

大兵萊恩 一路直前

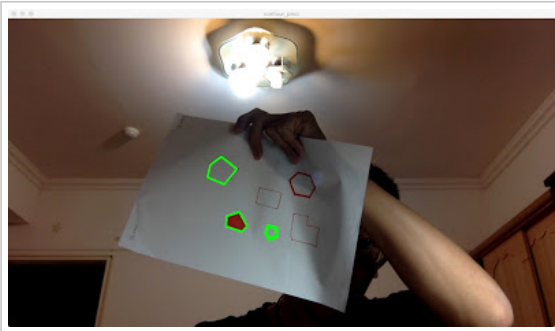
IOT, Digital Signage, Arduino, BLE, RPi, IOS Programming, 拆解玩具, 隨便亂做, 閒聊分享... 很難分類的部落格。

[Home](#)

To search type and hit enter...

[Home](#) » [OpenCV](#) » [Python](#) » 記錄, OpenCV 學習路徑, (2) 辨識多邊形 (OpenCV, Python)

記錄, OpenCV 學習路徑, (2) 辨識多邊形 (OpenCV, Python)

10
Aug
2015

因為想在四軸飛行器上整合影像辨識做些好玩的事, 所以萊恩大兵動念想自學 OpenCV. 胡亂摸索一陣之後, 覺得 OpenCV + Python 的學習資源較豐富, 就往這方向投入了。

看 [Github: PyCV-time](#) 的範例, 有教怎麼辨識多邊形 (polygon), 甚至是箭形 (arrow). 邊看程式, 邊查 API 說明, 倒也學到不少知識. 大致整理一下萊恩大兵的理解。

要辨識多邊形, 首先, 要能找出圖像的 Contour (輪廓)。

[1] 什麼是 Contour (輪廓)?

擷取 [OpenCV Python Tutorials](#) 裡面的定義: Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity.

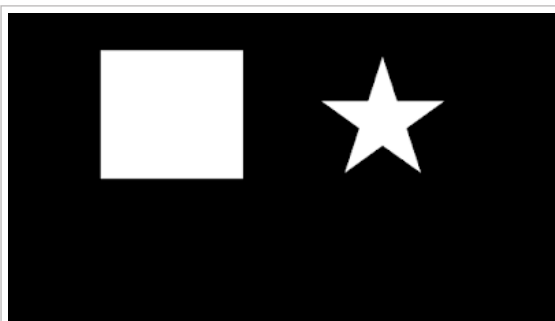
從上述的定義可理解, 一般在對圖像取 contour 前, 都會先轉黑白, 做 threshold, canny 等 edge detection 處理, 能提高 contour 的辨識效果。

[2] Contour 的資料結構長什麼呢?

OpenCV 有個叫做 FindContour 的指令, 能找出圖像的 contour(s)。

寫隻小程序來實驗:

原圖:



程式:

```
1 import numpy as np
2 import cv2
3
4 im = cv2.imread('test3.png')
5 imggray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
6 ret, thresh = cv2.threshold(imggray, 127, 255, 0)
7 contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
8 print "there are " + str(len(contours)) + " contours"
9
10 cnt = contours[0]
11 print "there are " + str(len(cnt)) + " points in contours[0]"
12 print cnt
```

ADSENSE廣告

總網頁瀏覽量



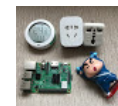
2,509,751

熱門文章



Home Assistant, (1) 系統安裝與簡單上手 (Internet of Things, Home Assistant)

最近這陣子, 萊恩大兵都泡在物聯網領域裡. 做了很多的嚐試, 學到不少的經驗. 有個好東西要介紹給大家~ Home Assistant .



Home Assistant, (2) 遙控小米開關與條件式自動化 (Internet of Things, Home Assistant)

最近這陣子, 萊恩大兵都泡在物聯網領域裡. 做了很多的嚐試, 學到不少的經驗. 有個好東西要介紹給大家~ Home Assistant . 在前一篇文章中, 萊恩大兵介紹了 HA 的系統安裝與簡單上手. 請參考: Home Assistant, (1) 系統...



泡綿實作, 遮蔽膠帶咬漆實驗 (EVA Foam)

萊恩大兵做 低面數黑豹頭 泡綿模型的過程中, 曾遇上一個麻煩 ~ 在上漆時, 萊恩大兵先將尖牙噴上銀漆. 待乾燥後, 用紙膠帶將尖牙黏起來做遮蔽, 再去噴其它顏色的漆. 沒想到撕下紙膠帶時, 原本亮閃閃的尖牙竟然掉漆了。



iOS App 上架流程, (1/3) 申請 Apple ID for Company

因為工作需要, 萊恩大兵上架了一隻 iOS App 到 Apple App Store . 整個申請/審核/發佈的過程有點繁瑣, 怕下次再要做的時候已忘光光, 便順手將上架流程記錄下來, 以備日後查看. 首先第一步, 要有一個開發者帳號. 開發者帳號可以個人, 也可...



藍色小舖, 回憶之光, 3D 列印的經驗分享 (3D printing)

一般想法, 要將 3D 印表機發揮最大效用, 最好是要能自己製圖建模型, 就是所謂 "自己的零件自己印" 的概念. 像萊恩大兵這種完全沒基礎的初階者, 有一台 3D 印表機之後, 大概就是上網下載模型來列印, 然後... 就沒有然後了。

BLOGGERADS

MAKER 好站連結

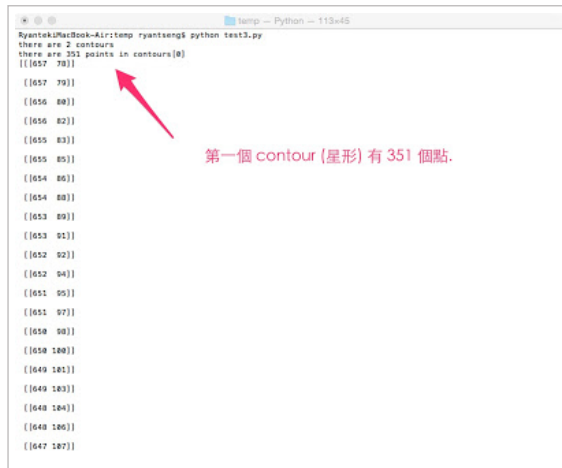
[馬馬夫夫 \(馬爸的部落格\)](#)[MAKERPRO \(新創圈開發社群平台: Maker to Startup\)](#)[Cooper Maa- 一家烤肉萬家香, 開放分享才會快樂](#)[葉難](#)[3D print and Arduino 學習筆記](#)[Ark Lab 多旋翼工坊 - Open Skyler](#)

```

13 cnt = contours[1]
14 print "there are " + str(len(cnt)) + " points in contours[1]"
15 print cnt
16
17
18 cv2.imshow('im', im)
19 cv2.waitKey(0)
20 cv2.destroyAllWindows()

```

執行結果:



從程式的執行結果可以看出: contour 的資料結構, 是有 n 組點座標串列 (list) 組成的大串列. (n 代表掃描到 contour(s) 的個數, 點座標串列則是某個 contour 的點數.)

[3] 對 contour 做多邊形逼近 (approxPolyDP)

對 contour 做多邊形逼近的目的, 可以想成是用粗一點的線來描邊, 來忽略掉細微的毛邊或雜點. 再用隻程式來實驗:

程式:

```

1 import numpy as np
2 import cv2
3
4 im = cv2.imread('test3.png')
5 imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
6 ret, thresh = cv2.threshold(imgray, 127, 255, 0)
7 contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
8 print "there are " + str(len(contours)) + " contours"
9
10 cnt = contours[0]
11 print "there are " + str(len(cnt)) + " points in contours[0]"
12 approx = cv2.approxPolyDP(cnt, 30, True)
13 print "after approx, there are " + str(len(approx)) + " points"
14 print approx
15 cv2.drawContours(im, [approx], 0, (255, 0, 0), -1)
16
17
18 cnt = contours[1]
19 print "there are " + str(len(cnt)) + " points in contours[1]"
20 approx = cv2.approxPolyDP(cnt, 30, True)
21 print "after approx, there are " + str(len(approx)) + " points"
22 print approx
23 cv2.drawContours(im, [approx], 0, (0, 255, 0), -1)
24
25 cv2.imshow('im', im)
26 cv2.waitKey(0)
27 cv2.destroyAllWindows()

```

執行結果:

動動手 輕鬆Make!

Richyway位老師教學部落格

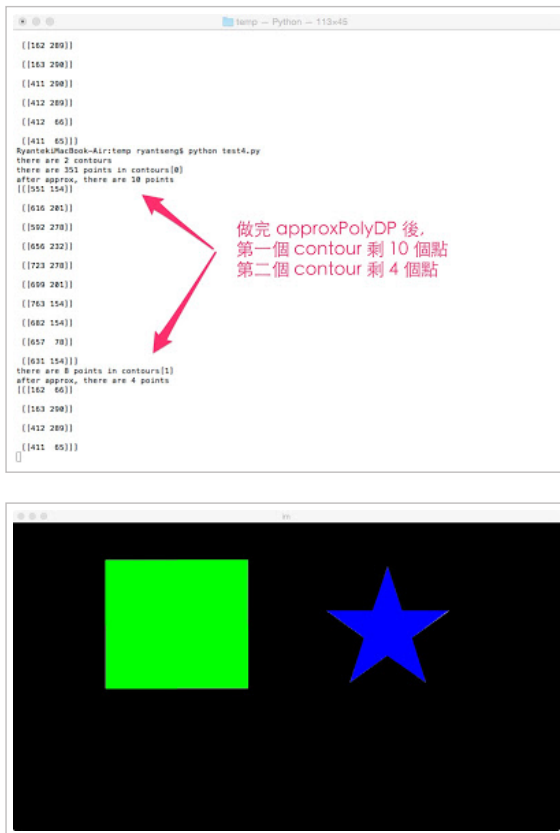
標籤

3D printing Alchemy Vision Animatronic Eye arduino Automatic Vehicle automation Bluemix bluetooth capacitive sensor CC2540 CG022 Charger Dialog Service Docker Drone electric imp electric speed control experiment fablab Face Recognition Force of Friction Futaba GPS gundam hackathon HC-06 HC-SR04 Humix I2C iBeacon IMU interactive internet of things iOS app java script Kalman Filter Kamen Rider Kumamon Kura Sushi laser cut LED LEGO Lightsaber LM35 make Maker Faire Taipei Masked Rider MAX7219 mbed microSD MPU6050 MultiWii murmur NanoWii Node-RED NXT OpenCV plot clock processing Python Qmote Quadcopter raspberry pi robot SG90 share Smart Glasses spacebrew Taiwando the physical web toy vibration motor Xively

網誌存檔

- ▶ 2020 (7)
- ▶ 2019 (18)
- ▶ 2018 (45)
- ▶ 2017 (66)
- ▶ 2016 (74)
- ▼ 2015 (85)
 - ▶ 十二月 (7)
 - ▶ 十一月 (7)
 - ▶ 十月 (6)
 - ▶ 九月 (8)
 - ▼ 八月 (7)
 - 學習記錄, LEGO NXT 機器人, (10) 用超音波感應器玩鬼抓人 (LEGO, NXT)
 - 學習記錄, LEGO NXT 機器人, (9) 用觸碰感應器做賽馬比賽 (LEGO, NXT)
 - iOS App 上架流程, (2/3) 產出 .P12 憑證與 Provisioning Profi...
 - iOS App 上架流程, (1/3) 申請 Apple ID for Company
 - 記錄, OpenCV 學習路徑, (2) 辨識多邊形 (OpenCV, Python)
 - 藍色小舖, 回憶之光, 3D 列印的經驗分享 (3D printing)
 - 拆, 震動發光充氣加油棒 (LED, vibration switch)
 - ▶ 七月 (8)
 - ▶ 六月 (5)
 - ▶ 五月 (8)
 - ▶ 四月 (8)
 - ▶ 三月 (6)
 - ▶ 二月 (7)
 - ▶ 一月 (8)
- ▶ 2014 (76)

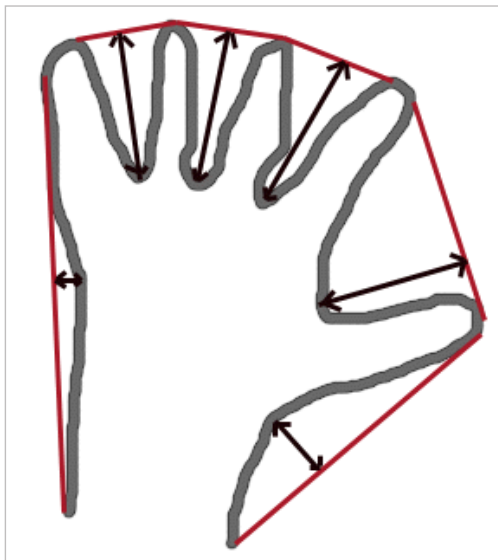
技術提供: Blogger.



從程式的執行結果可以看出: 第一個圖像為星形, 從 FindContour 找出來的 contour 卻有 351 個點, 在 approxPolyDP 處理過後, 才變成只有 10 個點. 若 approxPolyDP 的參數取的好 (用來逼近的線不會過粗, 也不會太細), 則根據輸出的座標點數, 就比較能判斷圖像的形狀. 藉由 FindContour 與 approxPolyDP 的處理, 就足夠寫隻辨識多邊形的小程式了.

[4] 什麼是 Convex Hull (凸多邊形框)?

看 [OpenCV Python Tutorials](#) 裡面的圖 (手) 即可明瞭, Convex Hull 概念上是取一個物件的凸多邊形框.



OpenCV 有個叫做 convexHull 的指令, 能找出圖像的 convex hull.

程式:

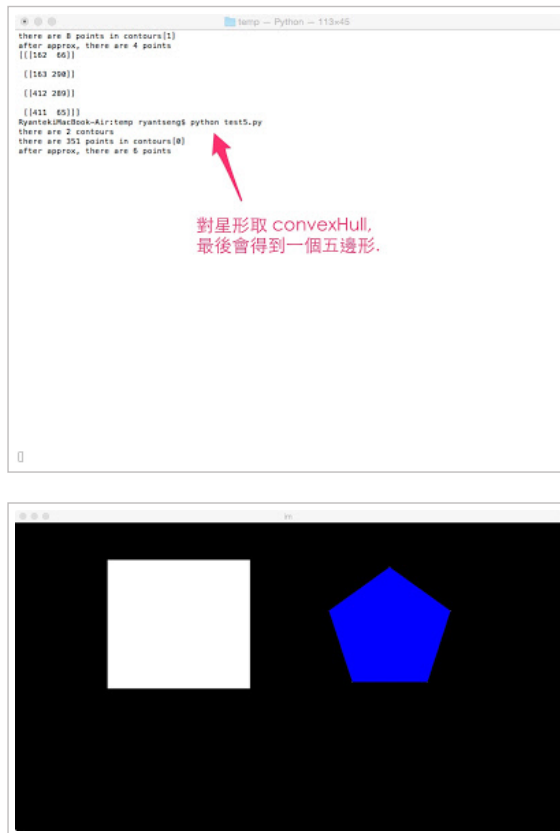
```
1 import numpy as np
2 import cv2
3
4 im = cv2.imread('test3.png')
5 imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
6 ret, thresh = cv2.threshold(imgray, 127, 255, 0)
7 contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
8 print "there are " + str(len(contours)) + " contours"
```

```

9
10 cnt = contours[0]
11 print "there are " + str(len(cnt)) + " points in contours[0]"
12 hull = cv2.convexHull(cnt)
13 print "after convexHull, there are " + str(len(hull)) + " points"
14 cv2.drawContours(im, [hull], 0, (255, 0, 0), -1)
15
16 cv2.imshow('im', im)
17 cv2.waitKey(0)
18 cv2.destroyAllWindows()

```

執行結果:



這個 Convex Hull 有什麼用呢? 往下看.

[5] 什麼是 Convexity Defects (凸多邊形缺陷)?

有了 Convex Hull 後, 它和原圖像的 deviation (偏差), 也就是那些凹陷的部份, 就叫做 Convexity Defects (凸多邊形缺陷?).

OpenCV 有個叫做 convexityDefect 的指令, 能找出圖像的 Convexity Defects.

看一下 Convexity Defects 的資料結構, 裡面有三個點座標與長度值.

```

1 struct CvConvexityDefect
2 {
3     CvPoint* start; // point of the contour where the defect begins
4     CvPoint* end; // point of the contour where the defect ends
5     CvPoint* depth_point; // the farthest from the convex hull point within the defect
6     float depth; // distance between the farthest point and the convex hull
7 };

```

程式:

```

1 import numpy as np
2 import cv2
3
4 im = cv2.imread('test3.png')
5 imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
6 ret, thresh = cv2.threshold(imgray, 127, 255, 0)
7 contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
8 print "there are " + str(len(contours)) + " contours"
9
10 cnt = contours[0]
11 print "there are " + str(len(cnt)) + " points in contours[0]"
12 hull = cv2.convexHull(cnt, returnPoints = False)
13 defects = cv2.convexityDefects(cnt, hull)
14 print "there are " + str(len(defects)) + " defects in contours[0]"
15
16 for i in range(defects.shape[0]):
17     s, e, f, d = defects[i, 0]
18     start = tuple(cnt[s][0])
19     end = tuple(cnt[e][0])
20     far = tuple(cnt[f][0])
21     cv2.line(im, start, end, [0, 255, 0], 2)

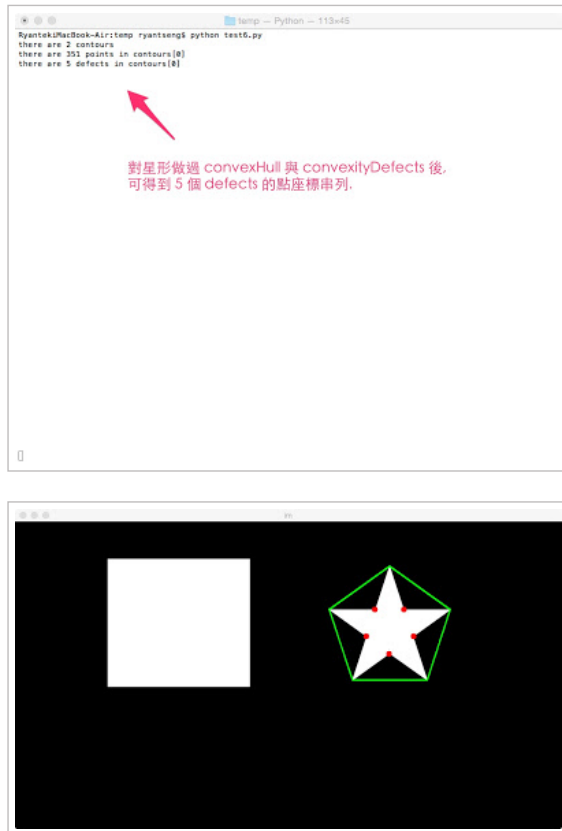
```

```

22     cv2.circle(im, far, 5, [0, 0, 255], -1)
23
24     cv2.imshow('im', im)
25     cv2.waitKey(0)
26     cv2.destroyAllWindows()

```

執行結果:



[6] 辨識箭形

有了 Convex Hull 與 Convexity Defects 的概念後, 就能來看 [Github: PyCV-time](#) 的辨識箭形程式碼。

```

1 def isArrow(heptagon):
2     hull = cv2.convexHull(heptagon, returnPoints = False)
3
4     if len(hull) > 2:
5         defects = cv2.convexityDefects(heptagon, hull)
6         if defects is None or len(defects) != 2:
7             return False
8
9         farpoints = [d[0][2] for d in defects]
10        if not np.abs(farpoints[0] - farpoints[1]) in [3, 4]:
11            return False
12
13        for defect in defects:
14            s, e, f, d = defect[0]
15            # print defects
16            # s, e, f, d = defect[0]
17            ps = heptagon[s, 0]
18            pe = heptagon[e, 0]
19            pd = heptagon[f, 0]
20            if angle(ps, pd, pe) < 120:
21                return True
22
23        return False

```

[7] 確保多邊形品質的程式技巧

看 [Github: PyCV-time](#) 的辨識五邊形/平行四邊形程式碼, 萊恩大兵學到幾招:

(i) 用多邊形的邊長或對角線長度的比值, 來篩選適當形狀的多邊形。

```

1 edges = [LA.norm(a-b) for a, b in zip(simp_ctr, np.roll(simp_ctr, 2))]
2 ratios = [(1 - e/edges[0])**2 for e in edges]

```

(ii) 用 2-norm 來計算兩個點之間的距離

關於 Norm 的概念, 請參考:

- * [OpenCV線性代數-cvNorm向量長度,距離計算](#)
- * [\[線性系統\] 拓談各種基本範數 \(Norm\)](#)

(iii) Python 對串列的處理, 包括: zip(), roll(), shape(), arrange(), tuple() 等

[8] 參考資料

- * [OpenCV Python Tutorials, Contours - 1 : Getting Started](#)
- * [OpenCV Python Tutorials, Contours - 2 : Brotherhood](#)
- * [OpenCV Python Tutorials, Contours - 3 : Extraction](#)
- * [OpenCV Python Tutorials, Contours - 4 : Ultimate](#)

(2015/8/11, 更新) 這篇文章在臉書上的一些討論.

黃雅信、謝小進、李信詮以及其他 244 人都說讚。

黃無名 讚喔~久沒看到你分享四軸,以為你沒玩了呢!

昨天 23:31 · 讚

曾建評 最近是沒怎麼玩四軸沒錯. 先繞去做別的東西, 未來會再合一起

昨天 23:33 · 讚 · 1

施翰誠 曾大, 可以加個好友一起增進嗎~

23小時 · 收回讚 · 2

王廣安 可以看看新的opencv3 新增了很多辨識的東西

23小時 · 收回讚 · 9

Cruise Chen 多謝分享

23小時 · 讚 · 1

Neil Lin 用cortex m4小顆的 把部分不要的linux移掉後把open cv移上去~可以更快

23小時 · 讚 · 1

龍青 有點不太了解OPENCV, 是不是OPENCV一定要裝在個人電腦上, 由WEBCAM作影像輸入, 再由COM Port和Arduino通訊, 作應用。這樣的架構, 如果作成飛行器, 會不會很麻煩?

15小時 · 讚

Jungfu Chen Neil 哥說的沒錯,cortex m4雙核心搭free rtos

14小時 · 讚 · 1

王廣安 openCV 也只是幫你包裝好的C++, 你可以用openCV快速的達到你要的功能, 再把裡面包好的C++ 放進去你的控制器就好了。

13小時 · 讚

龍青 openCV的底層, 沒限用哪種硬體架構嗎? M4有浮點運算器, 應該跑起來比較順。樹莓派, 甚至伽利略, 都可以用嗎?

13小時 · 已編輯 · 讚

王廣安 我指的是先了解"影像處理"的原理, 再去看openCV裡面的函示, 自然在任何處理器都可以自己寫, 至於openCV底層架構我就不太清楚了, 我只知道有很多版本, 可以支援android windows mac linux 等等

13小時 · 讚

龍青 你提的都是作業系統....看來OpenCV, 一定要綁在作業系統內了...

13小時 · 讚

王廣安 應該是吧, 所以才要了解"影像處理"的原理阿

13小時 · 讚 · 2

龍青 因為我一直用傳統的單晶片寫法在想這個事, 因為一般來說, 比較少用作業系統, 甚至用組合語言來寫。我再去了解一下OpenCV的架構好了。

13小時 · 讚

王廣安 樹莓派其實有opencv!

13小時 · 讚 · 1

[萊恩大兵的其它文章]

自製大四軸

- * [自製大四軸, 實作分享@華山文創園區](#)
- * [自製大四軸, \(1\) 零組件篇, 遙控器 \(Drone, Quadcopter, Futaba, Maker, Arduino, Animatronic Eye\)](#)
- * [自製大四軸, \(2\) 零組件篇, 飛控板 \(Drone, Quadcopter, MultiWii, Arduino, Futaba, Maker\)](#)
- * [自製大四軸, \(3\) 零組件篇, 自行雷切木質機架 \(Drone, Quadcopter, Maker, Laser Cut\)](#)
- * [自製大四軸, \(4\) 零組件篇, 馬達與電變調整 \(Drone, Quadcopter, Maker, Electric Speed Control, Motor\)](#)
- * [自製大四軸, \(5\) 組裝篇, 四軸飛行器成形 \(Drone, Quadcopter, MultiWii, Arduino, Maker, Electric Speed Control, Motor\)](#)
- * [自製大四軸, \(6\) 調整篇, 飛行前兩三事 \(Drone, Quadcopter, Maker, Futaba, Arduino, MultiWii\)](#)
- * [自製大四軸, \(7\) 充電篇, iMax B6 充電器操作記要 \(Charger, Battery\)](#)
- * [自製大四軸, \(8\) 問題篇, 機架損壞維修 \(Drone, Quadcopter, Laser Cut\)](#)
- * [自製大四軸, \(9\) 改良篇, 方便拆卸的木質機架 \(Drone, Quadcopter, Maker, Laser](#)

Cut)

- * 自製大四軸, (10) 外飛篇, 新手的青蛙跳與遛狗 (Drone, Quadcopter, Maker, MultiWii)

自動報球速的棒球

- * 自動報球速的棒球, (1) 概念與雛形 (Arduino, MPU-6050, HC-06)
- * 自動報球速的棒球, (2) 第一版試作品 (Arduino, NanoWii, microSD, MPU6050)
- * 自動報球速的棒球, (3) 拋接實驗的數據分析 (Arduino, NanoWii, microSD, MPU6050)
- * Arduino, 自己的電源線自己做
- * 筆記, HC-06 藍牙模組, 搭配 Arduino 建立 serial port 連線 (Bluetooth, HC-06, Arduino)
- * 筆記, MPU-6050, google 搜尋與實驗心得
- * 做實驗, 寫入 EEPROM 的速度能否跟得上 MPU6050 的數據產出? (Arduino, MPU-6050, EEPROM)
- * 筆記, NanoWii, 一些經驗分享 (Arduino, NanoWii, MPU6050)
- * Murmur, 很小很強大的穿戴式裝置模組 (Realtag, Bluetooth, CC2540, MPU6050, BMP180)

CC2540 Bluetooth Low Energy

- * 筆記, CC2540 Bluetooth Low Energy, (1) 開發環境 架設 (Bluetooth, CC2540)
- * 筆記, CC2540 Bluetooth Low Energy, (2) 跑第一個範例程式 (Bluetooth, CC2540)
- * 筆記, CC2540 Bluetooth Low Energy, (3) SimpleBLEPeripheral 簡單介紹 (Bluetooth, CC2540)
- * 筆記, CC2540 Bluetooth Low Energy, (4) 在智慧手機上執行範例程式 (Bluetooth, CC2540)
- * 筆記, CC2540 Bluetooth Low Energy, (5) 偵測與發送 iBeacon 訊號 (Bluetooth, CC2540, iBeacon)
- * 實作, iBeacon 發訊器 x 防丟器 (Bluetooth, CC2540, iBeacon)
- * 實作, iBeacon 尋寶遊戲 (Bluetooth, CC2540, iBeacon, iOS app)
- * 實作, BLE + iOS app, 遙控燈泡君 (Bluetooth, CC2540, iOS app)
- * 做實驗, 用 iBeacon 做自動控制的可行性 (Bluetooth, iBeacon, CC2540, Automation, URL Scheme, iOS app)

藍色小鋪一起來做

- * 藍色小鋪一起來做, (1) 用 beacon 控制開關的柏燈
- * 藍色小鋪一起來做, (2) 講解 BLE CC2540 UART 通訊範例程式 (Bluetooth, CC2540, UART)
- * 藍色小鋪一起來做, (3) 藍牙柏燈專案實作 (上) (Bluetooth, CC2540)
- * 藍色小鋪一起來做, (4) 藍牙柏燈專案實作 (下) (Bluetooth, CC2540)
- * 藍色小鋪一起來做, (5) iBeacon scanner 專案示範與解說 (Bluetooth, CC2540, iBeacon)
- * 藍色小鋪一起來做, (6) 完成, 用 iBeacon 控制開關的柏燈 (Bluetooth, CC2540, iBeacon)
- * 藍色小鋪, 初嚐樹莓派 (Raspberry Pi)
- * 藍色小鋪, iBeacon 應用, 自動記錄到訪時間 (iBeacon, Geohopper, Zapier)
- * 藍色小鋪, 菲力普的 Docker 應用分享 (Docker)
- * 藍色小鋪, PTT地震文團隊分享三連發, (1) Maker 的 IOT 遊樂場 (PTT, Hackathon, Python, Xively, Internet of Things)
- * 藍色小鋪, PTT地震文團隊分享三連發, (2) mbed 新手分享 (mbed, MPU6050, Hackathon, Internet of Things)
- * 藍色小鋪, 空中提升軌道車, 作品進化分享 (Pneumatic Tube System, Force of Friction, Mini 4 WD)
- * 藍色小鋪, 數字管時鐘, 作品進化分享 (Nixie Tube, VFD)
- * 藍色小鋪, 回憶之光, 3D 列印的經驗分享 (3D printing)

OpenCV 學習路徑

- * 記錄, OpenCV 學習路徑, (1) 環境安裝與第一個範例 (OpenCV, Python)

小惡魔 無線溫度感測器

- * 小惡魔, (1) 溫度感測 + 物聯網 (Electric Imp, Xively, LM35, Internet of Things)
- * 小惡魔, (2) 溫度感測 + 物聯網 + 事件觸發 (Electric Imp, Xively, LM35, Internet of Things)

108 大眼仔

- * 108 大眼仔, 初登場 (Arduino, SG-90, Maker Faire Taipei 2014, Animatronic Eye)
- * 108 大眼仔, 進化, (1) 專屬程式庫 (Arduino, SG90, Animatronic Eye)
- * 108 大眼仔, 進化, (2) 當我們串在一起 (Arduino, SG90, Animatronic Eye, I2C)
- * 108 大眼仔, 進化, (3) 檢查 Gmail 信箱 (Arduino, SG90, Animatronic Eye, Temboo)
- * 108 大眼仔, 進化, (4) 看著我的臉 (Arduino, SG90, Animatronic Eye, OpenCV, Processing, I2C)
- * 108 大眼仔, 進化, (5) 迎著人來人往 (Arduino, SG90, Animatronic Eye, OpenCV, Processing, I2C)
- * 108 大眼仔, 檢討筆記, 我要一個打十個 (Arduino, SG90, Animatronic Eye)

Plot Clock

- * 自造 Time Plotting Clock, (1) 零件採購
- * 自造 Time Plotting Clock, (2) 組裝
- * 自造 Time Plotting Clock, (3) 校正
- * 自造 Time Plotting Clock, (4) 英文字母

- * 自造 Time Plotting Clock, (5) 遠端遙控
- * Murmur, 有趣的零件售價
- * Murmur, Arduino 保險桿 (Arduino, bumper, 3D printing)
- * Murmur, 許一個 maker 分享網站
- * Murmur, 物聯網新概念- The Physical Web (Internet of Things, The Physical Web)
- * Murmur, 關於 HP Sprout 的一點想法
- * Murmur, 說中文很難嗎? (Toy, Reed Switch, Voice Recorder Module)
- * Murmur, 停車場自動繳費機的兩三事 (Kiosk)
- * Murmur, 為什麼是 WiFi? 關於小米空氣清淨器的一點看法.. (Internet of Things)
- * Murmur, 機器人是時尚元素? (Robot, Fashion)
- * Murmur, Nordic 才會是 Bluetooth LE 晶片戰爭的恐龍王者? (Bluetooth, Nordic)
- * Murmur, Maker Faire Taipei 2015, 讓人印象深刻的創作品 (Maker Faire Taipei)
- * 土砲, 電阻收納卡
- * 踢鐵板的自製 microSD 模組
- * 好創意, 心跳機械人 (Heart Bot, Plot Clock)
- * Arduino LED 跑馬燈 (SpaceBrew, Processing, JavaScript, MAX7219, LED)
- * 遠端遙控 Arduino 倒一杯水 (SpaceBrew, Firmata, SG90, Java Script, Processing, Internet of Things)
- * 好設計, Maker 專屬 T-shirt (maker, t-shirt)
- * 新奇, 上海地鐵車廂外移動廣告 (Signage, Zoetrope)
- * User Story, 感測器與自動控制的軟硬整合產品 (mFi, internet of things)
- * 體驗, 原住民互動故事書@宜蘭大同鄉泰雅雅生活館
- * 體驗, 蛋生音互動裝置@兒童美術館 (Arduino, 3D Printing, HC-SR04, Interactive)
- * 開箱, 鋼彈小劇場 (Pepper's Ghost, GUNDAM)
- * 開箱, Flying Alien Sphere 異形飛球 (Drone, Flying Alien Sphere)
- * 開箱, 偉力控二號機, 小四軸飛行器 (CG022, Quadcopter)
- * 偉力控二號機, 修理防護罩與飛行心得 (CG022, Quadcopter)
- * 偉力控二號機, 我想有個家 (CG022, Quadcopter)
- * 偉力控二號機, 換馬達 (CG022, Quadcopter)
- * 偉力控二號機, 盒子上的洞 (CG022, Quadcopter)
- * 記錄, 參觀帆船工廠@哥倫遊艇
- * 滿到溢出來的機器人零距離體驗@祥儀機器人夢工廠
- * Arduino作品集, (1), 電容感應音樂樹 (Capacitive Sensor)
- * 筆記, microSD 模組, 一些經驗分享 (Arduino, microSD)
- * 筆記, 低電壓警報器的用法 (Low Voltage Buzzer)
- * 筆記, 應用 OpenCV 做目標捕獲 (OpenCV, Python)
- * 實作, 電容感應音樂樹
- * 實作, Arduino 聲納雷達 (HC-SR04, SG90)
- * 豈止於大, 謝銑鈺的鋁罐燈 (Arduino, MAX7219, LED)
- * 修理, 樂高馬達 8883 (LEGO 8883 Power Functions M-Motor)
- * 修理, (part 2) 樂高馬達 8883 (LEGO 8883 Power Functions M-Motor)
- * 修理, 液晶螢幕 (LCD)
- * 想修理, 免治馬桶座的薄膜按鍵開關
- * 想拆, 聲控錄音變聲熊本熊 (Kumamon, Vibration Motor, VOX)
- * 想拆, 小七的無線充電座 (Wireless Charging, 7-11)
- * 拆, 電流急急樂
- * 拆, 無線光學滑鼠
- * 拆, 元宵節紙雕小燈籠燈芯 (7-color blinking LED)
- * 拆, 堅若磐石手電筒
- * 拆, 省電燈泡 (Compact Fluorescent Bulb)
- * 拆, 玩具吸塵器 (toy, step motor, vibration motor)
- * 拆, USB 車用充電器 (USB Car Charger)
- * 拆, 震動發光充氣加油棒 (LED, vibration switch)
- * 做實驗, 香蕉也能當開關
- * 做實驗, 寫入 EEPROM 的速度能否跟得上 MPU6050 的數據產出? (Arduino, MPU-6050, EEPROM)
- * 做實驗, 把 iPad 釘在牆上
- * 做實驗, 全世界最簡單的迷你電車
- * 做實驗, 全世界最簡單的馬達
- * 口丁工作坊, 轉不停陀螺 (Perpetual Spinning Top)

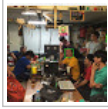
RELATED POSTS


筆記, 應用 OpenCV 做目標捕獲 (OpenCV, Python)

萊恩大兵前些天看到菲力普貼了一篇應用 OpenCV 做特定圖像的對比與追蹤的文章 (Target acquired: Finding targets in drone and quadcopter)...[Read more](#)


記錄, OpenCV 學習路徑, (1) 環境安裝與第一個範例 (OpenCV, Python)

因為想在四軸飛行器上整合影像辨識做些好玩的事, 所以萊恩大兵動念想自學 OpenCV. 胡亂摸索一陣之後, 覺得 OpenCV + Python 的學習資源較豐富, 就往這方向投入了. 簡單記...[Read more](#)


記錄, OpenCV 學習路徑, (3) 人臉辨識 (OpenCV, Python, Face Recognition)

OpenCV 人臉辨識, 就差沒認出馬爸. 因為想在四軸飛行器上整合影像辨識做些好玩的事, 所以萊恩大兵動念想自學 OpenCV. 胡亂摸索一陣之後, 覺得 OpenCV + Python...[Read more](#)


路過, 在某停車場發現的智慧尋車系統 (OpenCV, Computer Vision)

萊恩大兵看到一個很不錯的電腦視覺 (Computer Vision) 技術的應用案例. 最近, 民權公園地下停車場新設了一個智慧尋車系統. 它是一台自助式的觸控查詢機 (KIOSK), 讓停...[Read more](#)


藍色小舖, PTT地震文團隊分享三連發, (1) Maker 的 IOT 遊樂場 (PTT, Hackathon, Python, Xively, Internet of Things)

萊恩大兵在 Openlab Taipei 混了一年多, 這期間遇見很多才華洋溢的 maker. 他們各有擅長, 作品創意滿載. 只是, 這些作品常常都是做完了在臉書貼張相片得...[Read more](#)

ALSO ON 大兵萊恩 一路直前

從紙模到 3D 建模, (9) 噴火龍 瓦楞紙版本 ...

3 years ago • 2 comments

萊恩大兵摸索實作紙模的經驗分享.

週二來一招, Node-RED 簡單講講 (Node-RED)

5 years ago • 4 comments

分享記錄~ Node-RED 簡單講講.

藍色小舖一起來做, (2) 講解 BLE CC2540 ...

6 years ago • 5 comments

藍色小舖 BLE 研究會. 第一個實作計畫的第二次聚會. 聚會主題是講解 BLE ...

Home 裝 MQTT

a year

Home Mosquitto MQTT

2 Comments 大兵萊恩 一路直前 [Disqus' Privacy Policy](#)

[Login](#)

[Recommend](#) [Tweet](#) [Share](#)

Sort by Best



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name