Computer Vision-Driven Ultimate Frisbee Tracking and Analytics

Caroline McKee and Hitha Revalla
Approach/Methodology

Goal: To implement a pipeline to process high-definition ultimate frisbee footage in order to track the locations of both the frisbee and players, with the ultimate goal of automatically computing the number of passes which occur.

Figure 1. Snapshots showing the motion of a typical frisbee pass. In our simulated footage, only one team will be present and players will not be wearing white in order to maintain adequate color contrast with the frisbee.
Preliminary Results
Phase 1: MATLAB Image Processing Pipeline

Preprocessing

Approach #1: Morphological-based Tracking

Approach #2: Template-based Tracking

Approach #3: Background subtraction-based Tracking

Color-based Player tracking/region labeling

Calculate number of passes
Preprocessing

For each frame:

- Adaptive histogram equalization with 16x16 tiles (adapthisteq)
  - Corrects for differences in lighting across image and presence of shadows
- Sharpen frame (imsharpen)
Morphological Based-Tracking

- Thresholding → frisbee must be white
- 3 structuring elements: horizontal line, horizontal line +/- 15 degrees
- Erosion, then dilation
- From binary image: region labeling

**Problem:** White regions on player clothes and other places in image

**Solution:**
- Clothes regions will appear within the eventual player regions, so we know they can’t be the frisbee
- Regions in other parts of image are stationary, constrain such that we can isolate which region is frisbee based on its motion
Morphological-Based Processing Output
Frisbee Centroid Tracking

- Centroid of binary frisbee region used as a proxy for frisbee location
- Used later to calculate when frisbee is caught and thrown
Template-Based Tracking

- Selection of various templates through frozen frame screenshots.
- Highly dependent on the quality of the template.
- Sharpening the Frames helped to an extent.

Problem:

- Template worked for only the throw in which it was captured.
- Templates had a high correlation to the white regions in the jersey.

Solution: Ran Template matching with 4 templates randomly chosen across the footage.
Template-Based Tracking Output

Output shows the tracking for a select template from the “Throw”
Background Subtraction-Based Tracking

- Choice of a relatively constant background frame, namely frame 1, where frisbee is absent
- Subtraction of the constant frame and Post-processing by erosion and closing.

**Problem:** Useful only when the players are relatively stationary and only frisbee moves, also picks up white from jersey that is initially invisible in frame 1.

**Solution:**

- Dynamically keep track of the background, subtract the prior frame, thereby tracking only the object that moves.
- Regions in other parts of image are stationary, constrain such that we can isolate which region is frisbee based on its motion
Background Subtraction-Based Tracking Output
3-Player Footage
3-Person Result and Problems

Applying the same techniques as in the case of 2 person tracking

- **Morphological Based-Tracking:**
  - Erosion and Dilation by 4 structuring elements: horizontal line, horizontal line +/- 15 degrees, horizontal line +45 degrees
  - The same problem with the white regions on player clothes and other places in image still persists.

- **Background Subtraction:**
  - Same as in the case of 2 people tracking. Also included dilation to improve visibility of long shots.

- **Template Matching:**
  - Results are far worse than Morphological Tracking and Background Subtraction.
Morphological-Based Processing Output
Background Subtraction-Based Tracking Output
Color-Based Player Tracking

- Both players in test video wore red: shirts were segmented from the image using RGB slicing to create binary mask for each frame
- Small region removal to isolate shirt regions
- Centroid calculated

**Problem:** Shirt became two regions when arm crossed over body

**Solution:** Take average of two shirt region centroids

- Player region box/region created: size determined empirically based on footage scale
Player Extraction - 2 players
Player Extraction - 3 players
Player Regions (For Passes Calculation)
Player Regions (For Passes Calculation)
Player Centroid Tracking

- Used to draw a rectangular box around player
- Labeled box region = “player region”

Potential Problems: (1) Region is not exact, (2) Frisbee crosses in front of player, but not actually a pass to that player
Counting Passes

- Two approaches:
  1. Determine number of times frisbee centroid enters player region and does not exist the other side
  2. Determine number of times there are \( (\text{number of players} + 1) \) regions present: not as viable for large scale games because the number of players in the frame may change + more background noise

- Result: Program successfully counted the 17 passes which occur using (2), registered extra passes using (1) due to wind-ups leaving player region
Next Steps

- Create MAP detector for player tracking using test images
- Apply processing pipeline to footage in which players are moving
- Apply processing pipeline to real game footage
- Test efficacy of Python OpenCV trackers in comparison to our pipeline
References


OpenCV: cv::TrackerCSRT Class Reference. (n.d.). Retrieved February 22, 2019, from https://docs.opencv.org/3.4.2/d2/da2/classcv_1_1TrackerCSRT.html#details

OpenCV: cv::TrackerKCF Class Reference. (n.d.). Retrieved February 22, 2019, from https://docs.opencv.org/3.4/d2/dff/classcv_1_1TrackerKCF.html

OpenCV: cv::TrackerMOSSE Class Reference. (n.d.). Retrieved February 22, 2019, from https://docs.opencv.org/3.4.3/d0/d02/classcv_1_1TrackerMOSSE.html
