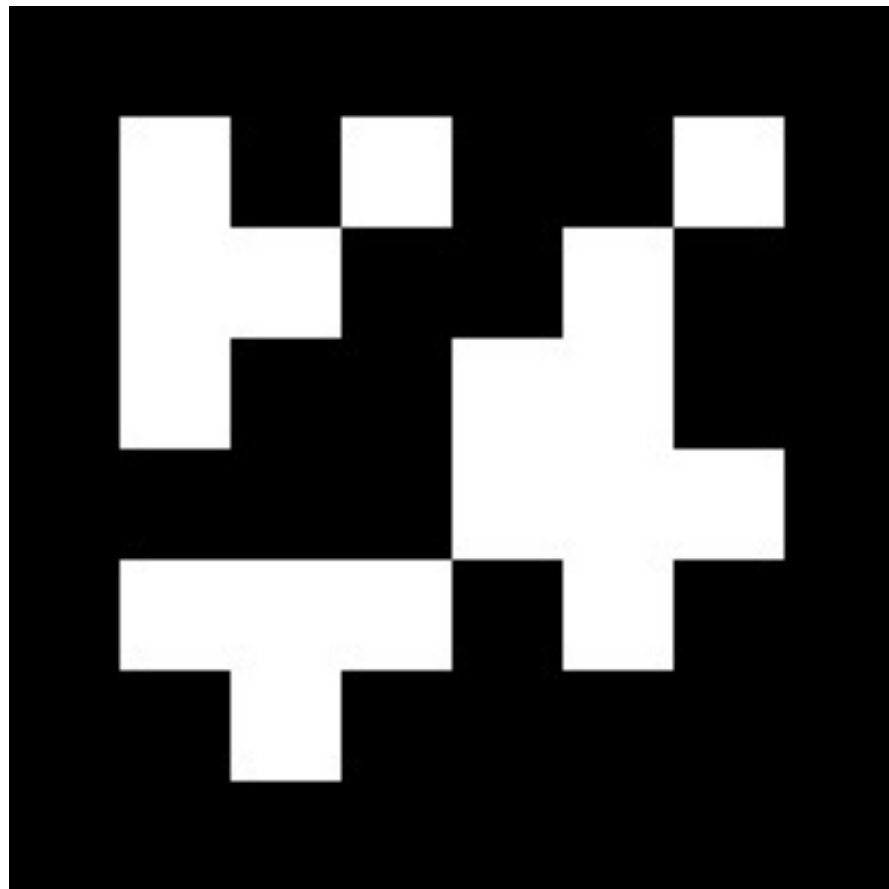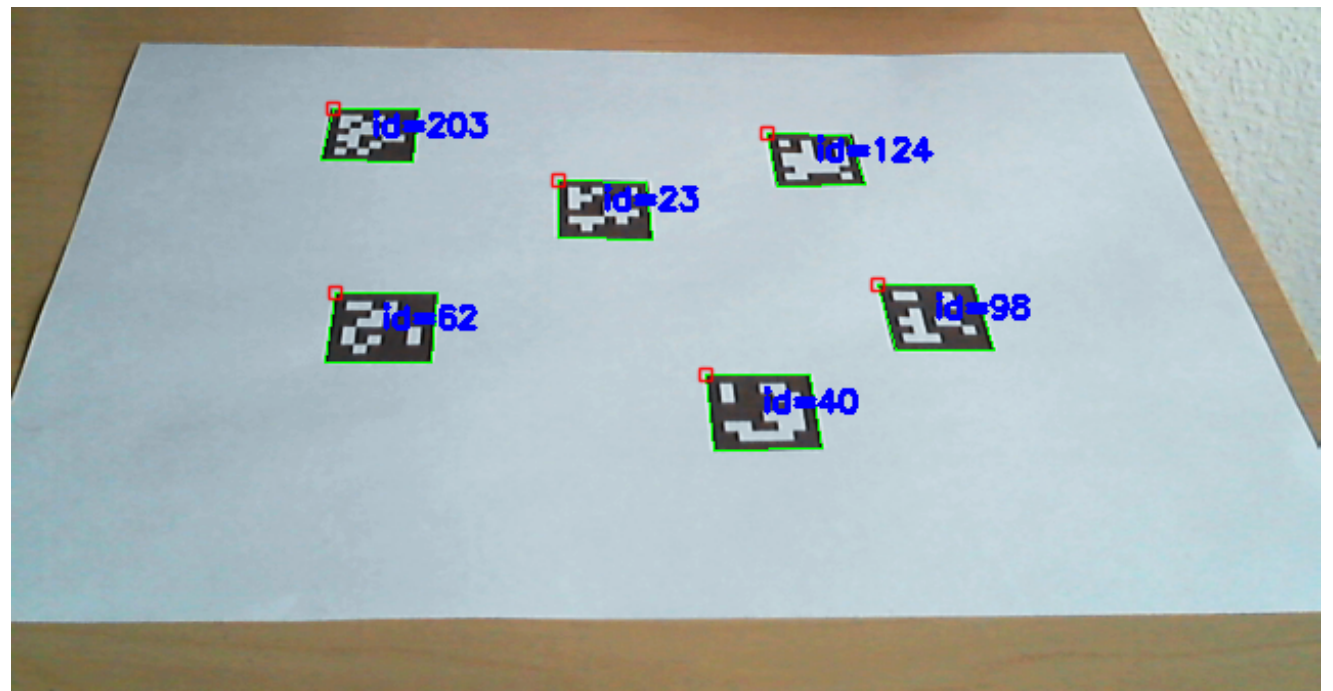# Marker & CVDrone

2018/04/12

1. Marker Detection (50%)

aruco marker

marker detection

1. Marker Detection (50%)

a. calibration

b. marker detection

c. pose estimation

d. controlling

1. Marker Detection (50%)

```
cv::Ptr<cv::aruco::Dictionary> dictionary =
cv::aruco::getPredefinedDictionary(cv::aruco::DICT_6X6_250);
```
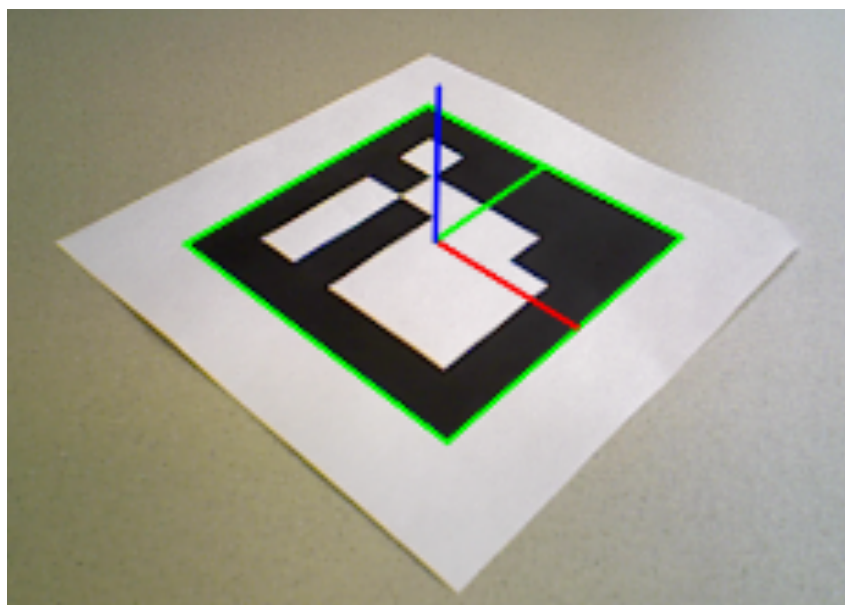
1. Marker Detection (50%)

```cpp
std::vector<int> ids;
std::vector<std::vector<cv::Point2f>> corners;
cv::aruco::detectMarkers(image, dictionary, corners, ids);
```

1. Marker Detection (50%)

```
std::vector<cv::Vec3d> rvecs, tvecs;

cv::aruco::estimatePoseSingleMarkers(corners,
markerLength, cameraMatrix, distCoeffs, rvecs, tvecs);
```

1. Marker Detection (50%)

a. calibrate the drone camera

b. marker detection by drone camera

c. pose estimation

x: 10.3478
y: 21.5618
z: 3.9908

2. CVDrone (50%)

Download CVDrone project

- Link: https://github.com/puku0x/cvdrone

Compile and Run

- for Windows VS:
  Click cvdrone/build/vs2015/test.sln to open project

- for Ubuntu & MacOS:
  $ cd cvdrone/build/unix
  $ make
  $ ./test.a

  Note:
  1. You should install OpenCV and ffmpeg before "make"
  2. You should connect to drone by WiFi before running "test.a"

Install ffmpeg on Ubuntu and MacOS

- for Ubuntu
  sudo add-apt-repository ppa:mc3man/trusty-media
  sudo apt-get update
  sudo apt-get dist-upgrade
  sudo apt-get install ffmpeg

- for MacOS
  brew install ffmpeg

Folder Overview

- **build/**

  test.a (執行檔)

  (for Unix, 修改程式碼後記得"make"再執行)

- **src/**

  main.cpp

  你也可以在這個資料夾中加入其他程式檔

- **samples/**

  e.g. sample_camera_calibration.cpp

  cvdrone 作者提供的一些參考程式檔，

  將你想要測試的程式碼覆蓋到 main.cpp 中即可。

Keyboard control

- Note:
  往後撰寫自動飛行的程式碼時，
  一定也要有 keyboard control 功能，
  且要有最高優先權，確保自動飛行狀況
  不佳時仍能手動控制。

```
"*******************************************"
"*        CV Drone sample program          *"
"*           - How to play -               *"
"*******************************************"
"*                                         *"
"* - Controls -                            *"
"*     'Space' -- Takeoff/Landing          *"
"*     'Up'    -- Move forward             *"
"*     'Down'  -- Move backward            *"
"*     'Left'  -- Turn left                *"
"*     'Right' -- Turn right               *"
"*     'Q'     -- Move upward              *"
"*     'A'     -- Move downward            *"
"*                                         *"
"* - Others -                              *"
"*     'C'     -- Change camera            *"
"*     'Esc'   -- Exit                     *"
"*                                         *"
"*******************************************"
```

ARDrone API

- takeoff();
  landing();
  onGround();

- Mat getImage();

- move3D(double vx, double vy, double vz, double vr);

- setCamera(int mode);