

Tobii installation

<http://andrewd.ces.clemson.edu/et/ab>

Tobii ;

hardware specs

model: ET-1750

50 Hz

0.5° accuracy (bias error) [Tobii 2003]

1280 x 1024

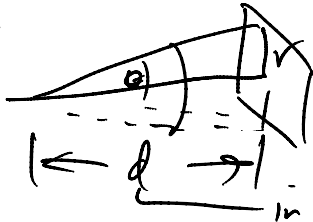
17" diagonal

} what's

← see manual

the visual angle?

expression =



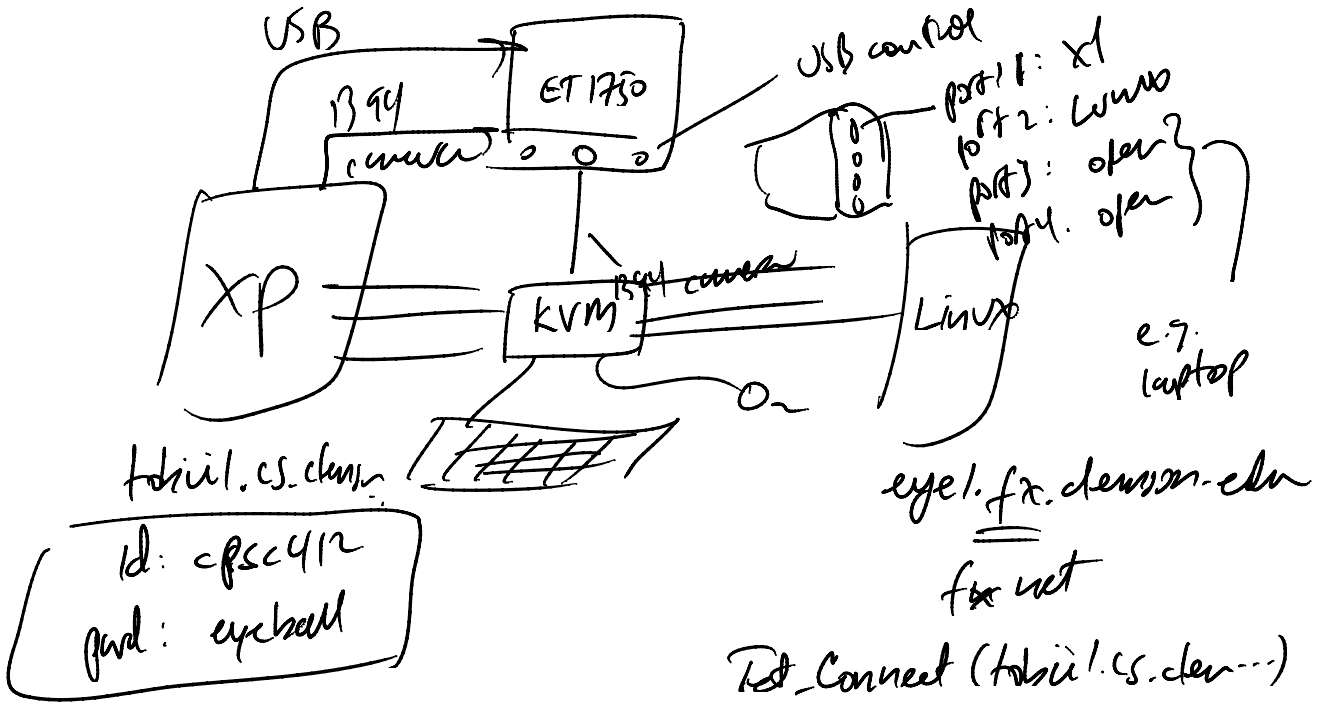
$$\frac{r}{d} = \tan \theta$$

$$2\theta = \tan^{-1} \left(\frac{r}{2d} \right)$$

in in or cm $[50\text{cm}] = 19.7^\circ$

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{r}{2d} \right)$$

HW: confirm

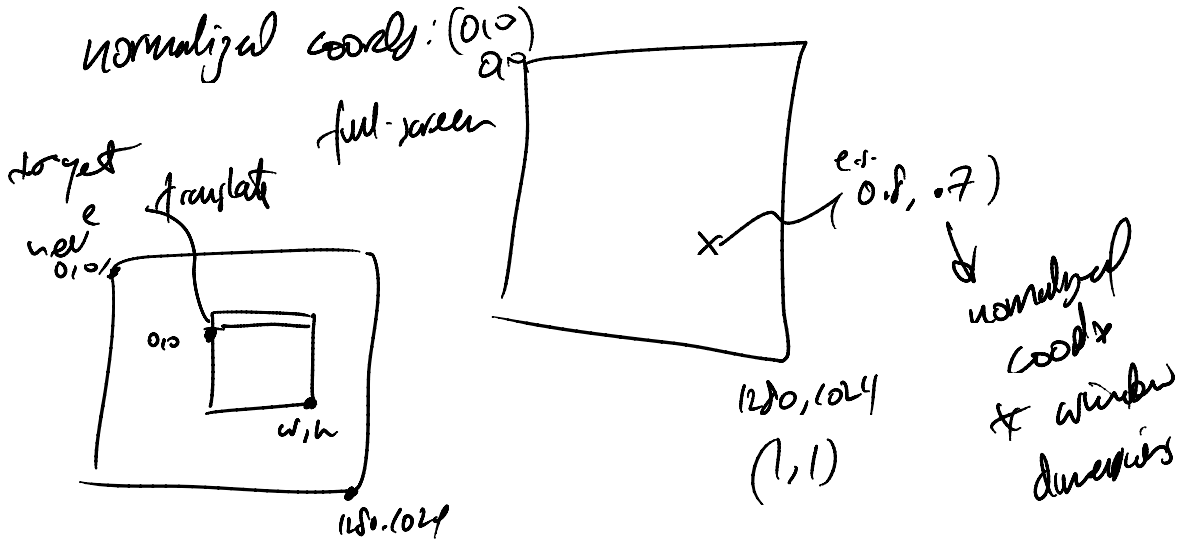


Your client app:

1. Connect (goes thru TCP/IP sockets)
↳ done for you in Ret-Connect
 2. calibrate
 3. synchronize
 4. retrieve data (x_{ij} , y_{ij})
-

2 Key concerns:

1. synchronization
2. x, y coordinate mapping — ET-175°
transmits normalized
coordinates (really
nice)



- Advancements from old hardware (ISCAN):
lot of hardware problems went away
(like unnormalized coords)
- Now, "problems" (sync, mapping) relegated to software:
old way:
eye tracker (ET) sponsored device
sends control codes to app over serial port
(operator presses button to set ET state)
- new way:
app controls ET
app sets ET state (i.e., idle, calib, run)

ET control: all done thru API

Link API:

Test-Init } basic communication
Test-Connect }
" Disconnect } control
Start
Stop

Test-CalibClean } calibration calls
" CalLoadFromFile }
" }
↓
app sends (key)
to ET

Test-SynchronizeTime }
" - PerformSystemCheck } Utilities

Net-Init(void);

- must be called once at ~~ET~~ thread initialization
- only needed in Linux AIZ

Net-Connect(char* pServerAddress, - "tohil.cs.clemson.edu"
 ushort port, - 4455
 char* pLogfile) - log file

- do once, but don't forget to,

Windows if a box,
127.0.0.1

Tet_start (gazeDataReceiver, NULL, 0); ← usage example

void * appData
interval

call this after calibration, to start ET running — this is a blocking call, program control given up at this statement to ET thread — if runs \neq calls gazeDataReceiver()

interval in ms when callback fn is called — with "TET_CALLBACK_TIMER" as the ET_CallbackParam.

C def:

void (*Tet_CallbackFunction)(ET_CallbackParam C, void * data, void * appData);

function pointer.

callback routine

Test_stop (void)

- stop ET (changes mode from run → idle)

- tto gets called when ET is running.

But when ET is running, your ET client is only "accessed" via ET callback function

∴ Test_stop has to be called from within callback routine

Tet. Callback Function (ETet_Callback Person Person,
void* pData,
void* (APIData))

Person: TET_CALLBACK_GAZE_DATA — when running
" " _TIMER or calibrating

my pData = (STet_GazeData) pData;
see tet.h {
 ulong ————— time stamp - sec
 ulong ————— " - micro sec,
 float ————— x-gaze px - left eye } eyeballs
 float ————— y- " - left eye } from camera's
 float ————— x-camera - left eye } view point.
 ulong ;
 ulong (validity-left eye) } } see tet.h
 ... }
 }

Tuesday, August 08, 2006
12:15 PM