

DIP Final Report

鷹眼 - 籃球影片動態捕捉

• 目標

- 偵測出籃球在影片中的位置
- 追蹤籃球移動軌跡並判斷該次出手是否有進球

• 動機

- 網球比賽中有實際應用，且非常精準，想在籃球場上嘗試
- 希望可以提供更準確的判決

• 實作步驟

1. 將影片切成許多frame以便處理

- 透過 `OPENCV` 的函式，可將影片以35~40 fps讀入
- 每個frame可以被視作一張彩色圖片來進行處理

2. 偵測目標物體所在位置與其範圍，透過顏色偵測與形狀偵測兩個方法提高準確率

■ 顏色偵測

1. 為了方便調整適當的顏色區間，首先將原本RGB表示的色彩空間轉換為以HSV表示的色彩空間。
2. 利用**color filter**來將我們想要的顏色區間給留下來。但因為在不同角度或位置下，物體的色彩的會有所差異，因此使用filter之後仍會有多餘物體被留下來。以下為我們針對不同物體使用的color filter，以HSV表示。

- 籃球： $(-30, 80, 70) \sim (15, 205, 205)$
- 籃框： $(0, 43, 46) \sim (4, 205, 205)$
- 籃網： $(0, 0, 70) \sim (180, 43, 205)$
- 下方為原圖及使用籃球的color filter後的結果，可看出上方籃球被留下來了，但也有多餘的物體如手沒有被濾掉。



3. 接著透過形態學中的**opening**來使留下來的圖形形狀變得更加完整平滑。
4. 我們使用 `findContours()` 這個函式將圖片中顏色相近且相連的物體找出，並且利用找出的Contours的面積與平均y座標來篩選出我們真正要的物體。
5. 透過顏色偵測可以個別準確地找出籃框以及籃網的位置及範圍，但籃球會有所誤差，因此我們將顏色偵測找出的籃球設為candidate，配合下個步驟的形狀偵測。

■ 形狀偵測

1. 我們使用**Background subtraction**來去除背景，也就是只留下前景在移動的物體。
 - 實作上使用 `createBackgroundSubtractorMOG2()` 這個函式，依照 *Zoran Zivkovic, Ferdinand van der, Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. 2006.* 這篇paper來實作。
 - 使用background subtraction後如下，右方為做完之後的結果



2. 因為原本的影像可能有很多複雜的物體在內，為了方便我們偵測，我們一樣使用和顏色偵測相同的**color filter**來去除多餘的東西。
3. 利用**bilateral filter** 和 **Gaussian filter**來去除影像中多餘雜訊，測試下不同大小的filter結果差異不大，最後我們使用9x9的filter。
4. 接著一樣使用形態學中的**opening**來使留下來的圖形變得完整。
5. 對影像做**邊緣偵測**，這邊我們採用**Canny edge detection**，因為我們要找的圖形並不複雜，也不需要偵測一些內部的細節，因此邊緣偵測的好與壞對我們的影響不大。
6. 最後使用 `findContours()` 來找出相連contours後，利用 `minEnclosingCircle()` 來找出包含這個contours的最小圓，比較這些圓和顏色偵測留下的candidate的面積與圓心距離的關係，最後找出真正的籃球位置。
 - 原本我們使用Hough transform來找出圓形物體，但因為影像中物體太多，加上圓形可能不夠完整，因此做出來效果很差，故改用其他方法來找圓形。

3. 追蹤與預測球的移動軌跡

- 因為原本影片可能不夠清晰，或者在某些frame中球的形狀可能不夠完整，會造成無法偵測到的情形，在這種情況下，我們需要預測球可能在的位置。
- 我們測試了兩個不同的方法，拉格朗日插值法與卡爾曼濾波。
- **拉格朗日插值法 (Lagrange Interpolation Polynomials method)**
 - 這個方法就是透過前三個已知的籃球球心直所在位置，形成一個二次曲線之後，預測下個瞬間球心所在位置。
 - 這個方法實作很簡單，但只透過三個點來預測一定不夠完整，我們做出來的結果是，在小曲線的運動軌跡下預測很精準，但如果是比較大弧度的拋物線，則會出現很大的誤差。
- **卡爾曼濾波器 (Kalman filter)**
 - 根據卡爾曼濾波器，可以透過預測值、測量值與相關的變異數來追蹤籃球的軌跡

Predict

$$\hat{x}_t^- = F_t \hat{x}_{t-1} + B_t u_t$$

$$\Sigma_t^- = F \Sigma_{t-1} F^T + Q$$

Update

$$K_t = \Sigma_t^- H^T (H \Sigma_t^- H^T + R)^{-1}$$

$$\hat{x}_t = \hat{x}_t^- + K_t (y_t - H \hat{x}_t^-)$$

$$\Sigma_t = (I - K_t H) \Sigma_t^-$$

- 實作上：我們使用 <https://github.com/SriramEmarose/Motion-Prediction-with-Kalman-Filter> 的作法，我們只要依序把偵測出的籃球球心座標丟到卡爾曼濾波器中，即可得到預測出的下個軌跡位置。
- 做出來的結果比拉格朗日法精準，但仍然容易出現誤差
- 最後我們採用卡爾曼濾波器來進行預測，但因為實際上在我們先前偵測物體時，會出現沒有偵測到的時候不多，因此會使用到預測軌跡的地方不多，預測誤差造成的影響也就比較小了。

4. 進球判斷

- 我們用很簡單的方法來偵測，只要球的位置由上而下經過籃框位置，那就有機會是進球。
- 有時會遇到籃外空心的問題，也就是雖然球從上而下經過籃框座標，但實際上球是從籃框外經過的，如下圖



- 為了排除籃外空心的問題，我們觀察進球與籃外空心球時籃框與籃網的變化
 - 進球時：籃框面積與質心無大幅度變動，籃網面積無大幅變動。
 - 籃外空心：籃框面積與質心在球經過時會有大變動，籃網與籃框位置也會發生變動。
- 因此我們針對籃框及籃網的變動來判斷是否為籃外空心球。
- 最後，若進球，我們會在影片上呈現出"make"的字眼。

● 結果

- 所有測試影片僅包含三位球員在內，其中兩位負責撿球，影片中可能同時出現數顆球。鏡頭角度會隨著球有小幅度的移動。球員進行定點投籃，偶爾會有小幅度的移動。
- Demo影片（共6次出手）

正確判斷	錯誤判斷	進判斷不進	不進判斷進	準確率
6	0	0	0	100%

- Evaluation影片（共128次出手）

正確判斷	錯誤判斷	進判斷不進	不進判斷進	準確率
95	33	21	12	74.2%

● 實作上遇到的困難或缺點

1. 當遇到影片較模糊，或者鏡頭角度有比較大的變動時，容易出現誤判或漏判。
2. 還無法處理較複雜的影像，如真正籃球場上有許多球員、觀眾時，無法偵測到目標物體，且鏡頭跟著球員大幅度變動時，背景去除法就沒有效果。
3. 實作內容較針對單一影片，若遇到不同角度拍攝的影片，或者色調較不同的影片時，需要手動調整程式內的參數，比較不通用。

● 未來願景

1. 提升判斷進球的準確性
2. 增加額外的功能，如偵測球有沒有出界、出手為兩分球或者三分球。
3. 增加這個系統的通用性，希望可以應用到各個真實比賽的影像中，並提供即時的判斷。

● 參考資料

- Shiu-KuWeng, Chung-MingKuo, Shu-KangTu. Video object tracking using adaptive Kalman filter. Journal of Visual Communication and Image Representation, 2006.
- Bodhisattwa Chakraborty, Sukadev Meher. Real-Time Position Estimation and Tracking of a Basketball. 2012 IEEE International Conference on Signal Processing, Computing and Control .
- Zoran Zivkovic, Ferdinand van der, Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. 2006.