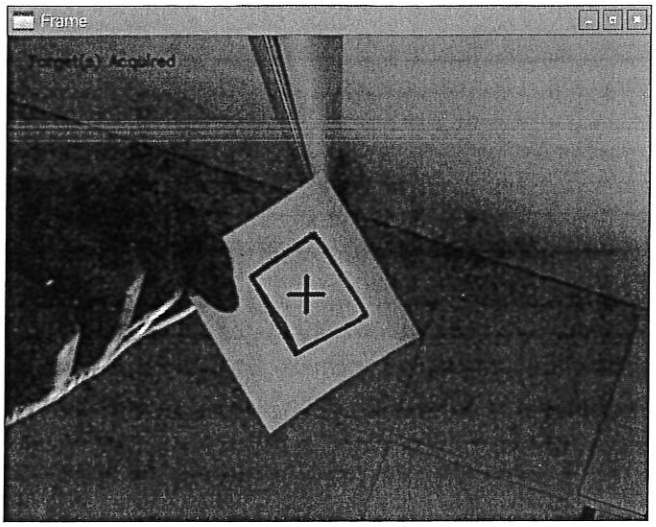
- shape_tracking.py, följer ett mål av en specificerad form, här angiven som en kvadrat (se bild nedan).



- camera_test.py, för att testa vilken framerate som kan fås i Python (beroende av vilken upplösning som används)

Ett demoprogram för färgbaserad målföljning skrivet i C++ finns under /home/pi/multirotor/cpp/color_tracking. Färgvalet görs med att välja ett område på målet med musen (Hue) medans gränser för saturering (S) och värde (V) görs med reglage i GUI:t.

Referenser:

1. https://www.raspberrypi.org
2. https://www.raspbian.org
3. http://opencv.org
4. https://github.com/Itseez/opencv/blob/master/samples/python2/mosse.py samt http://www.cs.colostate.edu/~bolme/publications/Bolme2010Tracking.pdf

Raspberry Pi 2 med tillbehör

Ett utvecklingskit för kompentensutveckling som använts till programutveckling för quadcopter/multirotor projektet för att använda dess kamera till navigering. Det består av följande delar:

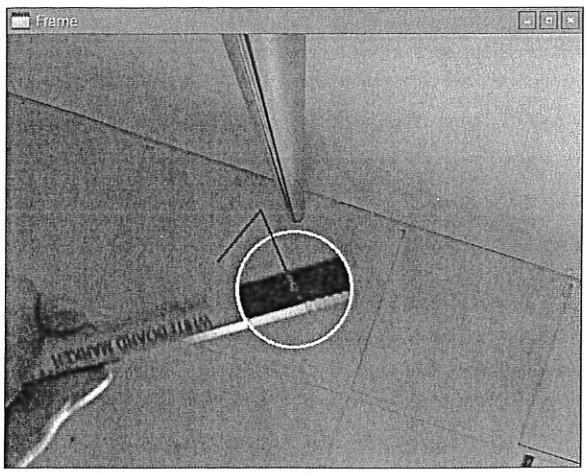
- Raspberry Pi 2B med genomskinlig låda
- Kameramodul
- USB Wifi-dongel, D-Link DW-121 (använts för att koppla upp Pi:n mot AF-Guest wifi)
- 32GB klass 10 mikro SD-kort med adapter
- HDMI-till-DVI adapter
- Till detta behövs också ett USB-tangentbord samt mus och bildskärm med DVI-kabel.

Programvara som installerats på Raspberry Pi[1] är OS:et RASPIAN[2] som är baserat på Linuxdistributionen Debian samt Opencv[3] (version 3.0 med bindning till Python 2.7) som är ett ramverk för bildbehandling.

Första steget har varit att skriva Pythonskript för att testa några enkla typer av målföljning. Då processningen i Python är relativt långsam är nästa steg att skriva om dessa i C++/C för att se vilken framerate som är möjlig att uppnå. Slutligen finns det bättre målföljningsalgoritmer (ex vis MOSSE [4]) som behöver implementeras för att förbättra prestandan.

Pythonskripten finns under /home/pi/multirotor/py och dessa är:

- color_tracking.py, följer ett mål av en specificerad färg som anges som ett intervall i HSV-koordinater (se bild nedan).



- get_hsv_value.py, skriver ut i terminalen, HSV-värdet för den pixel klickas på med musen (för att få fram HSV-värdena som behövs i color_tracking.py).