

## **Video-Based Detection of Hurdle Crossings by Athletes to Gather Intermediate Race Timings**

Aela Le Sommer<sup>1</sup>, Jelle Vanhaeverbeke<sup>1</sup>, Robbe Decorte<sup>1</sup>,  
Maarten Slembrouck<sup>1</sup>, Steven Verstockt<sup>1</sup>

<sup>1</sup>Ghent University – imec, IDLab, Sports Data Science, [steven.verstockt@ugent.be](mailto:steven.verstockt@ugent.be)

### **Introduction**

The 400-meter hurdles is an athletics event where athletes stay within their designated lanes, running 400 meters while clearing ten evenly spaced hurdles. To evaluate the speed of an athlete between two consecutive hurdles or to evaluate the time difference between two athletes at a hurdle, this work proposes a methodology to automatically detect the hurdle crossings. Whether by biomechanical analysis [1,2] or by using video-based pose estimation algorithms [3], the technique of crossing hurdles has been studied in literature. However, methods to extract hurdle crossing timings in an automatic way seem not to have been investigated yet. The proposed method is based on: the detection of athletes and hurdles, the extraction of the lanes from the athletic track and hurdle crossing detection methods based on the athlete, hurdle and lane relationships.

### **Proposed Methodology for Hurdle Crossing Detection**

#### Datasets

For the detection of athletes and hurdles, a dataset has been constructed of several YouTube videos of 400-meter hurdles races as well as videos that we recorded during the 2023 IFAM athletics meeting in Flanders, Belgium. The final dataset consists of 9921 annotated images. Next to that, a second dataset is constructed based on the same videos consisting of annotated track masks which will help with the track lane detection. This dataset contains 768 frames. From both datasets, 80% is used for training and 20% for validation of the models.

In addition to the real-world video data, a script to generate synthetic athletic videos has also been constructed. This Unity script allows us to further optimize and test the models without the need for real-world data acquisition, i.e., additional annotated data can be generated at low-cost and context (lighting, weather, crowd, etc.) can be easily adjusted. It also allows us to objectively detect the camera positioning/viewpoint that can best recognize the hurdle crossings. This could be taken into account in the video recording of future races. The current Unity videos are already rather realistic, however further optimizations can probably be done to make them even more similar to real-world footage. User experience testing will also be very valuable when further optimizing the Unity models/scripts.

#### Athlete and Hurdle Detection

For the detection of athletes and hurdles, a YOLOv8 object detection model is trained. This resulted in a recall of 91.61% and precision of 94.25% for the validation datasets. We noticed that the detection of athletes and hurdles is more stable on the last part of the race since the viewpoint the first 300 meters is much more varied than the last 100 meters. However, by generating additional data for the first 300 meters using the Unity video generation tool, we

hope to solve this problem. To keep track of an athlete as he/she moves over the video frames, the YOLOv8's default tracker, i.e., BoT-SORT, is used [4].

### Lane Detection

Now that the position of an athlete can be followed through a video, we want to be able to detect when the athlete crosses a hurdle. First, the athletics track is segmented using a SegNet semantic segmentation model [5]. This model obtains an F1 score of 95.21%. This model allows us to create a region of interest on the track on which then a Canny edge detector is applied, followed by the probabilistic Hough transform to detect straight lines. Once all the lines are detected, they are filtered in order to only keep the lines delimiting the different lanes.

To determine which hurdles are in the same lane as an athlete, the bottom center coordinates of each bounding box are analyzed. However, on some frames the height of an athlete's bounding box may be smaller than on other frames due to occlusion by other athletes or due to jumping of the athlete himself. To solve this, we have introduced a coefficient to increase the height of the bounding box of the athlete if it is strictly less than 90% of the average height of the bounding box of this athlete on the surrounding frames.

### Hurdle Crossing Detection

To detect when an athlete crosses a hurdle, the following three methods are investigated:

1. Intersection between the hurdle bounding box and the athlete bounding box,
2. Euclidean distance between bottom center point of hurdle and athlete bounding box,
3. Relative position of x-coordinates of center points of hurdle and athlete bounding box.

## **Results**

As the jump of a hurdle by an athlete is visible during several consecutive frames, we estimate that the detection is a success for any absolute error less than or equal to 5 frames. The proposed hurdle crossing detection currently achieves a recall of 83.75% for methods 1 and 2 and 77.5% for method 3. We also obtained a precision of 81.71% for method 1, 84.81% for method 2 and 93.05% for method 3. Future work will investigate if combining all methods together (and different methods on how to do that) positively impacts the accuracy.

## **References**

- [1] M. Čoh, N. Bončina, S. Štuhec and K. Mackala (2020) Comparative Biomechanical Analysis of the Hurdle Clearance Technique of Colin Jackson and Dayron Robles: Key Studies
- [2] A. Singh Sidhu and M. Singh (2015) Kinematical Analysis of Hurdle Clearance Technique in 110M Hurdle Race
- [3] P. Jafarzadeh, P. Virjonen, P. Nevalainen, F. Farahnakian, J. Heikkonen (2021), Pose Estimation of Hurdles Athletes using OpenPose
- [4] Multi-Object Tracking with Ultralytics YOLO, <https://docs.ultralytics.com/modes/track/>
- [5] V. Badrinarayanan, A. Kendall and R. Cipolla (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation