

Computer Vision- Volleyball Tracking

Fatbardha Hoxha - 189064 - fatbardha.hoxha@studenti.unitn.it

21 Maggio 2018

1 Introduzione e Obiettivi

Questo progetto si colloca nell'ambito di video tracking applicato agli sport al fine di fornire un supporto agli allenatori e ai giocatori. In questo particolare caso, lo sport in questione e' la pallavolo e lo scopo principale del progetto e' quello di verificare se la palla tocca la rete, poiche' questa azione comporta una penalizzazione durante una partita. Visionare i filmati di allenamenti o partite per individuare il verificarsi di quest'azione potrebbe richiedere molto tempo, quindi un'applicazione che e' in grado di individuare in modo automatico in quale momento questa azione si verifica, potrebbe essere molto utile per allenatori e giocatori.

2 State of the Art

Prima di iniziare a lavorare a questo progetto, ho cercato dei paper che mi fornissero lo stato dell'arte in questo campo e le principali tecniche adottate per tracciare la palla e il suo andamento.

Dalla letteratura e' emerso che l'analisi dei video di pallavolo, calcio, basket, tennis etc, riscontra delle problematiche comuni nel tracking della palla, quali:

- necessita' di video ad alti fps poiche' la palla si muove ad alte velocita' e le sue dimensioni sono molto piccole, soprattutto nel tennis;
- spesso la telecamera non riesce ad inquadrare la palla perche' essa viene nascosta dal corpo del giocatore;
- l'interazione palla-giocatore spesso e' rapida e imprevedibile;
- L'andamento della palla e' molto diverso quando e' in possesso del giocatore rispetto a quando compie un tragitto in aria.

Ball tracking in sports [3], riporta i principali vantaggi e svantaggi del tracking della palla nei diversi sports. Nel mio caso di interesse, la pallavolo, la principale problematica e' data dall'occlusione della palla da parte dei giocatori, ma grazie alla sua buona dimensione (diametro medio di 0.20 m) e alla velocita'

che e' relativamente bassa (massimo 120km/h), tracciare la palla risulta piu' semplice.

Le tecniche piu' avanzate per il tracking della palla utilizzando un sistema basato sulla predizione della traiettoria di una palla durante una partita. Questi metodi, sfruttano schemi di giochi ripetitivi per poter fare delle ipotesi accurate sullo spostamento della palla ed escludere ipotesi fisicamente impossibili.

L'idea di base descritta nel paper, *A Scheme for Ball Detection and Tracking in Broadcast Soccer Video* [5], e' quella di usare un grafo per descrivere le varie ipotesi di dove potrebbe trovarsi la palla. Successivamente, viene estratto il path ottimale dal grafo che descrive la posizione della palla. Questo procedimento viene applicato per riuscire a discriminare la palla da altri oggetti che potrebbero avere una forma simile ad essa e per migliorare l'operazione di tracking.

Nel paper *Tracking of Ball and Players in Beach Volleyball Videos* [1] vengono comparati due metodi per il tracking della palla; il primo e' Particle Filter mentre il secondo e' Integral Histogram. Quest'ultimo consiste nel computare gli istogrammi delle possibili regioni target, nello spazio cartesiano. Integral Histogram riporta delle prestazioni migliori rispetto a Partical Filter, perche' in Partical Filter i Particle vengono distribuiti in modo casuale attorno allo stato piu' probabile, mentre in Integer Histogram viene data una distribuzione predefinita dei Particle, in questo modo si ottengono delle prestazioni di calcolo migliori.

Nella letteratura analizzata, emerge che la parte di detection viene effettuata tramite Backround Subtraction, con erosion o dilation per definirne meglio i contorni, infine viene analizzata la forma dei blob, o contour, ottenuta. Il tracking invece viene fatto basandosi sulle probabilita' generate dagli schemi di gioco.

3 Aquisizione dei video

Per la realizzazione di questo progetto e' stato necessario acquisire dei video fondamentali per le fasi di progettazione e realizzazione del programma. I video sono stati realizzati tramite una GoPro Hero5, poiche' questa videocamera permette di registrare video ad alti frame rate, fino a 240 fps, e di avere un buon grandangolo in grado di riprendere tutto il campo da pallavolo.

Di seguito i tre principali posizionamenti della videocamera.

- **Posizione ideale.** Inizialmente si e' pensato di posizionare la videocamera in modo perpendicolare alla rete, cosi' da poter eliminare i problemi dati dalla prospettiva e poter facilmente individuare il momento in cui la palla sfiora la rete. Agli estremi della rete vi sono due tiranti che la collegano ai pali, questa conformazione rende impossibile posizionare la videocamera in modo perpendicolare alla rete.
- **Videocamera posizionata sul palo della rete.** Fissando la GoPro in questo punto si ottiene un'ampia visuale di tutto il campo di gioco, tuttavia vi sono delle problematiche legate al rumore creato dalla palla

quando si scontra sulla rete e alla presenza di giocatrici nelle vicinanze della rete.

- **Videocamera posizionata su un treppiede.** Per evitare il rumore causato dalla palla quando tocca la rete, la videocamera e' stata posizionata su un treppiede accanto alla rete, in modo che non subisse nessun urto. Posizionando la videocamera in questo punto si riscontrano due problematiche. La prima e' che il treppiede potrebbe essere urtato da qualche giocatore durante la partita; la seconda problematica e' causato dall'inquadratura, poiche' viene ridotta la visuale e si riprende solo una meta' campo, l'altra meta' viene oscurata dalla rete.



Figure 1: Posizione ideale

Figure 2: Videocamera sul palo

Figure 3: Videocamera sul treppiede

Analizzando i video ho notato che il punto migliore per fissare la videocamera corrisponde al palo della rete. In questa posizione si riesce ad inquadrare tutto il campo. Inoltre, il rumore causato dalla palla quando collide con la rete, verrà sfruttato per segnalare che la palla ha toccato la rete.

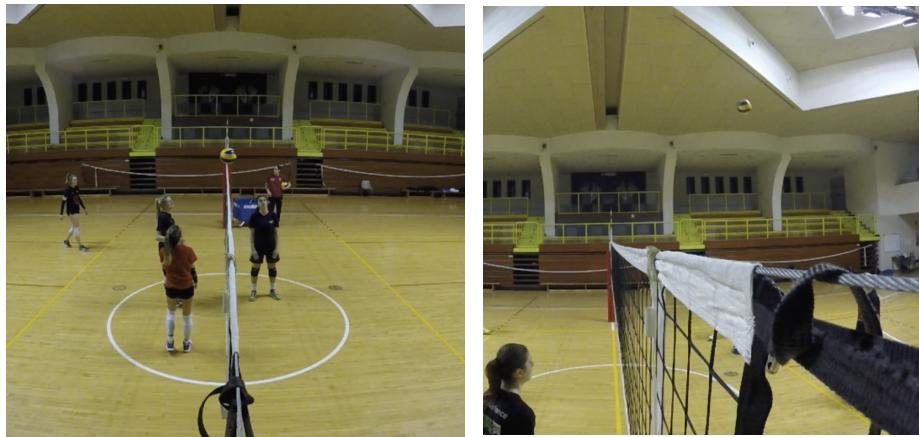


Figure 4: Inquadratura dal palo

Figure 5: Inquadratura dal treppiede

4 Struttura del codice e metodi applicati

Per la realizzazione di questo progetto e' stato utilizzato il linguaggio Python e la libreria OpenCV.

4.1 ROI

Inizialmente e' stata applicata una **ROI**, in modo da limitare la regione di interesse attorno all'area della rete. In questo modo si possono escludere i rumori creati dalle giocatrici e si puo' velocizzare la fase di detection e tracking della palla. Se i video vengono registrati sempre dalla stessa angolazione e con la stessa risoluzione, questo parametro puo' rimanere invariato, altrimenti e' necessario aggiustare il range della ROI poiche' si basa sui pixel del video in input.

4.2 BG Subtraction

Data la staticita' del background si e' potuto applicare l'operazione di Background Subtraction. Per rendere piu' precisa quest'operazione si e' optato per **Background Subtraction with Mixture of Gaussians** (linea 1). I parametri di questa funzione sono: *length of history*, che e' stata impostata a *500*, *threshold* che definisce quanto un nuovo candidato e' simile al suo esempio iniziale, impostata a *16* e infine *shadow detection* che e' stato impostato a *0* =false.

Infine, per definire ulteriormente i contorni della palla, sono state applicate le operazioni di opening e closing, utilizzando come forma morfologica un ellisse con kernel 3x3.

Listing 1: Background subtraction code

```
1 fgbg = cv2.createBackgroundSubtractorMOG2(500, 16, 0)
2 fgmask = fgbg.apply(frame2)
3 opening = cv2.morphologyEx(fgmask, cv2.MORPHOPEN, kernelOp)
4 closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, kernelC1)
```

4.3 Ball Detection

Per trovare i contorni del blob, ottenuto dall'operazione precedente, si e' utilizzata la funzione di **Contour Detection**. Per ogni contour individuato si sono posti dei limiti sull'area (areaMin=500px e areaMax=1500) e sull'aspect ratio, in modo da poter escludere le figure delle giocatrici in campo ed altre figure che non corrispondono alla palla.

4.4 Ball Tracking

Per il tracking della palla e' stata applicata la tecnica di frame differencing e successivamente e' stata calcolata la distanza, in pixel, prendendo in consider-

azione il centroide della palla. Per eseguire quest'operazione e' stata creata una funzione che aggiorna continuamente le nuove coordinate delle palla. In questo modo tramite l'operazione *getX*, *getY* e *getFrame* posso ottenere la posizione precedente, in pixel, in modo da poterla confrontare con quella attuale. Per essere piu' comoda nel calcolare la velocita' in metri/secondo, invece dei frame ho selezionato i millisecondi che sono poi stati convertiti in secondi. Una volta ottenuto lo spostamento in pixel, si e' disegnata la traiettoria percorsa dalla palla (linee 7-8).

Listing 2: Ball Tracking

```

1 space_x = abs(x - i.getX())
2 space_y = abs(y - i.getY())
3 #curr_frame = cap.get(cv2.CAP_PROP_POS_FRAMES)
4 curr_frame = cap.get(cv2.CAP_PROP_POS_MSEC)*0.001
5 delta_frame=abs(curr_frame-i.getFrame())
6 ....
7 pts = np.array(i.getTracks(), np.int32)
8 pts = pts.reshape((-1, 1, 2))
9 frame = cv2.polylines(frame2, [pts], False, (255,0, 0))

```

4.5 Calcolo della Velocita'

OpenCV [2] suggerisce di calibrare la videocamera tramite una scacchiera, ma per evitare quest'operazione, si e' pensato di sfruttare il concetto di profondita' del campo e di decadenza lineare delle dimensioni della palla. Come descritto dal link [4], possiamo calcolare la distanza del soggetto dalla videocamera, applicando il concetto di similarita' dei triangolo che ci permettono di derivare la lunghezza focale percepita dalla videocamera (illustrato in *Figure 6*).

La distanza focale e' definita dalla formula:

$$focalLength = (w_pixel * KNOWN_DISTANCE) / KNOWN_WIDTH.$$

In questa funzione, *KNOWN_DISTANCE* e *KNOWN_WIDTH* corrisponde alla distanza iniziale, in metri, dell'oggetto dall'obiettivo (larghezza del campo corrisponde a circa 9m) e alla larghezza dell'oggetto (il diametro di una palla da pallavolo e' di circa 0.20m). Mentre *w_pixel* equivale al diametro della palla, in pixel, quando si trova ad una distanza che pari a *KNOW_WIDTH*.

Una volta ottenuta la profondita' del campo, ho diviso il campo in quattro sezioni come illustrato in *Figure 7*, in modo da poter applicare il concetto di decadenza lineare, delle dimensioni della palla, in base alla profondita'. La palla ha una decadenza lineare del 10%: a 10 metri la sua dimensione (diametro) corrisponde a 10-20px mentre ad 1 metro dall'obiettivo abbiamo una dimensione (diametro) di 110-120px.

Dopo aver calcolato la distanza dalla videocamera ho convertito l'unita' di misura da pixel in metri (linee 8-9). Infine la velocita' totale in x e in y e' stata calcolata applicando la formula a linea 10. Questo calcolo della velocita' e' molto approssimativo, infatti non tiene conto delle principali leggi della fisica,

primo fra tutti il moto parabolico che spesso si presenta nella traiettoria della palla.

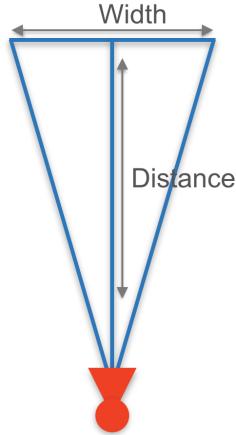


Figure 6: Calcolo profondita'

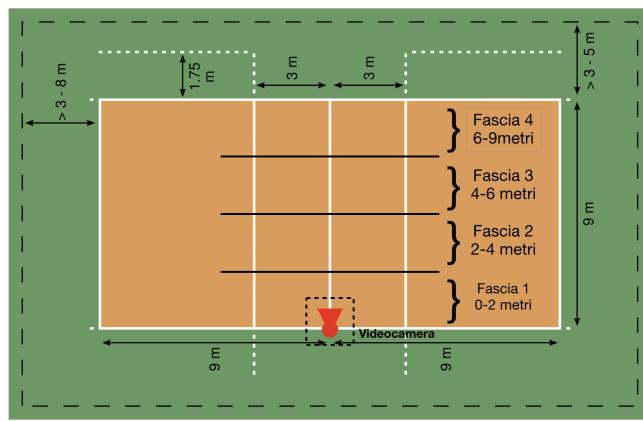


Figure 7: Divisione del campo per calcolo della profondita' e della velocita'

Listing 3: Calcolo della Velocita'

```

1 focalLength = (w_pixel * KNOWN_DISTANCE) / KNOWN_WIDTH
2 ...
3 #d_m = ((knownWidth * focalLength) / perWidth) perWidth percieved distance
4 d_m = distance_to_camera(KNOWNWIDTH, focalLength, float(w))
5 ...
6 if (d_m >= 0 and d_m < 2):
7     meter=1
8     v_x = (space_x * meter) / delta_frame
9     v_y = (space_y * meter) / delta_frame
10    v = (sqrt((v_x ** 2) + (v_y ** 2)))
11 ...
12 elif (d_m >= 2 and d_m < 4):
13 ...
14 elif (d_m >= 4 and d_m < 6):
15 ...
16 elif (d_m >= 6 and d_m <=9):
17 ...
18 elif (d_m > 8):
19     meter = 0.05
20     v_x = (space_x * meter) / delta_frame
21     v_y = (space_y * meter) / delta_frame
22     v = (sqrt((v_x ** 2) + (v_y ** 2)))

```

5 Risultati e Conclusioni

Tramite questo progetto, con una semplice videocamera ad un frame rate di 60-120 fps, sono riuscita ad individuare la palla applicando le tecniche di MOG-Background Subtraction e contour detection. In seguito sono riuscita a tracciare il suo percorso e individuare la sua velocità istantanea, tramite il calcolo dei pixel percorsi dal centroide della palla durante un certo numero di frame. Per individuare quando la palla sfiora la rete, ho sfruttato il rumore generato da essa e l'ho interpretato come un segnale da riportare in caso di collisione.

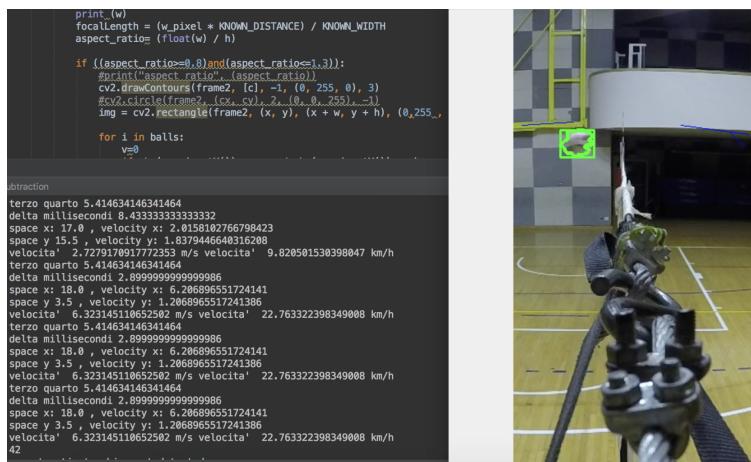


Figure 8: Tracking e Speed Detection

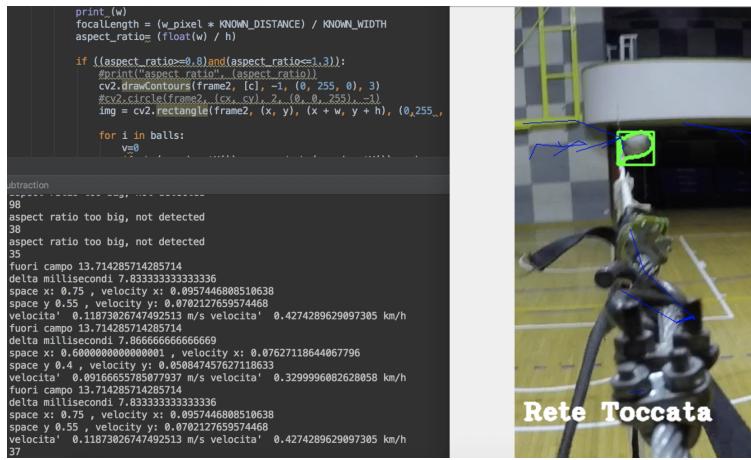


Figure 9: Collisione con la Rete

Tuttavia vi sono ancora delle problematiche relative alla fase di detection della palla e al calcolo della sua velocita'. Nel primo caso, durante la fase di detection, quando le giocatrici entrano nella ROI, oltre al blob della palla potrebbe venir identificato anche il blob relativo al busto o alla testa delle giocatrici. Questo problema viene in parte risolto limitando le aree e l'aspect ratio a quelle relative alla palla. Nel secondo caso, calcolo della velocita' istantanea, la velocita' non e' sempre precisa. Questo problema e' causato da una mancanza di una calibrazione precisa. Nel mio caso le imprecisioni sono date dal calcolo della profondita' che viene fatto stimando la posizione della palla all'inizio del processo di tracking. Inoltre, per migliorare il calcolo della velocita', bisognerebbe tenere conto delle leggi fisiche che regolano il moto della palla.

Per maggiori dettagli sul progetto si possono visionare i seguenti link:

- GitHub: https://github.com/FataHoxha/CV_project
- GoogleDrive: <https://goo.gl/4MSMvu>

References

- [1] Daniel Link Bjoern Eskofier Gabriel Gomez, Patricia Herrera López. Tracking of ball and players in beach volleyball videos. <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0111730>, 2014.
- [2] OpenCV. Camera calibration with opencv, 2012.
- [3] KeskarKishor M. Bhurchandi Paresh R. Kamble, Avinash G. Ball tracking in sports: a survey. <https://link.springer.com/article/10.1007/s10462-017-9582-2>, October 2017.
- [4] Adrian Rosebrock. Find distance from camera to object/marker using python and opencv, 2015.
- [5] Hyoung Joong Kim Yo-Sung Ho. A scheme for ball detection and tracking in broadcast soccer video. https://link.springer.com/chapter/10.1007/11581772_76, 2005.