows XP 32-bit. To run the tracker, simply double click on run.bat script.

The first time the tracker is launched, it will notify you that the system needs to be calibrated. Follow the steps below to configure and calibrate the system. The tracker window will look like figure 2.


FPS (f/sec): 176

Display: ● Camera input ○ Analyzed image

Exposure: 479

Threshold: 3

Data send rate: 27

Min blob size: 18

Max blob size: 4000

□ Correct distortion

Smoothing: ○ None ● Median 3x3 ○ Gaussian 3x3 ○ Median 5x5 ○ Gaussian 5x5

Figure 2: tracker program

2.1 Modifying camera exposure

The camera exposure affects the sensitivity of the camera. The higher the exposure, the more sensitive the cameras are. To modify the exposure, simply drag the slider labeled 'Exposure'. Normally, the exposure should be set to its maximum. After the exposure is changed, the background image should be refreshed (see next section).

2.2 Refreshing the tracker's background image

The tracker program needs to know what the images from the IR cameras look like when there are no fingers on the display. This is refereed to as the 'background image'. The background image needs to be refreshed every time the tracker is started, or the lighting conditions in the room change. To refresh the background image, simply click the 'Capture Background' button. No one should be touching the display when the background is being captured. The 'Accumulate Background' button behaves similarly, except it that it combines the cold background with the current view. This is useful when a new lighting source is turned on in the environment and the tracker starts registering false positives.

2.3 Enabling/disabling filtering

Filtering helps the tracker discard light from external sources making the tracking more accurate. However, if your system is not fast enough, it might reduce the frame rate. You may select the type of filtering from the 'Smoothing' box on the lower-left corner of the tracking program. The recommended filters are Median 3x3 and Median 5x5 (more aggressive filtering but slower). The frame rate is displayed in the upper-left corner of the tracker program (labeled FPS).

2.4 Modifying the threshold

The threshold affects the sensitivity of the screen to finger touches. The lower the threshold, the higher the sensitivity. Obviously a low threshold is desirable. However, a very low one might make the tracker sensitive to lights from external sources causing it to report fingers touches when there are not any. You should therefore select a threshold that is suitable to your environment. In a dark room, a threshold of 1 to 3 is good. However, in other environments where there are many light sources, you might need to set the threshold to 7-10. The minimum value is 1 and the maximum is 255.

To select and appropriate threshold, use the following procedure:

1. Refresh the tracker's background image by clicking 'Capture Background' button.
2. Switch the camera view to 'Analyzed Image' from the upper-left corner of the program.
3. Set the threshold to a low value like 3.
4. You should see a solid black image in all three cameras. If you see white flickering, then you need to increase the threshold, or enable filtering (section 2.3).
5. Place your fingers on the display. You should be able to see clear outlines of your fingers in white surrounded by red squared (see figure 3).

Keep experimenting with the value of the threshold until you are satisfied with the sensitivity of the display. Remember, a lower threshold means a higher sensitivity. Also note that dragging your fingers across the display will require a lower threshold to register the fingers than simply tapping or pressing with fingers without moving.

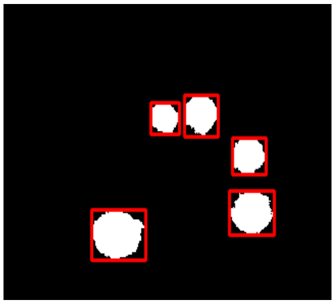


Figure 3: detected finger blobs

2.5 Min/Max Blob size

These values reflect the minimum and maximum size of blob (in pixels) required to register a finger. If your screen is not that sensitive, you can reduce the min blob size to about 15 to 30 pixels. Conversely, if your screen is very sensitive, the blobs generated by the fingers will be larger, and thus you need a higher maximum blob size. You can tell if the tracker is discarding blobs if you see the white blobs in the 'Analyzed Image' but not seeing red outlines around them

2.6 Data send rate

This specifies the speed at which the tracker notifies client applications of finger movement. The recommended range is 20 to 60. The default value of 24 should be adequate for most applications.

III. Calibration

Before applications can be run on TacTile, the system needs to be calibrated. The calibration is done with the help of a utility program which can be downloaded from TacTile's software page (http://www.evl.uic.edu/cavern/tactile/software.html). The calibration utility requires the following prerequisites to run:

- Electro: http://www.evl.uic.edu/rlk/electro
- LuaSocket: http://www.tecgraf.puc-rio.br/~diego/professional/luasocket/

The calibration utility should be run from the computer connected to the display. Before running it, you need to modify its configuration file to let it know the network address of the tracker machine. To do this, open the 'config.lua' file and modify the following line:

```
TRACKER_HOST             = "10.0.8.13"
```

Replace 10.0.8.13 with the network address of the machine that will run the tracker. You may also modify the number of calibration points by editing the following two lines:

```
CALIBRATION_ROWS         = 6
CALIBRATION_COLUMNS      = 12
```

Those specify the number of calibration points in each row/column respectively. A larger number increases the accuracy of finger positions. The default values are sufficient however in most cases.

Before starting the calibration, make sure that 'Correct distortion' in the main tracker window is UNCHECKED. Now you can run the calibration process. To do this, launch the script 'calibration.lua' with Electro. On Linux, you can do this from the terminal window by moving into the directory of the calibration utility and running

```
electro calibration.lua
```

The calibration utility will display a series of dots. Place ONE FINGER only on each of them. When all the points have been touched, the utility program will sound a chime and display all points. At this point the calibration is complete, and you may hit ESC to exit the utility. If you look at the tracking program, you should see a window similar to figure 4. If you see this, then calibration is successful. Otherwise, there is something wrong. You can look at the DOS console that opens with the tracker program and look for any error messages. Also make sure that your machine supports OpenGL with Framebuffer extension.

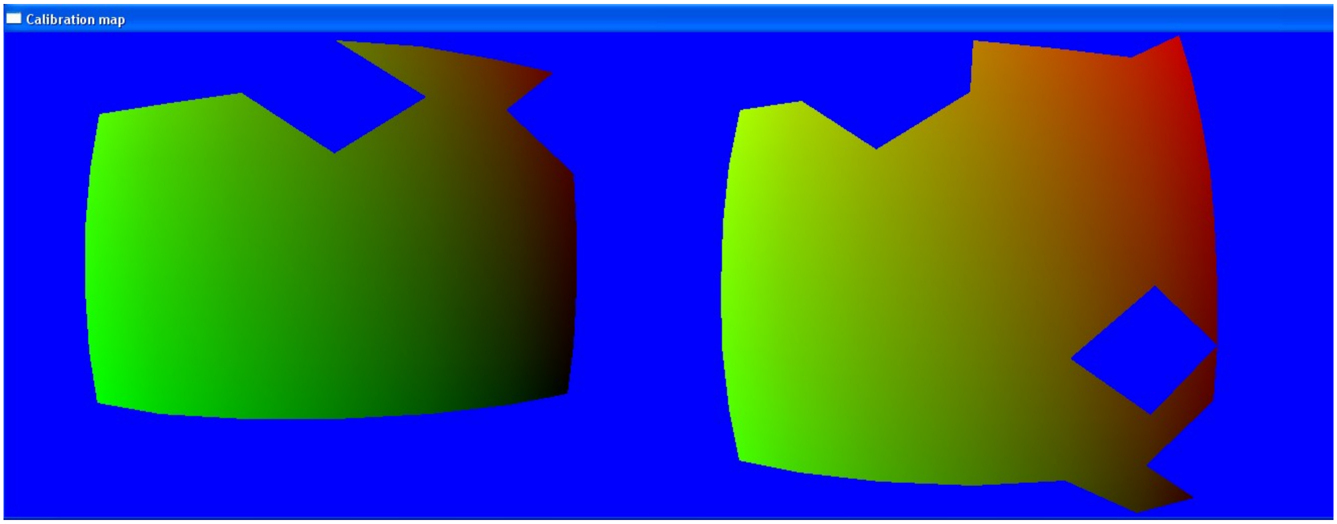


Figure 4

To verify the accuracy of the calibration, you can run another program that comes with the calibration utility called draw.lua. Again, on Linux this is done from the terminal window:

```
electro draw.lua
```

By placing your fingers on the screen, the program will draw circles where your fingers are. If the circles are noticeably off your fingers, try repeating the calibration process. If you get the same result, increase the number of calibration points as described above. ESC exists the draw program.