

Computervisiegebaseerde detectie van drafting tijdens het fietsen

Simon Van den Bossche

Studentennummer: 01503365

Promotoren: prof. dr. ir. Peter Veelaert, prof. dr. ir. Wilfried Philips
Begeleider: Gianni Allebosch

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: elektronica-ICT

Academiejaar 2018-2019

Computervisiegebaseerde detectie van drafting tijdens het fietsen

Simon Van den Bossche

Studentennummer: 01503365

Promotoren: prof. dr. ir. Peter Veelaert, prof. dr. ir. Wilfried Philips
Begeleider: Gianni Allebosch

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: elektronica-ICT

Academiejaar 2018-2019

Wereldkampioenschap Ironman triatlon 2018 in Hawaï



“Op geen enkele manier was ik aan het stayeren of had ik gestayerd. Ben ik een wieltjeszuiger?”

— Frederik Van Lierde

Toelating tot bruikleen

‘De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.’

‘The author gives permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the limitations of the copyright have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.’

Simon Van den Bossche - juni 2019

Woord vooraf

Bij aanvang van deze masterproef scriptie wil ik eerst enkele mensen bedanken.

Vooreerst mijn oprechte dank aan mijn begeleider ing. Gianni Allebosch om mij de kans te geven dit onderzoek te kunnen volbrengen. Ik wil bovendien graag mijn promotoren prof. dr. ir. Peter Veelaert en prof. dr. ir. Wilfried Philips bedanken voor het ondersteunen van mijn masterproef aan de onderzoeksgroep Telin en voor hun kritische en constructieve feedback.

Graag een woord van dank aan alle sportliefhebbers die mijn enquête hebben ingevuld. Een speciaal dankwoord voor de triatleten van *GETT*, *SMO* en *Oil Service* die voor mij hebben gefietst tijdens verschillende triatlonwedstrijden om zo mijn dataset te kunnen uitbouwen. Ik wil ook graag mijn ouders bedanken voor de vele fietsritopnames die zij hebben gedaan.

Abstract

Computervisiegebaseerde detectie van drafting tijdens het fietsen

door Simon Van den Bossche

Masterproef voorgelegd voor het behalen van de graad master in de industriële wetenschappen: elektronica-ICT

Academiejaar: 2018 – 2019

Promotoren: prof. dr. ir. Peter Veelaert, prof. dr. Wilfried Philips

Begeleiders: Gianni Allebosch

Faculteit ingenieurswetenschappen en architectuur

Universiteit Gent

Vakgroep Telecommunicatie en Informatieverwerking

Voorzitter: prof. dr. ir. Joris Walraevens

De laatste jaren hebben verschillende sporten voordeel gehaald uit de introductie van slimme technologie. In de triatlon is er echter tot op vandaag nog geen uitgewerkt systeem op de markt dat de functionaliteit van draftingdetectie realiseert bij de fietsdiscipline. Bij draften wordt dicht achter iemand gefietst waardoor minder luchtweerstand wordt ondervonden. Volgens het triatlonreglement zijn de regels omtrent drafting overtreden indien er minimaal 20 seconden in een draftingzone van 10 meter bij 3 meter van de voorligger is verbleven.

Deze scriptie heeft als doel een concept voor een draftingdetectiesysteem op basis van computervisie te ontwikkelen. Na het detecteren en volgen van een fiets doorheen de verschillende scènes, wordt de afstand tot dit object bepaald. Er zijn twee methoden onderzocht om een afstandsbeplating te bewerkstelligen. Er wordt dusdanig na uitgebreid testen tot een gemiddelde afstandsfout van 0,47 meter gekomen in een gesimuleerde wedstrijsituatie.

Trefwoorden – triatlon, draften, stayeren, objectdetectie, objecttracking, afstandsbeplating

Computer vision based detection of drafting while cycling

Simon Van den Bossche

Supervisors: Gianni Allebosch, Peter Veelaert, Wilfried Philips

Abstract—Smart technology has been introduced in various sports in the last few years. In triathlon, however, there is still no system used that realises the functionality of drafting detection in the bicycle discipline. Drafting involves cycling so close behind someone such that wind resistance is significantly reduced. According to triathlon regulations, the rules concerning drafting are violated if at least 20 seconds have been spent in the drafting zone of 10 by 3 metres from the bicycle in front. In this paper a proof of concept for a drafting detection system based on computer vision is proposed. After detecting and tracking a bicycle through the various scenes, the distance to this object is determined. Next, two methods are discussed to achieve a distance value. Subsequently, these algorithms are tested using a static recording and a recording that simulates a race situation with ground truth distances from a LIDAR system.

Index Terms—triathlon, bicycle, object detection, object tracking, distance determination

I. INTRODUCTION

In many triathlon (and duathlon) events, it is not allowed to form a peloton, in contrast to regular road cycling. The lack of reliable detection of illegal drafting leads in some cases to arbitrariness in the best case, and possibly even to corruption, such as favouring some individuals in the worst case. Nowadays drafting is being checked by referees driving a motorbike. The purpose of this paper is to discuss a proof of concept that automates the task of granting drafting penalties. For this automation, the robustness of the system is very important in all possible triathlon situations such as both on flat or uneven roads and in different weather conditions.

The structure of this paper is as follows: In Section 2, the problem of drafting is discussed. Section 3 describes common situations in a triathlon race. Subsequently, Section 4 deals with the proposed concept of the drafting detection system. In Section 5 a number of experiments and results of the proposed method are presented. The paper concludes in Section 6 with a discussion about the performed work and results.

II. DRAFTING PROBLEM

Triathletes try to reach the finish as quickly as possible during races. In addition to optimising training, rest and nutrition, these athletes can greatly improve their performance by working on a streamlined posture. A huge advantage can also be gained by cycling closely behind another competitor

S. Van den Bossche is with the Telecommunications and Information Processing - Image Processing and Interpretation Department, Ghent University (UGent), Ghent, Belgium. E-mail: sidnboss.vandenbossche@ugent.be .

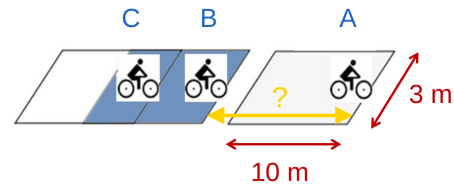


Fig. 1. Official drafting zone triathlon [1].

during a race, so this athlete acts as a wind shield. This technique is called drafting or slipstreaming.

According to the official rules, triathletes in non-drafting races may not stay longer than 20 seconds in the 10 by 3 metres drafting zone shown in Figure 1. The longitudinal distance is measured from front wheel to front wheel. According to this figure cyclist C is drafting behind B who, in turn, is not slipstreaming behind A. Person C may therefore be in the drafting zone of B, coloured in blue in the respective figure, for a maximum of 20 seconds to for example pass by. Once overtaken, cyclist B must be outside the drafting zone of C within 5 seconds in order to prevent a drafting penalty. However, monitoring these rules in practice is difficult so these are often violated. A first reason is that in a typical triathlon race there are too few referees available to check all triathletes anywhere, anytime. In addition, the referees' decisions are still subjective as they can only roughly estimate the distances between the bicycles. Secondly it is not always easy for the cyclist to properly estimate the distance to its predecessor. A survey with a sample size of 351 indicates that 45.10% sometimes draft unintentionally in a non-drafting race.

A drafting detection system based on video taken from a camera mounted under the saddle of a bicycle together with computer vision techniques can offer a solution for these problems. For this system, further intelligence can be implemented to account for certain situations such as natural accordion movements due to a bend, a slope etc. Moreover, according to the triathlon rules it is possible to challenge the penalty, whereafter the chief referee will clarify the time penalty with any supporting documents. Due to a drafting detection system based on video, the images and drafting score can be requested afterwards and used as evidence. In summary, it can be stated that, through continuous checking, this will in time lead to less drafting and consequently ensure a fairer race.

In a previous implementation of a drafting detection system,

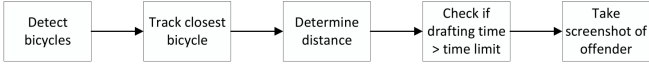


Fig. 2. Block diagram of the drafting detection system.

GPS and Web Services (WS) are used whereby an average accuracy of 1.32 metres is obtained for the absolute position of each athlete. By determining the relative distance between two cyclists, the error can therefore increase up to 2.64 metres or even higher in cases of Urban Canyons and tree lines. This distance deviation is unacceptable for the drafting detection application [2]. Two more accessible solutions for a drafting detection system are: Light Detection And Ranging (LIDAR) and Radio Detection And Ranging (RADAR). The former is an expensive technique, but is very accurate. The latter is more compact, cheaper and operates on the basis of radio waves instead of optical laser beams. However, for object detection with a RADAR system, sufficient reflection points are required, making this more difficult compared to video-based object detection.

In general the operation of the real-time drafting detection system based on a camera can be summarised in the block diagram of Figure 2. This diagram forms the basic structure of Section 4. The next section deals with different situations during a triathlon race.

III. TRIATHLON RACE SITUATIONS

During a triathlon race there are certain situations that can cause difficulties such as: distinguishing (fellow) cyclists from the background and the onlookers, dealing with vibrations and fluctuations in the image due to the typical movement of a bicycle, experiencing a possible non-flat road surface etc. These different circumstances are shown in Figure 3 and are part of a self-created data set taken during four non-drafting triathlon races in Belgium.

A monocular camera alignment is considered where the camera is mounted under the saddle of the bicycle and looking backwards. The camera can only be affected by the tilt angle in this mounting position. The possibility for the camera tilt to change intermediately (e.g. due to vibrations) must be made impossible or limited in a final product of this system, because this may have a negative effect on the result of the distance calculation, which is discussed in the following section.

IV. DRAFTING DETECTION

In this section the various parts of the drafting detection system are explained and analysed based on the block diagram in Figure 2.

A. Bicycle detection

A first step for the drafting detection system is to detect cyclists in real-time in images. The aim is to recognise cyclists up to 20 metres in order to have sufficient margin if, for example, the drafting distance is chosen at 15 metres (which is

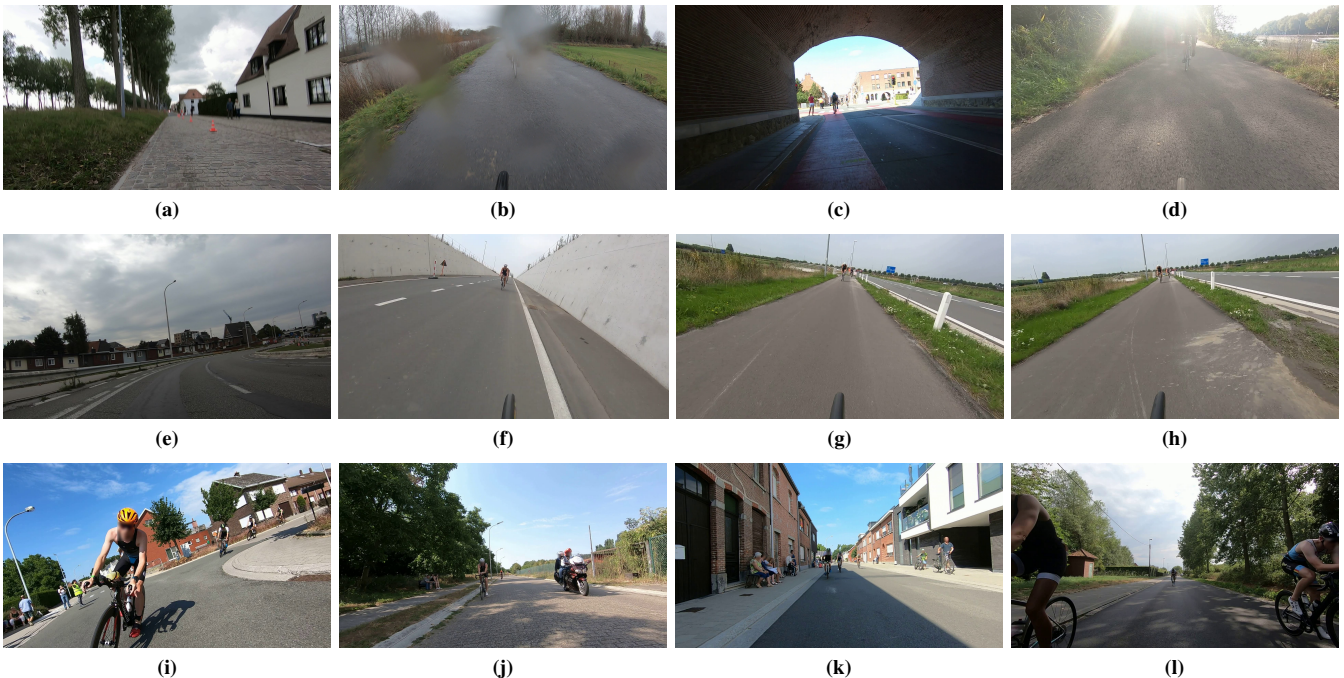


Fig. 3. Different circumstances in triathlon races: (a) introduction of motion blur, (b) water on the camera lens that leads to a blindness, (c) driving in a tunnel, (d) sun rays on the camera lens that leads to a short flash blindness of the camera, (e) an ascending and (f) a descending road, standing pedalling which leads to (g) a left swing and (h) a right swing, (i) taking a turn, interference (j) by a motorcycle or (k) by onlookers, (l) three common situations at once (passing by, drafting and being passed by).

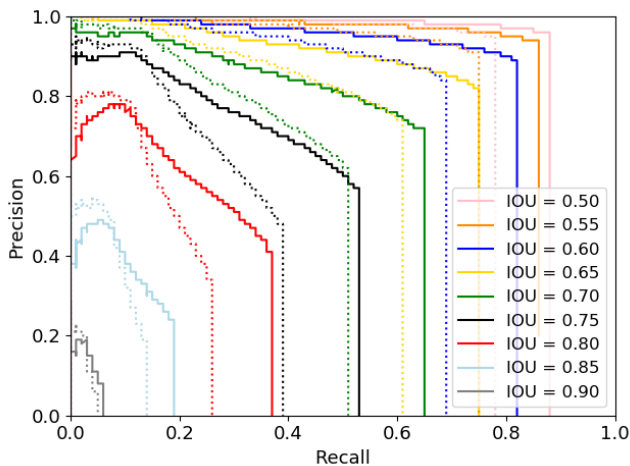


Fig. 4. Precision recall of the trained YOLOv3 (solid lines) and Tiny YOLOv3 (dotted lines) network for different IOU values.

applied by some race organisers). To achieve this, a YOLOv3 and Tiny YOLOv3 network have been trained for triathlon bikes [3]. The detection speed on the test set is 24.57 fps and 42.25 fps, respectively. The data set is recorded at 30 fps but a frame throughput of 20 fps is still sufficient. Figure 4 shows the Average Precision (AP) for the ‘bicycle’ class for different Intersection Over Union (IOU) values for the YOLOv3 (continuous lines) and Tiny YOLOv3 (dotted lines) network. Because of the importance of real-time detection, the trained Tiny YOLOv3 network is used as a bicycle detector, because other processing steps are still required after object detection, as discussed further in this section.

B. Bicycle tracking

Since no object detector is perfect and consequently false positives and false negatives occur, the method discussed in this second paragraph attempts to track the closest detected bike over time by using an object tracker. By means of object tracking the motion trajectory of an object is tracked and recorded. In order to associate bicycles through the scenes, track management is discussed to solve the correspondence problem.

One of the reasons for using an object tracker is to lower the miss rate. By realising continuous monitoring, the estimated position of the closest cyclist through the various images is improved. Visual object tracking is used [4] in cases where the object detector does not detect a bicycle, while in reality there is one present in the current scene: the False Negatives (FN). More specifically, the principle of tracking-by-detection of the bounding box is applied. If a bicycle is detected in the previous frame ($t - 1$) but not in the current frame (t), this bicycle in frame t is estimated by tracking the last detected one. Thus, the effect of missed detections is mitigated and bicycles in the scene can be better monitored continuously. This will result in a better track management and more samples to determine the distance, which will be discussed in the next paragraph.

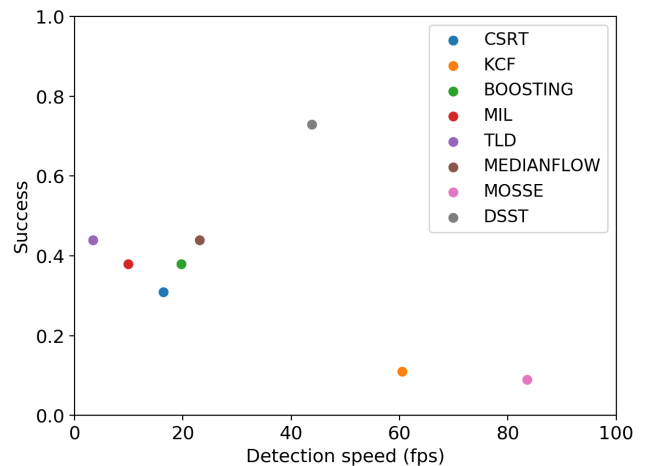


Fig. 5. Success rate versus detection speed.

In order to make a responsible choice with regard to the object tracker, eight object trackers from the literature are benchmarked according to detection speed and success rate as shown in Figure 5: CSRT [5], KCF [6], Boosting [7], MIL [8], TLD [9], Medianflow [10], MOSSE [11] and DSST [12]. With regard to this benchmark, it appears that the Discriminative Scale Space Tracking (DSST) object tracker is the most appropriate solution. This tracker uses a multi-scale pyramid to estimate the scale of an object. This makes it possible to track objects that change with regard to translation (for example, a cyclist who moves to the left in the image) and scaling (a cyclist who comes closer).

Both the results of the YOLOv3 network and Tiny YOLOv3 are tested in combination with the DSST object tracker for their improvement in the detection rate. An initial AP of 77.19% for Tiny YOLOv3 was obtained, but after applying object tracking an increase of the recall ensures an average precision of 88.27%. Afterwards track management is applied to follow the detected cyclist through the different video frames. In this way, the same bicycle can be uniquely identified by its ‘bicycle id’. This is decisive for the drafting detection system because the time a particular cyclist is in the drafting zone must be measured.

C. Distance calculation

In this section two methods are proposed in order to determine the distance to the bounding box of a bicycle. Since the camera is mounted at the rear of the saddle and according to the rules the drafting distance is calculated from front wheel to front wheel, there exists a difference which must be accounted for and corrected.

1) *Height method*: A first method assumes the height of the camera on the ‘camera bicycle’ to be equal to the height of the handlebars of the bicycle behind. These heights, respectively h_1 and h_2 , are shown in Figure 6. In this representation, f and y respectively correspond to the focal length of the camera and the height of the bounding box on the image sensor. In this

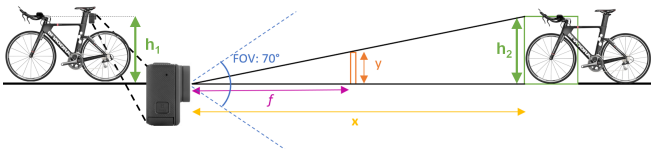


Fig. 6. Height method principle for distance determination*.

way a relationship can be established to determine the distance x as indicated in formula (1).

$$\frac{f}{x} = \frac{y}{h_2} \Leftrightarrow x = f \cdot \frac{h_2}{y} \quad (1)$$

The disadvantage of this method lies in the fact that there can be a difference between the camera height and the handlebar height, in practice up to a maximum of 15 cm, which depends on the type of bike on the one hand and the athlete's wishes for an aerodynamic posture on the other hand. The advantages of this method are an easy calculation and the influence of the tilt angle of the camera can be neglected. As a consequence the location of the bounding box relative to the horizon is not important, which is not the case for the method described in the next paragraph.

2) *Ground plane method*: This approach relates the location of the bottom of the bounding box to the ground plane, starting from a known tilt angle and camera height h_1 . More specifically, the bottom of the bounding box is expressed as a corner α , whereby a tilt angle θ_{tilt} is also taken into account. The vertical horizon location changes depending on this tilt angle. Figure 7 shows the principle of this methodology. In order to determine the distance x a relationship can be prepared to achieve a distance formula (2).

$$x = \frac{h_1}{\tan(\alpha + \theta_{tilt})} \quad \text{with} \quad \alpha = \tan^{-1}\left(\frac{y}{f}\right) \quad (2)$$

If the camera is perfectly aligned, the vertical position of the horizon corresponds with the height of the camera centre (c_x , c_y). The disadvantage of this method is the dependence on the tilt angle of the camera. First, the initial tilt angle must be known for each setup separately. Moreover, there could be situations where the camera is not perfectly aligned due to a production error or a difference in the slope of the road as indicated in Figure 3 (e, f). In addition to the current recording option, the camera can twist due to vibrations it experiences.

If the absolute error on the vertical position of the bounding box is 5 pixels, a deviation of 0.29 degrees for the tilt angle

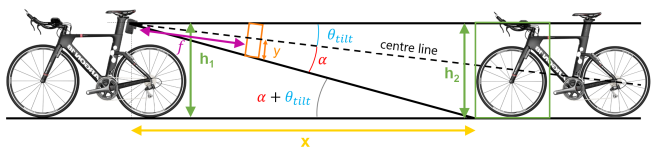


Fig. 7. Ground plane method principle for distance determination*.

*Figure is modified from [13].

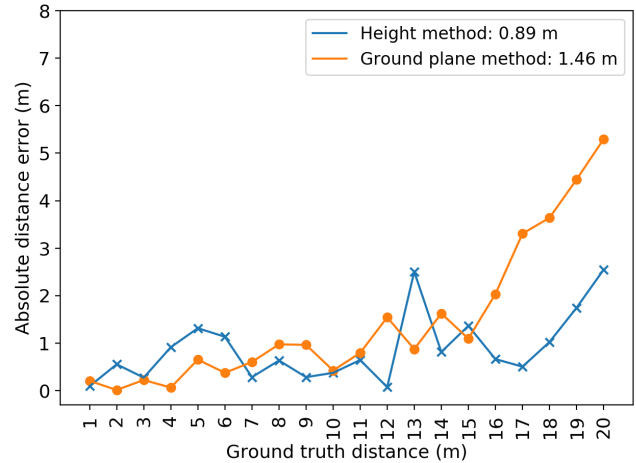


Fig. 8. Static test - the absolute distance error for the height method and ground plane method. Over all the ground truth distances, the former has an average absolute error of 0.89 metres whereas the latter has an average absolute error of 1.46 metres in the case of a perfect tilt angle.

of the camera is encountered according to formula (2) with a focal length of 990 pixels. In practice, this deviation will be higher due to the placement of the camera. A further evaluation and testing of both methods for distance determination will be discussed extensively in the following section.

V. EVALUATION AND RESULTS

In the previous section two strategies were proposed for determination of the distance from the camera to a cyclist: the height method and the ground plane method. Both methods are analysed and benchmarked in-depth in order to make a well-founded choice concerning robustness and accuracy. To achieve this, two types of test recordings have been made and are discussed: static and dynamic situation test.

A. Static situation test

To determine the accuracy of the calculated distance, a static situation for well-known reference distances is considered. The bicycle with the camera is fixed at the 0 m mark. Another bicycle is placed at distances from 1 m to 20 m to the camera, at regular intervals of 1 m each. For this test an average of the measured distances over 100 frames is taken for each ground truth distance. The results of the height method and the ground plane method are shown in Figure 8. From this graph it can be deduced that the ground plane method obtains better overall results for distances less than or equal to 6 metres. In the other cases the height method gives a lower overall absolute distance error. However, these measurements have been realised with a known ideal tilt angle, therefore the tilt angle deviation is assumed to be zero. If the tilt angle differs by 1 degree, an average absolute distance error over all ground truth distances of 2.54 metres is perceived for the ground plane method.

This benchmark gives a good indication of the deviation on the measured distance of the drafting detection system. This test indicates that the height method should be favoured to the

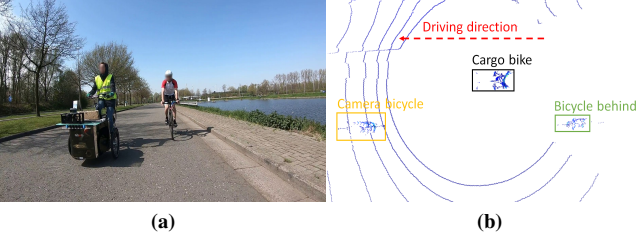


Fig. 9. LIDAR setup on the cargo bike for a dynamic distance measurement: (a) view from the camera bicycle and (b) top view. The ‘Camera bicycle’ is the bike with the camera mounted under the saddle. ‘Bicycle behind’ is the triathlete who is cycling behind the camera bicycle. The driving direction of the three cyclists is indicated with an arrow.

ground plane method. On the one hand it gives better overall results and on the other hand, from a practical point of view, it is independent of the tilt angle. The disadvantage of this static situation test is that this situation is different from what happens during a triathlon race. Moreover, an evaluation for the zone of approximately 8 to 12 metres is important as the drafting rule prescribes 10 metres as a limit. In addition, the reference distances form a limitation, since jumps of one metre are made each sequence. To avoid these limitations, a dynamic situation test is discussed in the following paragraph.

B. Dynamic situation test

To test the drafting detection algorithm in a more realistic environment, a LIDAR system is used on a cargo bike to simulate a triathlon race situation. A flat road with no gradients is considered in this test setup. The LIDAR of the *Velodyne VLP 16 type* provides a 3D point cloud of the environment through which the measured distance of the drafting detection system can be compared with the LIDAR distance with an accuracy of 0.10 metres. The arrangement is shown in Figure 9. Figure 10 shows the absolute distance error calculated for 400 ground truth LIDAR distances for both the height method and the ground plane method. These distances vary from approximately 3 to 19 metres. The graph demonstrates that the height method is more accurate than the ground plane method with average absolute errors of 0.47 metres and 1.16 metres respectively. In the case that a tilt angle deviation of 1 degree is chosen, the average distance becomes 1.96 metres. In other words, for the dynamic situation test the height method appears to be the preferred choice as well.

The following paragraph deals with the detection of drafting for multiple measurements over time. More specifically, a probability is determined that the subject is drafting over a 20 second period, as this is the maximal time a triathlete can remain in the drafting zone.

C. Drafting detection score over a 20 second period

To determine the final score of the drafting detection system, the probability, given in formula (3), is calculated.

$$P(\text{drafting rule is violated} | \text{cyclist in drafting zone over a certain time interval}) \quad (3)$$

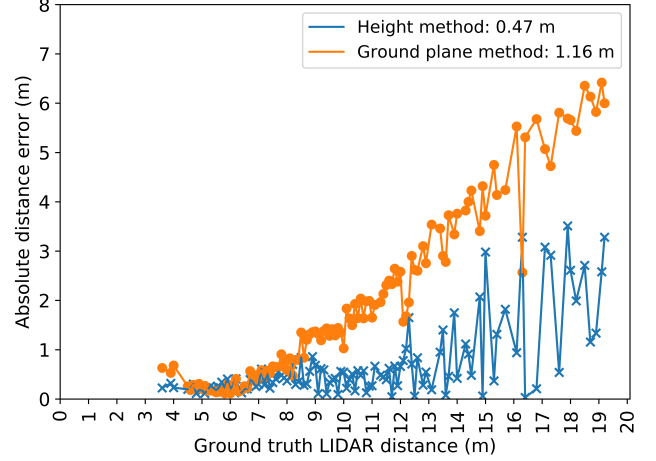


Fig. 10. Dynamic test - the absolute distance error for the height method and ground plane method. Over all the ground truth distances, the former has an average absolute error of 0.47 metres whereas the latter has an average absolute error of 1.16 metres in the case of a perfect tilt angle.

The error function (erf) can be used to determine the probability that the real distance a is less than 10 metres and the athlete is therefore in the drafting zone, is indicated in formula (4), where σ corresponds with the standard deviation on the distance: 0.58 metre.

$$P(a, \sigma) = \frac{\sigma - \sigma \operatorname{erf}\left(\frac{-10 + a}{\sqrt{2} \sigma}\right)}{2 \sigma} \quad (4)$$

This probability gives rise to a binomial probability distribution but there is a correlation between the measurements. In particular, breaking the drafting rule corresponds to staying at least 20 consecutive seconds in the drafting zone. In the proposed implementation, the probability of violating the drafting rule at a given instant is determined every 2 seconds, over a total period of 30 seconds. The presence in the drafting zone will be represented by so-called drafting patterns, each with a length of 15 units. These patterns are made from all combinations of 15 bits of which at least 11 consecutive bits must be ‘1’ to be considered a valid pattern. In this way, 104 possible drafting patterns can be created. The first five patterns are shown in formula (5).

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

The first pattern indicates that the athlete was in the drafting zone for the last 11 samples, which is not the case for the first 4 instances. This corresponds to a valid pattern, where the drafting rule is violated. An invalid pattern can be: [010101011111000].

In order to determine the probability the drafting rule has been violated, a series of 15 sampled distances is considered (e.g.: 9.39, 9.14, 9.98, 9.62, 9.93, 9.19, 8.99, 8.79, 8.61, 8.13,

8.06, 7.91, 7.64, 7.56, 7.39). The probability that the first drafting pattern occurs, is 0.059%. In addition, the pattern [1111111111111111] corresponds to the greatest probability, in particular 14.25 %. The drafting rule is violated if at least one of the drafting patterns occurs. Since these drafting patterns are mutually exclusive, the probabilities of all patterns can be added to find the final probability that at least one of these, no matter which, has occurred. In the case of the given example, this corresponds to a final probability of 88.36 %. The drafting rule has therefore been violated if the total probability, given all possible drafting patterns, is greater than 0.5.

The results for some situations are shown in Table I.

TABLE I
SAMPLED DISTANCES OVER A 30 SECOND PERIOD WITH THE
CORRESPONDING PROBABILITY THAT THE DRAFTING RULE HAS BEEN
VIOLATED.

Sampled distances (m)	Drafting score
10.84, 10.67, 10.98, 10.48, 10.83, 10.42, 9.59, 9.07, 8.56, 8.45, 7.83, 7.69, 7.88, 7.87, 7.97	0.0291
10.67, 10.98, 10.48, 10.83, 10.42, 9.59, 9.07, 8.56, 8.45, 7.83, 7.69, 7.88, 7.87, 7.97, 8.30	0.2352
10.98, 10.48, 10.83, 10.42, 9.59, 9.07, 8.56, 8.45, 7.83, 7.69, 7.88, 7.87, 7.97, 8.3, 9.39	0.6394
10.48, 10.83, 10.42, 9.59, 9.07, 8.56, 8.45, 7.83, 7.69, 7.88, 7.87, 7.97, 8.30, 9.39, 9.14	0.8170
10.83, 10.42, 9.59, 9.07, 8.56, 8.45, 7.83, 7.69, 7.88, 7.87, 7.97, 8.30, 9.39, 9.14, 9.98	0.8303
...	...
9.98, 9.62, 9.93, 9.19, 8.99, 8.79, 8.61, 8.13, 8.06, 7.91, 7.64, 7.56, 7.39, 7.72, 9.05	0.9565

If a cyclist is in the drafting zone for more than 20 seconds or longer than 5 seconds if the person has passed, the drafting rule has been violated and a drafting penalty should be awarded. The current implementation saves a screenshot of the offender, but other concepts are possible.

VI. CONCLUSION

In this paper a proof of concept for a drafting detection system in triathlon is proposed. The drafting concept and different situations during a triathlon race are discussed in the second section. The system is composed of three important building blocks: object detection, object tracking and distance determination. Detecting and then following the closest cyclist through the different scenes ensures a continuous monitoring in time. In this way an average precision, AP, 88.27% for bicycle detection is obtained on the test set. The height method appears to be the most accurate distance determination method, as the ground level method is too sensitive to potential tilt angle changes or poor estimation. In the static situation test the average absolute error over all ground truth distances is 0.89 metres and 1.46 metres for the height method and the ground

plane method respectively. But in practice this error will be bigger due to the difference in deviation for the theoretical tilt angle. To approach a more realistic race situation, a dynamic test was conducted with LIDAR distances as the ground truth. Over all ground truth distances, this experiment shows that the height method has an average absolute error of 0.47 metres. For the ground plane method, this error is 1.16 metres when a perfect tilt angle is assumed and 1.96 metres in the other case. Compared with an earlier design from the literature based on GPS and mobile networks where the error can increase from 1.32 metres to 2.64 metres, an improvement of up to 82.20% has therefore been achieved. In addition, a drafting detection score is determined which involves the probability over a 30 second period whether the drafting rule has been violated in a distance calculation every 2 seconds. In particular this rule is violated if an athlete has been in the drafting zone of 10 by 3 metres for more than 20 seconds.

The drafting detection system with an average absolute distance error of 0.47 metres which works in different situations can definitely enter triathlon and duathlon races in order to obtain fairer race course.

REFERENCES

- [1] B. T. & D. Federatie, "BTDF Sportreglementen 2018", 2018.
- [2] I. Fister, D. Fister, S. Fong, and I. Fister, "Widespread mobile devices in applications for real-time drafting detection in triathlons", *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 3, pp. 310–321, 2013.
- [3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", 2018. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [4] Seung-Hwan Bae and Kuk-Jin Yoon, "Robust Online Multiobject Tracking With Data Association and Track Management", *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2820–2833, jul 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6807759/>
- [5] A. Lukežič, T. Vojči, L. ČehovinZajc, J. Matas, and M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability", *International Journal of Computer Vision*, vol. 126, no. 7, pp. 671–688, 2018.
- [6] João F. Henriques, Caseiro Rui, Martins Pedro, and Batista Horge, "High-Speed Tracking with Kernelized Correlation Filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. [Online]. Available: http://www.robots.ox.ac.uk/~joao/publications/henriques_tpami2015.pdf
- [7] R. E. Schapire, "Explaining AdaBoost". [Online]. Available: <http://rob.schapire.net/papers/explaining-adaboost.pdf>
- [8] B. Babenko, S. Belongie, and M. H. Yang, "Visual tracking with online multiple instance learning", in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, vol. 2009 IEEE. IEEE, jun 2009, pp. 983–990. [Online]. Available: <http://ieeexplore.ieee.org/document/5206737/>
- [9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, jul 2012. [Online]. Available: <http://ieeexplore.ieee.org/document/6104061/>
- [10] —, "Forward-backward error: Automatic detection of tracking failures", in *Proceedings - International Conference on Pattern Recognition*. IEEE, aug 2010, pp. 2756–2759. [Online]. Available: <http://ieeexplore.ieee.org/document/5596017/>
- [11] Y. Lu, Z. Zhou, and J. Zhao, "Visual Object Tracking Using PCA Correlation Filters", 2018. [Online]. Available: http://www.cs.colostate.edu/~vision/publications/bolme_cvpr10.pdf
- [12] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative Scale Space Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017. [Online]. Available: <https://arxiv.org/pdf/1609.06141.pdf>
- [13] R&A Cycles, "2018 Argon 18 E-117 Tri Ultegra Bike". [Online]. Available: <https://www.racycles.com/product/detail/14750>

Inhoudsopgave

Lijst met figuren	xv
Lijst met tabellen	xix
Lijst met afkortingen	xx
1 Inleiding	1
2 Probleembeschrijving	2
3 Fotogrammetrie en cameramodel	10
3.1 Fotogrammetrie	10
3.2 Cameramodel	11
3.2.1 Extrinsieke parameters	12
3.2.2 Intrinsieke parameters	12
3.2.3 Lensdistorsie	12
3.3 Camerakalibratiemethoden	13
3.3.1 Fotogrammetrische kalibratie	13
3.3.2 Zelfkalibratie	13
4 Cameraopnames en wedstrijdsituaties	15
4.1 Cameraopstelling	15
4.2 Wedstrijdsituaties	18
5 Objectdetectie	21
5.1 Objectdetectie - literatuur	21
5.1.1 Convolutioneel neuraal netwerk	22
5.1.2 Objectdetectiealgoritmen	23
5.2 Benchmarking van fietsdetectienetwerken	29
5.3 Benchmarking fietsobjectdetectie na training	33
5.3.1 Precision recall curves	36
6 Objecttracking en trackbeheer	39
6.1 Motivatie voor objecttracking	39
6.1.1 Objecttrackervereisten	41
6.2 Tracking van objecten - literatuur	42
6.2.1 Uitdagingen objecttracking	42
6.2.2 Feature selectie voor tracking	43
6.2.3 Methoden en technieken om een bewegend object te volgen	44
6.2.4 Associatie	46
6.3 Objecttracker	48

6.3.1	<i>Successplots</i>	49
6.3.2	Precisieplots	50
6.3.3	Successrate versus detectiesnelheid	50
6.3.4	Precision voor fietsdetectie	52
6.4	Trackbeheer	54
7	Afstandsbepaling	57
7.1	Berekenen van de afstand	57
7.1.1	Hoogtemethode	58
7.1.2	Grondvlakmethode	59
7.2	Tilthoekverandering	62
7.3	Filteren van de afstandsberekeningen	66
8	Testing en resultaten	69
8.1	Nauwkeurigheid afstandsberekening	69
8.1.1	Statische situatietest	69
8.1.2	Dynamische situatietest	72
8.2	Draftingdetectiescore voor 20 seconden	77
8.3	Het nemen van een screenshot	81
8.4	Robuustheid testing	81
8.4.1	Cameralensverblinding	83
8.4.2	Het volgen van de dichtste fietser	83
8.4.3	Valse detecties	83
9	Toekomst en vervolg	85
9.1	Automatische camerakalibratie	85
9.1.1	Vaste afmetingen in de scène	85
9.2	Situaties waar drafting is toegelaten	86
9.2.1	Bochtdetector	86
9.2.2	Locatiebepaling	88
9.3	Uitbreidingsmogelijkheden	88
9.3.1	Meerdere fietsers detecteren en tracken	88
9.3.2	Laterale afstandsbepaling	90
9.4	Apparaten met beperkte rekenkracht	90
	Conclusie	94
	Referentielijst	95
	Bijlagen	105

Bijlage A: Enquête	105
Affiche Enquête	105
Bijlage B: Fietsafmetingen bij wedstrijdopnames	106
Bijlage C: Precision recall curves basis fietsdetectornetwerken	108
SSD	108
Faster RCNN	108
Mask RCNN	109
YOLOv3-tiny basic	109
YOLOv3-416 basic	110
Bijlage D: Flowcharts voor het algoritme van draftingdetectie	111
Hoofdapplicatie	111
Dichtste fiets bepaling	113
Draftingcontrole	113
Nieuwe fiets bepaling	113
Draftingtijd beginwaarde bepaling	113
Toekenning van een draftingpenalty	115
Bijlage E: Flowchart voor het algoritme van de kalibratiechecker	118
Bijlage F: Afstandsgrafieken voor enkele situaties tijdens een triatlonwedstrijd	119
Draften	119
Vorbijsteken	120
Vorbijgestoken worden	120
Bijlage G: afstandtesten beeldregistraties	121
Beeldregistratie - Statische situatietest	121
Beeldregistratie - Dynamische situatietest	121
Bijlage H: Kostprijsraming en specificaties	122

Lijst met figuren

Figuur 1	Een aantal atleten in een lange-afstandstriatlon	2
Figuur 2	Officiële draftingzone triatlon	3
Figuur 3	Renner C is aan het draften bij renner B	3
Figuur 4	Rijden in een peloton	3
Figuur 5	Waaierrijden	3
Figuur 6	Aerodynamische weerstand in een fietspeloton van 121 renners . . .	4
Figuur 7	Plaatsing van kegels om de afstandsgrens voor drafting aan te geven	5
Figuur 8	Enquêteresultaten voor een draftingdetectiesysteem	6
Figuur 9	Concept voor een draftingdetectiesysteem op basis van GPS	7
Figuur 10	Blokschema van het draftingdetectiesysteem op basis van compu- tervisie	8
Figuur 11	Een model van fotogrammetrie	10
Figuur 12	Geometrie van een pinhole camera	11
Figuur 13	Cameraopstelling GoPro Hero 6 achteraan het zadel	16
Figuur 14	Wegwerken van de vervorming met de methode van Zhang	16
Figuur 15	Verspringen van de camera als gevolg van slechte montage en een oneffen wegdek	17
Figuur 16	Drie situaties die zich tegelijk voordoen: voorbijsteken, draften en voorbijgestoken worden	19
Figuur 17	Verschillende voorkomende omstandigheden tijdens triatlonwedstrij- den	20
Figuur 18	Voorbeeld van een voorstelling van een <i>Deep Neural Network</i> voor gezichtsherkenning	22
Figuur 19	Aspecten met betrekking tot objectdetectie	24
Figuur 20	RCNN schematisch voorgesteld	26
Figuur 21	Mask RCNN toepassing op een beeld	27
Figuur 22	Voorbeeld van YOLO - 3x3 rooster	27
Figuur 23	Voorbeeld van YOLO - begrenzkader met <i>annotation</i>	28
Figuur 24	AP-50 versus detectiesnelheid voor verschillende basis objectdetc- tienetwerken	30
Figuur 25	Log Average Miss Rate voor verschillende basis objectdetectienet- werken	32
Figuur 26	Twee voorbeelden van een triatlonfiets	33
Figuur 27	Voorbeeld van een opgenomen beeld	34
Figuur 28	AP-50 versus detectiesnelheid voor verschillende objectdetectienet- werken samen met drie getrainde YOLO-netwerken	35

Figuur 29	Log Average Miss Rate voor verschillende basis objectdetectienetwerken samen met drie getrainde YOLO-netwerken	35
Figuur 30	Precision recall van een YOLOv3-netwerk voor verschillende IOU-waarden	36
Figuur 31	Precision recall van een Tiny YOLOv3-netwerk voor verschillende IOU-waarden	37
Figuur 32	Precision recall van het getrainde YOLOv3- (ononderbroken lijnen) en Tiny YOLOv3-netwerk (stippellijnen) voor verschillende IOU-waarden .	38
Figuur 33	Situatie waarbij atleet A en C niet dezelfde atleet zijn	40
Figuur 34	Tracking-by-detection principe voor het detecteren van fietsers . . .	41
Figuur 35	Algemeen concept van objecttracking	44
Figuur 36	Principe van de CAMshift methode	46
Figuur 37	Kalmanfilter - discrete cyclus	47
Figuur 38	Successplots voor verschillende objecttrackers met in de legende de AUC-score voor elke overeenkomstige objecttracker	50
Figuur 39	Precisionplots voor verschillende objecttrackers waarbij de gemiddelde precisiescore van elke objecttracker is weergegeven in de legende . . .	51
Figuur 40	Successrate versus detectiesnelheid voor verschillende objecttrackers	51
Figuur 41	Het ontstaan van trackingsdrift en lage precisie voor verschillende objecttrackers	52
Figuur 42	Detection rate voor verschillende IOU's voor het getrainde (Tiny) YOLOv3-netwerk en de DSST-objecttracker, waarbij de legende de detection rate weergeeft voor een IOU van 0.5	53
Figuur 43	Precision recall curve voor Tiny YOLOv3 met de DSST-objecttracker	53
Figuur 44	Pixelfout op de hoogte van de bounding box, met in de legende de gemiddelde fout	54
Figuur 45	Pixelfout op de hoogteligging van de bounding box ten opzichte van het grondvlak, met in de legende de gemiddelde fout	55
Figuur 46	Correspondentieprobleem voor het associëren van fietsen	56
Figuur 47	Cameraplaatsingcorrectie (paars) met stuurcorrectie (oranje)	58
Figuur 48	Principe van de hoogtemethode voor afstandsbepaling	58
Figuur 49	Hoogtemethode verschil tussen een gewone racefiets en een typische triatlonfiets	59
Figuur 50	Principe van de hoogtemethode y_1 en de grondvlakmethode y_2 . . .	60
Figuur 51	Principe van de grondvlakmethode voor afstandsbepaling	60
Figuur 52	Horizonverplaatsing afhankelijk van de tilthoek θ_{tilt} van de camera .	61
Figuur 53	Helling van een brug	62

Figuur 54	Ligging van de epipool als de tilthoek initieel 0° bedraagt	62
Figuur 55	Verticale epipoolligging bepalen	63
Figuur 56	Ligging van epipool aan de hand van optical flow vectoren (groen) volgens de convergentielijnen (rood)	64
Figuur 57	Kalibratiechecker twee strategieën	65
Figuur 58	Kalibratiechecker test	66
Figuur 59	Berekende afstand in functie van de tijd - zonder filtering	67
Figuur 60	Berekende afstand in functie van de tijd - na toepassen van het Kalmanfilter	68
Figuur 61	Statische opmeting - opstelling met een camerafiets en een achter- liggende fiets voor de statische situatietest	69
Figuur 62	Statische opmeting - absolute afstandsfout voor de hoogtemethode en grondvlakmethode, waarbij in de legende het gemiddelde van de absolute fout over alle grondwaarheidafstanden is beschouwd voor beide methoden .	71
Figuur 63	Dynamische opmeting - LIDAR-opstelling met een bakfiets, came- rafiets en een achterliggende fiets voor de dynamische situatietest	72
Figuur 64	Dynamische opmeting - LIDAR-afstanden in vergelijking met de afstanden van de hoogtemethode en grondvlakmethode	73
Figuur 65	Dynamische opmeting - absolute afstandsfout voor de hoogteme- thode en grondvlakmethode, waarbij in de legende het gemiddelde van de absolute fout over alle grondwaarheidafstanden is beschouwd voor beide methoden	73
Figuur 66	Dynamische opmeting - histogram van gemiddelde afstandsfouten met 95%-betrouwbaarheidsinterval	74
Figuur 67	Hoogte in pixels van de bounding box voor verschillende grond- waarheidafstanden voor de hoogte- en grondvlakmethode	75
Figuur 68	Absolute fout op de y -coördinaat voor verschillende afstanden voor de hoogte- en grondvlakmethode, waarbij in de legende de gemiddelde ab- solute fout over alle grondwaarheidafstanden van beide methoden is weer- gegeven	76
Figuur 69	De verdeling van de kans dat een triatleet zich in de draftingzone van 10 meter bevindt voor een gemeten afstand a	78
Figuur 70	Principe om de draftingdetectiescore over 20 seconden te bepalen gemeten over een venster van 30 seconden en een afstandsmeting om de 2 seconden	78
Figuur 71	Screenshot van een triatleet die de draftingregel heeft overtreden . .	81
Figuur 72	Onnauwkeurige detectie van een fiets als gevolg van regendruppels op de cameralens	83

Figuur 73	Afstand van de camera tot het achterwiel	86
Figuur 74	Gyroscoop en accelerometer data in het geval van een bocht	87
Figuur 75	Non-draftingzones aangeduid in het rood	89
Figuur 76	Het volgen van meerdere fietsers die in een peloton rijden	89
Figuur 77	Overzichtsschema van de drie belangrijke bemonsteringsparameters W, M en S met betrekking tot de bepaling van de draftingsscore	91
Figuur 78	Ideaal detectiemasker	92
Figuur 79	Detectiemasker opgebouwd op basis van de onderkant van de gedetecteerde bounding boxes uit de testset	93
Figuur 80	Detectiemasker opgebouwd op basis van de gedetecteerde bounding boxes uit de testset	93
Figuur 81	Affiche van de uitgevoerde enquête	105
Figuur 82	Afstand tot en met 10 meter	106
Figuur 83	Opgemeten afstanden	106
Figuur 84	Precision recall SSD	108
Figuur 85	Precision recall Faster RCNN	108
Figuur 86	Precision recall Mask RCNN	109
Figuur 87	Precision recall YOLOv3-tiny basic	109
Figuur 88	Precision recall YOLOv3-416 basic	110
Figuur 89	Flowchart hoofdapplicatie	112
Figuur 90	Flowchart voor de bepaling van de dichtste fiets	114
Figuur 91	Flowchart voor de draftingcontrole	115
Figuur 92	Flowchart voor de bepaling van een nieuwe fiets	116
Figuur 93	Flowchart voor de bepaling van de initiële draftingtijd	116
Figuur 94	Flowchart voor de toekenning van een draftingpenalty	117
Figuur 95	Flowchart voor de kalibratiechecker	118
Figuur 96	Een atleet die meer dan 30 seconden aan het draften is	119
Figuur 97	Een atleet die 100 seconden aan het draften is	119
Figuur 98	Situatie waarbij er iemand wordt voorbijgestoken	120
Figuur 99	Situatie waarbij de achterligger voorbijsteekt	120
Figuur 100	Voorbeeld van een beeldregistratie van de statische situatietest	121
Figuur 101	Voorbeeld van een beeldregistratie van de dynamische situatietest	121

Lijst met tabellen

Tabel 1	AP-50 en detectiesnelheid voor verschillende basis objectdetectienetwerken	30
Tabel 2	Confusion matrix voor het detecteren van fietsen	32
Tabel 3	AP-50 en detectiesnelheid voor getrainde YOLOv3- en Tiny YOLOv3-objectdetectienetwerken	34
Tabel 4	De AP voor YOLOv3 en Tiny YOLOv3 voor verschillende IOU-thresholdwaarden.	37
Tabel 5	Statische opmeting - afstandsmetingen voor de hoogte- en grondvlakmethode voor grondwaarheidafstanden van 0 tot 20 meter	70
Tabel 6	Gesamplede afstanden met de kans dat de draftingregel is overtreden	80
Tabel 7	Benodigheden, specificaties en kostprijs voor het draftingdetectiesysteem	122

Lijst met afkortingen

AUC	Area Under the Curve
BI	Betrouwbaarheidsinterval
BLE	Bluetooth Low Energy
CAM	Continuously Adaptive Meanshift
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CRO	Chief Race Official
CSRT	Channel and Spatial Reliability correlation filter Tracker
DNN	Deep Neural Network
DPM	Deformable Part Models
DSST	Discriminative Scale Space Tracker
EIS	Electronic Image Stabilization
erf	error function
FN	False Negative
FoM	Figure of Merit
FOV	Field Of View
FP	False Positive
FPPI	False Positives Per Image
fps	frames per second
GMM	Gaussian Mixture Model
GOTURN	Generic Object Tracking Using Regression Networks
GPS	Global Positioning System

GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
IOU	Intersection Over Union
ISM	Industrial, Scientific and Medical
IVS	Intelligent Visual Surveillance
KCF	Kernelized Correlation Filter
LAMR	Log Average Miss Rate
LIDAR	LIght Detection And Ranging
LOS	Line Of Sight
LP-WAN	Low-Power Wide Area Networks
mAP	mean Average Precision
MIL	Multiple Instance Learning
MOSSE	Minimum Output Sum of Squared Error
MR	Miss Rate
PPV	Positive Predictive Value
RADAR	Radio Detection And Ranging
RCNN	Region based Convolutional Neural Networks
ROI	Region of Interest
SIFT	Scale Invariant Feature Transform
SOA	System Oriented Architecture
SSD	Single Shot MultiBox Detector
TLD	Tracking; Learning and Detection
TP	True Positive
TPR	True Positive Rate
UNB	Ultra-Narrowband

VAR	Video assistant referee
VOC	Visual Object Classes
WS	Web Service
YOLO	You Only Look Once

Hoofdstuk 1

Inleiding

Tijdens een wielervedstrijd moeten wedstrijdmotoren volgens de Internationale Wielervedunie 10 meter afstand houden van renners die er achter rijden. Het probleem is dat motoren en/of volgwagens zich niet altijd aan deze regel houden, gezien er voortreffelijke livebeelden genomen moeten worden. Een voertuig dat vóór een wielrenner of peloton rijdt, heeft een positieve invloed op de luchtweerstand die de deelnemers ondervinden. Een tevens niet te onderschatten effect is de situatie waar een volgwagen of motor naast of achter een wielrenner rijdt. De fietsdiscipline bij triatlon en duatlon hanteert gelijkaardige regels waarbij een draftingzone voor elke fietser wordt gedefinieerd. Indien een achterligger zich in deze zone bevindt, wordt dit omschreven als *draften* of stayeren.

In deze scriptie wordt er gefocust op het fietsen binnen de triatlon waar in tegenstelling tot het wielrennen niet in peloton mag gereden worden. Triatlon, en duatlon voor de volledigheid, wordt immers gezien als een individuele sport. De quote op pagina ii is een voorbeeld van het ongenoegen van Frederik Van Lierde na het wereldkampioenschap Ironman in Hawaï. Naar zijn mening heeft hij onterecht een draftingpenalty gekregen van de jury. Het ontbreken van betrouwbare metingen leidt in sommige gevallen tot willekeur in het beste, en mogelijk zelfs tot corruptie zoals de bevoordeling van sommige toppers in het slechtste geval. Aangezien het controleren op drafting of stayering vandaag de dag met op de motor rondrijdende scheidsrechters gebeurt, is het doel van deze masterproef om een *proof of concept* uit te werken dat de taak voor het toekennen van draftingpenalties automatiseert. Voor deze automatisatie is de robuustheid van het systeem in alle mogelijke triatlonsituaties zoals vlakke of oneffen wegen en verschillende weersomstandigheden erg belangrijk.

De structuur van deze thesis is als volgt: Hoofdstuk 2 diept het probleem en de definitie van drafting uit met oog op de triatlonsport. In Hoofdstuk 3 wordt een uiteenzetting gegeven over fotogrammetrie en het cameramodel. Aan de hand van de opgenomen dataset worden nadien wedstrijsituaties uit de triatlon besproken, wat correspondeert met Hoofdstuk 4. De Hoofdstukken 5, 6 en 7 focussen zich respectievelijk op de drie belangrijkste aspecten van het draftingdetectiesysteem: objectdetectie, objecttracking en afstandsbepaling. De resultaten van dit onderzoek worden behandeld in Hoofdstuk 8, gevolgd door Hoofdstuk 9 dat zich toelegt op eventuele optimalisaties en uitbreidingen. Tot slot vormt de laatste paragraaf een besluit met betrekking tot het eindresultaat.

Hoofdstuk 2

Probleembeschrijving

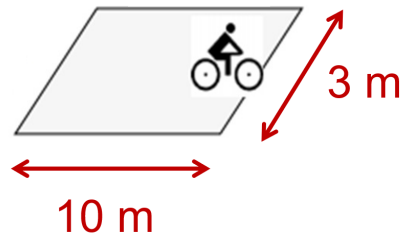
Wielrenners, duatleten of triatleten proberen tijdens wedstrijden uiteraard zo snel mogelijk de finish te bereiken. Naast optimalisatie van training, rust en voeding, kunnen deze atleten hun prestaties sterk verbeteren door te werken aan een gestroomlijnde houding of te zorgen voor aerodynamisch materiaal zoals opzetsturen en volle wielen. Er kan ook een enorm voordeel gehaald worden door tijdens een wedstrijd kort achter de voorligger te fietsen, zodat deze atleet de achterligger uit de wind zet. Deze techniek wordt drafting, *slipstreaming* of stayeren genoemd. In Figuur 1 is een beeld weergegeven waarbij een aantal atleten in een lange-afstandstriatlon¹ niet aan het draften zijn.



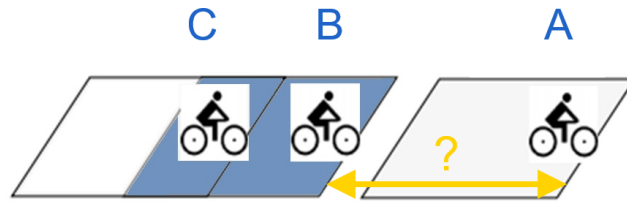
Figuur 1: Een aantal atleten in een lange-afstandstriatlon [2]

Er mag binnen de triatlon en duatlon aldus niet in peloton worden gereden daar deze individuele sporten zijn. Volgens de officiële regels van de triatlon mogen atleten, in het geval van non-draftingwedstrijden, niet langer dan 20 seconden in de draftingzone van 10 meter op 3 meter blijven zoals is weergegeven in Figuur 2. Deze longitudinale afstand wordt gemeten van voorwiel tot voorwiel. Dienovereenkomstig is in Figuur 3 renner C aan het draften bij B die op zijn beurt niet aan het stayeren is bij A. Renner C mag zich bijgevolg maximaal 20 seconden in de draftingzone van B bevinden, gekleurd in het donkerblauw in Figuur 3, om deze fietser bijvoorbeeld voorbij te rijden. Eens ingehaald,

¹Een Ironman lange-afstandstriatlon bestaat uit drie marathons: 3,8 km zwemmen, 180 km fietsen en 42,2 km lopen [1].



Figuur 2: Officiële draftingzone triatlon (eigen bewerking van [1])



Figuur 3: Renner C is aan het draften bij renner B (eigen bewerking van [1])

moet renner B zich binnen de 5 seconden laten afzakken tot buiten de draftingzone van C om een draftingpenalty te voorkomen.

Om een eerlijker krachtmeting te verkrijgen, moeten de atleten in tijdritten of lange-afstandstriatlons daarom de nodige afstand bewaren volgens de draftingzone zoals in Figuur 1 het geval is. Deze tussenafstand is afhankelijk van wedstrijd tot wedstrijd, maar bedraagt veelal 10 meter en in uitzonderlijke gevallen 15 meter. Kortom mag er binnen de triatlon niet in peloton of in een waaier gereden worden in tegenstelling tot het wielrennen zoals weergegeven in respectievelijk Figuur 4 en Figuur 5.

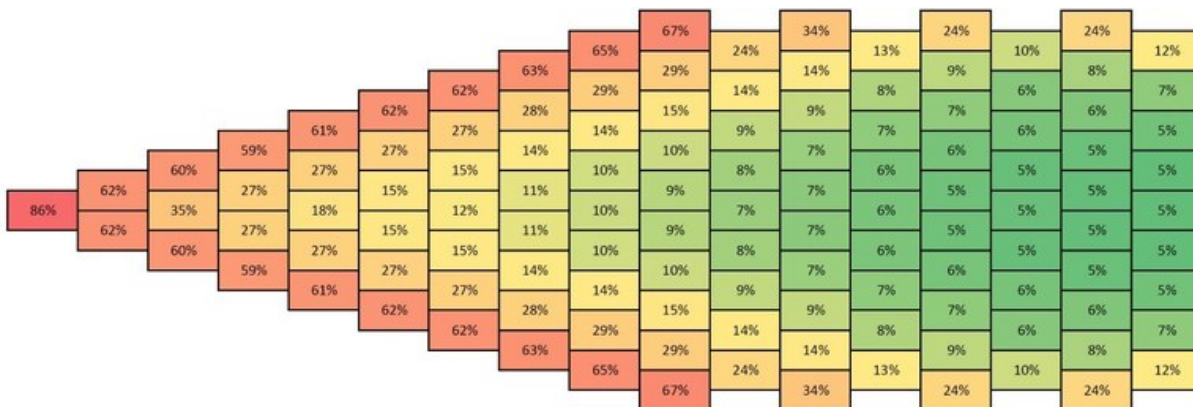
Om duiding te geven in welke mate het in peloton rijden invloed heeft op de luchtweerstand, wordt verwezen naar het recent windtunnelonderzoek aan de hand van een *Computational Fluid Dynamics* (CFD) simulatie [4]. Het resultaat in het geval van een peloton van 121 renners is weergegeven in Figuur 6 . De percentages in de beschouwde figuur duiden de luchtweerstand aan die de renner op die positie in het peloton ervaart ten opzichte van een renner die alleen rijdt aan dezelfde snelheid. In een



Figuur 4: Rijden in een peloton [3]



Figuur 5: Waaierrijden [3]



Figuur 6: Aerodynamica weerstand in fietspeloton van 121 renners [4]

wielervedstrijd zullen daarom sprinters van het team nooit eerst rijden, maar ergens in het peloton om zoveel mogelijk energie te sparen voor hun meest kritische deel van de wedstrijd.

Vandaag de dag wordt de toegelaten afstand op triatlonwedstrijden gecontroleerd door scheidsrechters op motors. Echter is het bewaken van deze regels moeilijk, zodat er in de praktijk toch vaak tegen gezondigd wordt. Een eerste reden is dat er te weinig scheidsrechters beschikbaar zijn om alle atleten altijd en overal te controleren, dan nog zijn hun beslissingen subjectief gezien ze de afstanden tussen de atleten enkel ruwweg kunnen schatten. In de praktijk blijkt bovendien dat de jury vaak alleen bij de kop of subtop van de wedstrijd aanwezig is, in de middenmoot is dit maar heel sporadisch. Uit een eigen uitgevoerde enquête² met een steekproefgrootte van 351 blijkt dat 89,40 % vindt dat drafting strenger gecontroleerd moet worden. Bovendien is het voor de renner zelf niet altijd gemakkelijk om de afstand tussen de voorganger goed in te schatten. In diezelfde enquête geeft 45,10 % aan dat hij of zij soms onbewust stayert in een non-draftingwedstrijd. Om hieraan vooraf te gaan worden bij bepaalde wedstrijden strepen getrokken of kegels geplaatst die op 10 meter van elkaar staan om de atleten een indicatie voor deze afstand te geven zoals weergegeven in Figuur 7.

Aangezien dit niet altijd het geval is en het ook voor de scheidsrechters een belangrijke taak is om correct op te treden tegen het stayeren, wordt er in deze masterproef een systeem ontwikkeld dat een melding kan geven indien drafting het geval is. Een detectiesysteem op basis van videobeelden genomen met een aan het zadel gemonteerde camera kan hiervoor samen met computervisietechnieken een oplossing bieden. Het automatisch detecteren van drafting kan bijgevolg ook als een hulpmiddel voor

²De vragen van deze enquête zijn terug te vinden via volgende link: https://docs.google.com/forms/d/e/1FAIpQLScwChAtTe3519aRNVjB4b3NLLpFJhGEOtDofQHy3aYvjzt-iQ/viewform?usp=sf_link, de affiche horende bij deze enquête bevindt zich in Bijlage A.



Figuur 7: Plaatsing van kegels om de afstandsgrens voor drafting aan te geven

beginnelingen ingezet worden om de 10 meter inschatting te vergemakkelijken.

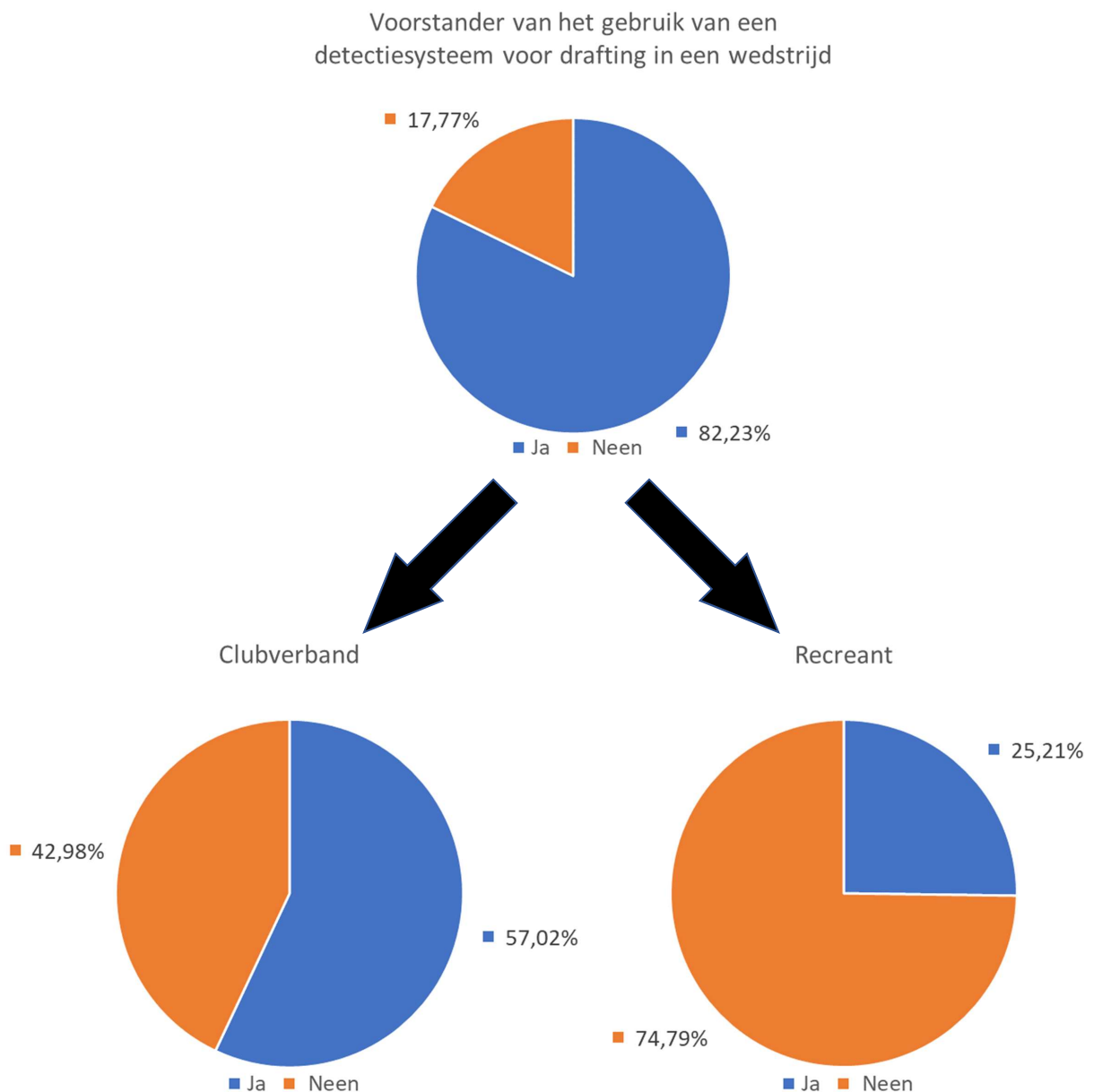
Uit de eerder aangehaalde enquête blijkt dat 82,23 % te vinden is voor een systeem dat stayering detecteert met een camera, zoals weergegeven in Figuur 8. De diagrammen tonen tevens aan dat het grote deel van de populatie die voorstander is van een detectiesysteem tegen drafting bij een club is aangesloten.

Naast subjectiviteit en het leiden tot gevaarlijke situaties op het parcours zijn de motoren waarop de scheidsrechters rijden ook hoorbaar waardoor ze zichzelf verraden. Op die manier kan men minder renners op heterdaad betrappen op drafting om zo een gepaste straftijd toe te kennen.

Voor dit systeem kan er intelligentie geïmplementeerd worden om bepaalde situaties zoals natuurlijke accordeonbewegingen als gevolg van een bocht, een helling, ... in rekening te brengen. In de praktijk zijn er namelijk situaties waar stayering onvermijdelijk is, bijvoorbeeld in een bocht waar de voorganger teveel afremt, in een beklimming waar de afstand korter wordt omdat er trager wordt gereden, ... Volgens het triatlonreglement is het overigens mogelijk om de penalty aan te vechten waarna de *Chief Race Official* (CRO), de hoofdscheidsrechter, de tijdstraf zal verduidelijken met eventuele bewijsstukken. Door een draftingdetectiesysteem op basis van videobeelden kunnen, zoals bij het voetbal de *Video assistant referee* (VAR), de beelden achteraf bekeken en gebruikt worden als bewijsmateriaal.

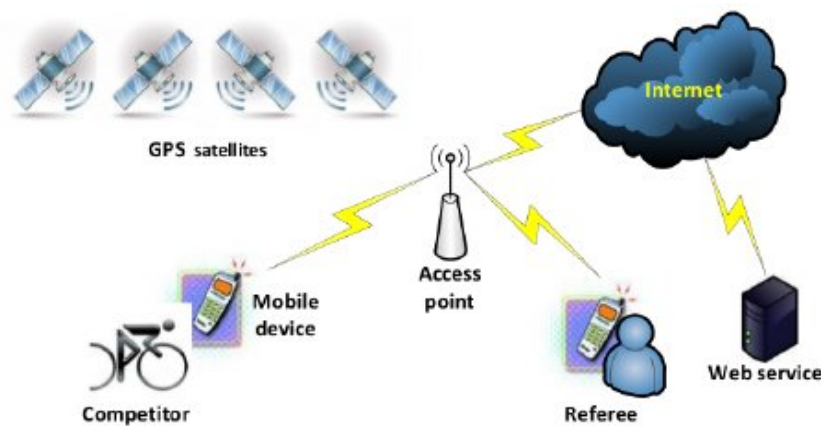
Samengevat kan er gesteld worden dat door een voortdurende controle dit op termijn zal leiden tot minder stayering in een wedstrijd en bijgevolg zorgen voor een eerlijker wedstrijdverloop.

Naast het gebruik van een camera zijn er nog andere objectieve oplossingen die dit probleem kunnen tegemoetkomen. Als er gekeken wordt naar het elektromagnetisch



Figuur 8: Enquête resultaten voor een draftingdetectiesysteem

spectrum zijn *Global Positioning System* (GPS) en *Bluetooth Low Energy* (BLE) mogelijkheden, doch met beperkte nauwkeurigheid. GPS ondervindt de typische fouten als gevolg van *Urban Canyons*, hoge bomen, tunnels of bruggen, ... Bluetooth in het andere geval werkt in de 2,4 GHz *Industrial, Scientific and Medical* (ISM) band en ondervindt bijgevolg interferentie door radiosignalen van andere apparaten. In [5] wordt een concept voor een draftingdetectiesysteem voorgesteld op basis van GPS en *Web Services* (WS) zoals is weergegeven in Figuur 9. Elke deelnemer beschikt volgens deze implementatie over een GPS-ontvanger die zijn signalen doorstuurt via een draadloze modem naar een web service. Deze service kan op zijn beurt bepalen of deze



Figuur 9: Concept voor een draftingdetectiesysteem op basis van GPS [6]

deelnemer de draftingregels al dan niet aan het overtreden is. De betreffende informatie kan tevens opgevraagd worden door de scheidsrechters op de motors. Zoals eerder gesteld is GPS beperkt in zijn nauwkeurigheid: tot 5 meter voor commercieel gebruik [7]. Als oplossing voor deze beperking biedt het globale navigatie systeem Galileo een precisie tot 1 meter [7]. Echter is dit Europees systeem pas operationeel tegen 2021. Om de positionele informatie door te sturen wordt gebruikgemaakt van het *System Oriented Architecture* (SOA) architectuurmodel. De relatieve euclidische afstanden tussen de verschillende atleten onderling worden bepaald door de afgelegde weg te beschouwen waarna een sortering van alle deelnemers wordt uitgevoerd. Als de afgelegde afstand tussen een deelnemer (i) en de volgende deelnemer in de gesorteerde lijst ($i + 1$) kleiner is dan de opgelegde draftingafstand en dit voor meer dan 20 seconden, dan wordt er een draftingpenalty toegekend aan deelnemer ($i + 1$). Voor de evaluatie van het systeem zijn verschillende soorten GPS-ontvangers gebruikt zoals een smartphone en een hooggevoelige GPS-ontvanger. De resultaten in [5] geven aan dat de gemiddelde nauwkeurigheid voor de absolute positie van elke triatleet 1,32 meter bedraagt. Om de afstand tussen twee fietsers te bepalen, wordt er relatief gewerkt. De fout op de relatief gemeten afstand kan bijgevolg in het slechtste geval 2,64 meter bedragen, wat een niet te overziene afwijking is voor de draftingdetectie applicatie. Een ander nadeel is dat elke renner over zo een systeem met GPS- en mobiele verbinding moet beschikken aangezien de afstandsbepaling globaal bekeken wordt over alle renners in de wedstrijd. De onderlinge afstanden tussen de renners worden vervolgens aangewend om drafting te detecteren. In deze thesis wordt gebruikgemaakt van een camerasysteem en wordt er lokaal gekeken in de scène. Er is hierdoor geen afhankelijkheid van een mobiel netwerk of positiebepaling op basis van satellieten.

Twee meer toegankelijke oplossingen voor een draftingdetectiesysteem betreffen: *LIght Detection And Ranging* (LIDAR) en *RAdio Detection And Ranging* (RADAR). Het

eerstgenoemde is een dure techniek, maar is erg nauwkeurig. Het laatstgenoemde is compacter, goedkoper en werkt op basis van radiogolven in plaats van optische laserstralen. Het nadeel ten opzichte van LIDAR is dat de nauwkeurigheid lager ligt. Het object moet bovendien voor objectdetectie (in dit geval een fietser) in beide methoden genoeg reflectiepunten hebben. Om het verschil tussen statische en bewegende objecten in het beeld te bepalen, kan er gesteund worden op het dopplereffect³. Beide oplossingen kunnen gebruikt worden om het computervisiegebaseerde systeem te testen met als grondwaarheid de output van één van de twee genoemde rangingstechnieken. Een laatste mogelijkheid voor draftingdetectie is om beroep te doen op *Ultra-Narrowband* (UNB) modulatie gebruikt in *Low-Power Wide Area Networks* (LP-WAN) voor langeafstandscommunicatie (tot enkele kilometers) zoals *Sigfox* [9]. In deze scriptie wordt niet verder ingegaan op deze draadloze communicatietechnologie.

In het algemeen kan de werking van het realtime draftingdetectiesysteem op basis van een camera in een blokschema samengevat worden zoals weergegeven in Figuur 10. Dit schema vormt de basisstructuur van deze thesis.



Figuur 10: Blokschema van het draftingdetectiesysteem op basis van computervisie

Het detecteren van fietsers wordt in Hoofdstuk 5 besproken. In een volgend hoofdstuk wordt besproken hoe een gedetecteerde fietser doorheen de scènes kan worden gevolgd. Nadien wordt in Hoofdstuk 7 de eigenlijke afstandsbepaling tot de beschouwde fietser uit de doeken gedaan. Aan de hand van meetkundige technieken, assumpties over de kijkrichting van de camera en de ligging van het grondvlak, kan de afstand tot nabijgelegen objecten bepaald worden. Deze toepassing brengt enkele specifieke uitdagingen met zich mee. Zo is er een nauwkeurige kalibratie van de camera nodig zodat de ligging van de camera ten opzichte van het grondvlak gekend is. Gedurende een wedstrijd zijn er situaties die voor moeilijkheden kunnen zorgen zoals: het onderscheiden van (mede-)fietsers ten opzichte van de achtergrond en de toeschouwers, het omgaan met trillingen en schommelingen in het beeld als gevolg van de typische beweging van een fiets, het opvangen van een mogelijk niet-vlak wegdek, ...

³Garmin heeft sinds 2015 een systeem, *Garmin Varia Bike Radar* [8], voor fietsen op de markt gebracht dat tot op een afstand van 140 meter kan waarschuwen voor achteropkomende voertuigen. Het detecteert wel geen voertuigen die aan dezelfde snelheid als de fiets voortbewegen.

Rekening houdend met algemene omstandigheden tijdens triatlonwedstrijden moet dus kortom, de afstand bepaald worden tot een in de scène bewegend achterliggende renner gevolgd door een controle of er meer dan 20 seconden (de draftingtijdslimiet) wordt gedraaft. Indien dit laatste het geval is, kan er een signaal naar buiten gestuurd worden; dit kan bijvoorbeeld een knipperend led zijn. Een andere mogelijkheid is dat er een snapshot van de overtreder wordt gemaakt. Het aangehaalde LIDAR-systeem wordt gebruikt als grondwaarheid om een grondige validatie van het systeem te bekomen, wat onderzocht wordt in Hoofdstuk 8. Het wordt dusdanig duidelijk hoe betrouwbaar de detectie is in verschillende omstandigheden en wat de mogelijke uitbreidingen kunnen zijn.

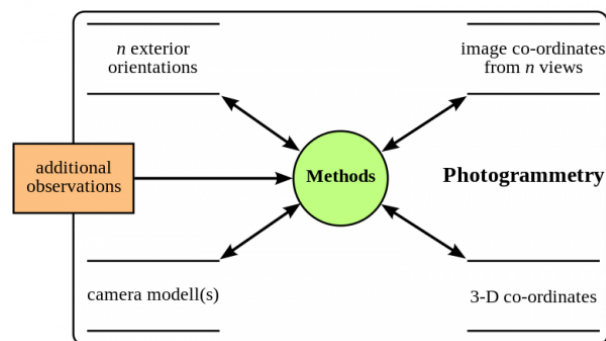
Hoofdstuk 3

Fotogrammetrie en cameramodel

In dit hoofdstuk wordt een uiteenzetting gegeven over een fundamenteel probleem in de computervisie: de structuur van een scène in de driedimensionale ruimte begrijpen op basis van verschillende afbeeldingen ervan. Technieken voor het oplossen van dit probleem zijn ontleend aan de projectieve geometrie en de fotogrammetrie en vormen een belangrijk onderdeel van dit onderzoek. Er wordt ook dieper ingegaan op het cameramodel en de daarmee verband houdende camerakalibratie voor het bepalen van de extrinsieke en intrinsieke parameters.

3.1 Fotogrammetrie

Fotogrammetrie is een term binnen de computervisie en houdt zich bezig met de interpretatie en opmeting van beeldmateriaal zodoende de vorm, afmeting en ligging van een object te beschrijven [10]. Kortom is fotogrammetrie de wetenschap van het maken van metingen van foto's, in het bijzonder voor het bekomen van de exacte posities van oppervlaktepunten van een 3D-omgeving. Het gegevensmodel in Figuur 11 laat zien welk type informatie kan worden gebruikt en uit de fotogrammetrische methoden komt.



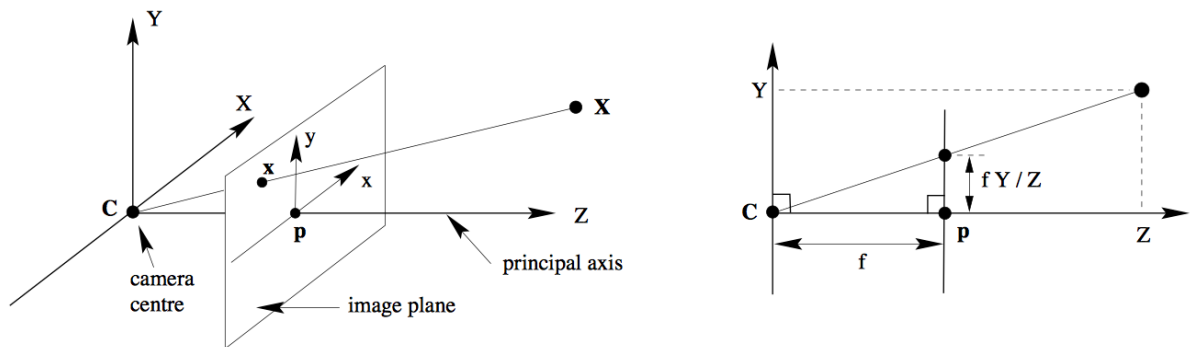
Figuur 11: Een model van fotogrammetrie [11]

De 3D-coördinaten betreffen de locatie(s) van objectpunten in de 3D-ruimte, ook wel de *world space* of het wereldcoördinatensysteem genaamd. De *image coordinates* daarentegen komen overeen met de corresponderende locaties van de objectpunten in een tweedimensionale afbeeldingsruimte.

De externe oriëntatie van een camera definieert de ruimtelijke locatie en de kijkrichting van het toestel en wordt gedefinieerd door de extrinsieke parameters. Er bestaan daarenboven ook interne of intrinsieke parameters die samen met de externe parameters hand in hand gaan met het model van de camera, zoals aan bod komt in het onderdeel 3.2.

3.2 Cameramodel

Een veelgebruikt cameramodel is het *pinhole* model. Het vastleggen van een afbeelding met een camera volgens dit model stemt overeen met het mappen van een 3D-beeld in een 2D-afbeelding zoals voorgesteld in Figuur 12. Elk punt van de 3D-wereld wordt bijgevolg toegewezen aan het 2D-afbeeldingsvlak. De coördinaten in de beschouwde



Figuur 12: Geometrie van een pinhole camera [12]

illustratie worden uitgedrukt in homogene coördinaten in plaats van projectieve coördinaten. In de computervisie en beeldverwerking wordt veel gebruikgemaakt van deze voorstellingswijze omdat het mogelijk is om operaties zoals translatie, schaling, rotatie en perspectiefprojectie te implementeren als matrixbewerkingen.

De mapping van een punt $\mathbf{X} = [X \ Y \ Z \ 1]^T$ in homogene coördinaten¹ in de 3D-wereld op een punt $\mathbf{x} = [x \ y \ 1]^T$ in het afbeeldingsvlak wordt voorgesteld door een 3x4 cameraprojectiematrix \mathbf{P} zoals aangegeven in formule (1) en (2).

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (1)$$

$$\mathbf{P} = \mathbf{K} \mathbf{R} [\mathbf{I} \ | \ -\mathbf{C}] \quad (2)$$

¹De vierde coördinaat van de vector heeft de waarde '1' wat wijst op een genormaliseerde vorm van homogene coördinaten. Een '0' daarentegen wijst op een zogenaamd *vanishing point* wat een punt is in een perspectieftekening waarbij parallelle lijnen die zich verwijderen van een waarnemer lijken te convergeren.

De matrix \mathbf{K} in formule (2) stelt de intrinsieke matrix voor en vertegenwoordigt de intrinsieke parameters van de camera. \mathbf{R} daarentegen stelt de extrinsieke rotatie en \mathbf{C} de translatie voor. Als de intrinsieke en extrinsieke parameters bijgevolg gekend zijn, is het cameramodel \mathbf{P} gekend.

3.2.1 Extrinsieke parameters

De extrinsieke parameters van een camera representeren volgens [13] de positie en de kijkrichting van de camera in de driedimensionale ruimte. Het legt bijgevolg de relatie vast tussen het 3D-wereldcoördinatensysteem tot het 3D-cameracoördinatensysteem aan de hand van de rotatie- en translatiematrix, respectievelijk \mathbf{R} en \mathbf{C} in formule (2).

3.2.2 Intrinsieke parameters

De interne beeldvervorming zou bij hetzelfde type camera met dezelfde instellingen quasi gelijk moeten zijn. Echter verschilt dit in de realiteit door kleine afwijkingen in de intrinsieke parameters van een camera. Deze parameters betreffen het optische centrum $C(c_x, c_y)$, de brandpuntsafstand of focale lengte f en de fysieke pixelgrootte die beïnvloed wordt door systeemruis van het toestel zoals aangegeven in [14]. Er moet daarnaast ook rekening gehouden worden met de eerder vermelde niet-lineaire radiale en tangentiële distorsie bij de niet-ideale lens.

De intrinsieke parameters representeren met andere woorden een projectietransformatie van het 3D-camera coördinatensysteem naar de 2D-afbeeldingscoördinatenruimte. Om deze parameters te bepalen moet er een kalibratie van de camera gedaan worden. Er bestaan hiervoor verschillende mogelijkheden zoals besproken wordt in onderdeel 3.3.

3.2.3 Lensdistorsie

Het ideaal pinhole cameramodel houdt geen rekening met lensvervorming omdat deze in principe niet over een lens beschikt. Om een reële camera bijgevolg zo goed mogelijk voor te stellen moet er rekening gehouden worden met radiale en tangentiële lensvervorming. De tangentiële distorsie is een gevolg van een verkeerde uitlijning van de lens. Radiale vervorming is de belangrijkste distorsie en komt voor als de lichtstralen meer afbuigen langsheen de randen van de cameralens dan bij het optische centrum. Hoe kleiner de lens, hoe groter deze vervorming is. Deze vorm van distorsie wordt vertegenwoordigd door de radiale distorsiecoëfficiënten van de lens: k_1 , k_2 en k_3 . Deze factoren worden gebruikt in de vervormde punten aangeduid als $x_{vervormd}$ en $y_{vervormd}$ in

formule (3) met $r^2 = x^2 + y^2$.

$$\begin{aligned}x_{vervormd} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\y_{vervormd} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6)\end{aligned}\tag{3}$$

Er wordt ook een onderscheid gemaakt tussen negatieve (*pincushion*) of positieve (*barrel*) radiale vervorming. Dit laatste geval is typisch voor een *fisheye* camera zoals een *GoPro*. Als gevolg van de aanwezige sferische lens bestaat de focale lengte f uit twee parameters: f_x en f_y . Voor meer informatie met betrekking tot lensvervorming wordt verwezen naar [12].

3.3 Camerakalibratiemethoden

Volgens [13] wordt camerakalibratie gezien als een essentiële stap in computervisie om metrische informatie te bekomen uit 2D-afbeeldingen en wordt ook *camera resectioning* genoemd. De kalibratietechnieken kunnen volgens [15] opgesplitst worden in twee categorieën: fotogrammetrische kalibratie en zelfkalibratie.

3.3.1 Fotogrammetrische kalibratie

Deze kalibratietechniek splitst zich op in een driedimensionaal objectgebaseerde en tweedimensionaal vlakgebaseerde kalibratie.

In het geval van de driedimensionaal objectgebaseerde kalibratie wordt gebruikgemaakt van een kalibratieobject waarvan de geometrie in de 3D-ruimte nauwkeurig gekend is. Zo'n object bestaat typisch uit twee of drie vlakken die loodrecht op elkaar staan om zo een assenstelsel te vormen. Het nadeel hiervan is dat er een nauwkeurig opzet is vereist en er nood is aan een kalibratieobject.

Bij de tweedimensionaal vlakgebaseerde kalibratie wordt een planair vlak onder verschillende oriëntaties bekeken. Een veel gebruikt patroon hiervoor is een dambordpatroon en is makkelijk eigenhandig te maken in tegenstelling tot een kalibratieobject. De twee bekendste methoden voor deze manier van kalibreren betreffen de methode van *Zhang* [13] en de methode van *Tsai* [16].

3.3.2 Zelfkalibratie

Bij deze manier van kalibreren is er geen kalibratieobject nodig en wordt de camera bewogen in een statische omgeving om puntovereenkomsten uit reeksen van afbeeldingen te bekomen. Volgens [15] zijn overeenkomsten tussen drie afbeeldingen al genoeg om zowel de interne als de externe parameters te verkrijgen en bijgevolg een

3D-reconstructie te bekomen. De keerzijde is echter dat er een groot aantal parameters bepaald moeten worden wat een zware mathematische berekening met zich meebrengt. In de literatuur zijn er nog enkele andere technieken voor camerakalibratie terug te vinden zoals kalibratie uit pure rotatie [17, 18] en kalibratie op basis van vanishing points voor orthogonale richtingen [19, 20].

Hoofdstuk 4

Cameraopnames en wedstrijdssituaties

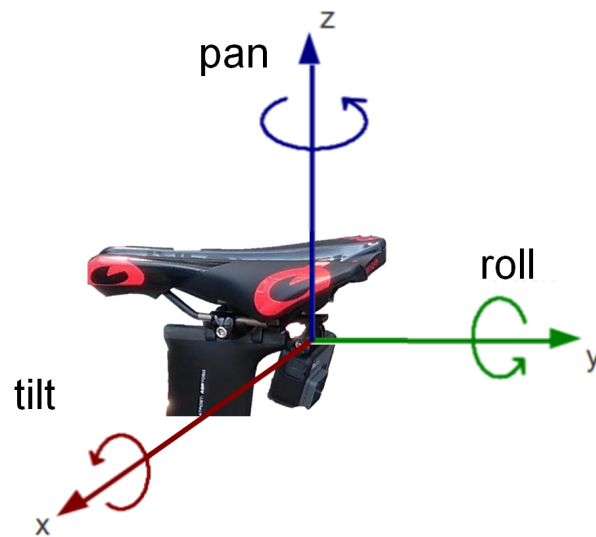
Aangezien er nood is aan een dataset om de applicatie grondig te testen in reële situaties, zijn er opnames gemaakt van telkens ongeveer een 1,5 uur en dit voor vier triatlonwedstrijden: ‘Triatlon Deinze’, ‘Den Halven van Damme’, ‘Swimkap Kwarttriatlon Kapelle-op-den-Bos’ en de ‘Zwintriathlon’ te Knokke. Meer bepaald zijn er triatleten die met een camera aan het zadel hebben gefietst gedurende de wedstrijd. Bijkomende uitleg omtrent de cameraopstelling wordt besproken in de eerste paragraaf van dit hoofdstuk. De dataset bestaat bovendien uit negentien ritten over verschillende fietslocaties in Oost- en West-Vlaanderen (Gent, Brugge, Oostende, Gavere, ...) waarbij een persoon continu al rijdend en stayerend in beeld is. In totaal is er ongeveer 35 uur aan eigen opgenomen beeldmateriaal waarvan ongeveer 6 uur tijdens triatlonwedstrijden. De veelvoorkomende situaties en uitdagingen van deze dataset worden in het tweede deel van dit hoofdstuk behandeld.

4.1 Cameraopstelling

Er wordt wegens praktische en economische redenen gekozen voor een systeem met een monoculaire cameraopstelling of anders gezegd: met één lens. De opstelling hiervan gebeurt achteraan het zadel zoals in Figuur 13 is weergegeven. Als gevolg hiervan kan de camera in deze montagepositie enkel een invloed ondervinden van de tilthoek. De camera kan tevens bewegen door trillingen en typische fietsbewegingen.

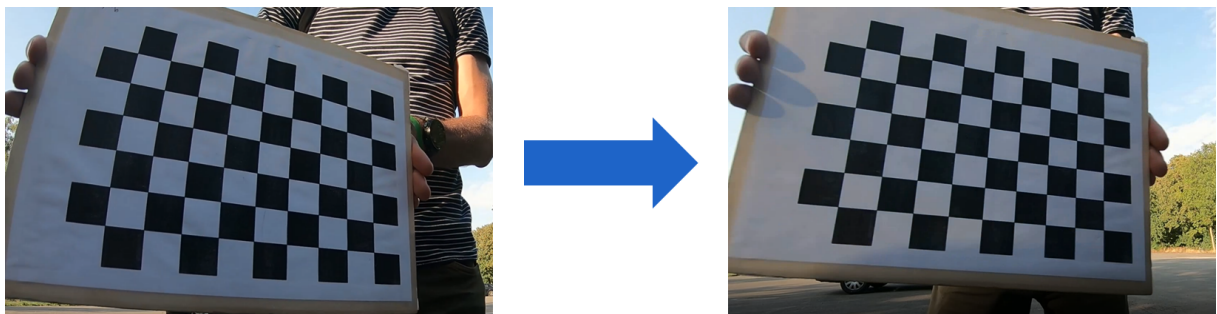
Er wordt volgens de relativiteitstheorie verondersteld dat de camera als waarnemer stilstaat en de rest in de omgeving relatief beweegt ten opzichte van de bewegende camera. De camera kan ook vooraan geplaatst worden, zodat een persoon bijvoorbeeld kan aantonen dat deze niet aan het draften is. Gezien de typerende bewegingen van het stuur is het geval waarbij de camera achteraan wordt geplaatst een betere keuze.

Voor ontwikkelingsdoeleinden wordt gebruikgemaakt van de *GoPro Hero 6 actioncamera*. Enerzijds wegens het gebruiksgemak en anderzijds beschikt deze camera over een in hardware geïmplementeerde *Electronic Image Stabilization (EIS)* functionaliteit. Het schokkerig beeld, als gevolg van trillingen geïntroduceerd door bijvoorbeeld een slecht wegdek, wordt gestabiliseerd door 10 % van het beeld bij te snijden. Alle beelden zijn opgenomen aan 30 fps (exact 29,97 fps) bij een HD-resolutie van 1920x1080. De desbetreffende camera heeft voorts een verticale *Field Of View*



Figuur 13: Cameraopstelling GoPro Hero 6 achteraan het zadel (eigen bewerking van [21])

(FOV) van $69,5^\circ$ en een horizontale FOV van $118,2^\circ$ wat hem als een breedhoekcamera classificeert [22]. Het nadeel van breedhoeklenzen is echter dat er radiale vervorming optreedt waardoor er een zichtbare kromming van de projectie is van rechte lijnen zoals gezien kan worden in Figuur 14. Door de methode van Zhang [13] toe te passen, zoals



Figuur 14: Wegwerken van de vervorming met de methode van Zhang

vermeld in onderdeel 3.3, kan aan de hand van een dambordpatroon de vervorming nadien verwijderd worden. Door een camerakalibratie te volbrengen, kunnen namelijk de intrinsieke parameters, extrinsieke parameters en vervormingscoëfficiënten verkregen worden. Uit de intrinsieke kalibratiematrix kan vervolgens de focale lengte van de niet-zoombare camera gehaald worden. Deze wordt in een eindontwerp constant verondersteld, maar kan variëren van ongeveer 987 tot 1020 pixels. De extrinsieken van de camera geven uitsluitsel over de tilthoek van de camera die, net zoals de focale lengte, nodig is voor de afstandsbepaling in Hoofdstuk 7. Voor experimentele doeleinden

is de tilthoek van de camera voor bepaalde ritten dermate ingesteld dat het achterwiel in beeld is zoals situatie (a) en (b) in Figuur 15.



(a) Achterwiel maximaal in beeld



(b) Achterwiel enigszins in beeld



(c) Achterwiel volledig uit beeld

Figuur 15: Verspringen van de camera als gevolg van slechte montage en een oneffen wegdek

Als gevolg van de verticale FOV van de camera geldt: hoe meer de camera naar onder is *getilt*, hoe kleiner de “dode zone” maar hoe meer de horizon naar boven komt te liggen en bijgevolg hoe kleiner de *Region of Interest* (ROI) van de scène. In deel (c) van de betreffende figuur kan bemerkt worden dat het achterwiel zich niet meer in beeld bevindt als gevolg van een ongewenste verdraaiing van de camera door een slecht wegdek. De enige vrijheidsgraad is bijgevolg de hoek waarover deze camera kan verdraaien: *tilthoek*. De kalibratie wordt als gevolg van het verspringen van de camera teniet gedaan, wat op zijn beurt een negatief effect kan veroorzaken op het resultaat van de afstandsrekening. Algemeen geldt: hoe minder vrijheidsgraden, hoe beter. De mogelijkheid voor het veranderen van de cameratilt moet bijgevolg onmogelijk worden beperkt worden in een eindproduct. De camera is in dit geval gemonteerd aan het zadel dat op zijn beurt ook getilt kan worden afhankelijk van renner tot renner. Dit tiltprobleem komt verder aan bod in Hoofdstuk 7.

Voor de opnames van de triatlonwedstrijden is in bepaalde gevallen voor en na de wedstrijd een camerakalibratie uitgevoerd. Als de camera niet is verdraaid dan zou de tilthoek in beide gevallen dezelfde moeten zijn. Bovendien zijn er afmetingen opgenomen van de gebruikte fietsen die gereden hebben in de wedstrijden, welke in Bijlage B te vinden zijn.

4.2 Wedstrijdsituaties

Een gedeelte van de beelden is geannoteerd voor fietsers en bijgevolg voorzien van grondwaarheid. De dataset kan opgesplitst worden in een trainingset, gebruikt voor het eventueel trainen van een objectdetector, en een testset voor het evalueren van de objectdetector en de applicatie. Voor de testset van de objectdetector zijn situatiebeelden gebruikt uit de verschillende triatlonwedstrijden waarbij over het algemeen de volgende situaties zich kunnen voordoen:

- Een atleet bevindt zich in de draftingzone en is bijgevolg aan het draften.
- Een triatleet steekt voorbij.
- Een deelnemer wordt voorbijgestoken.
- Een erg slecht of oneffen wegdek (bijvoorbeeld kasseien) waardoor er ruis, een *motion blur* artifact, wordt geïntroduceerd in het beeld en de EIS van de camera dit niet kan compenseren.
- Regendruppels belemmeren het zicht van de camera.
- Er wordt door een tunnel gereden waardoor de belichting verandert.
- De lichtstralen van de zon die op de lens invallen waardoor er een verblinding ontstaat.
- Er wordt een bocht genomen.
- Een helling van een beklimming, brug of tunnel.
- Op de trappers staan waardoor het beeld heen en weer schommelt.
- Er bevinden zich fietsen van toeschouwers aan de zijkant van het parcours die kunnen zorgen voor een valse detectie.
- Motors die rondrijden om de veiligheid te waarborgen kunnen ook verkeerdelijk als een fietser gedetecteerd worden.



Figuur 16: Drie situaties die zich tegelijk voordoen: voorbijsteken, draften en voorbijgestoken worden

De drie eerstgenoemde situaties doen zich over het algemeen het vaakst voor en kunnen daarenboven tegelijk voorkomen zoals weergegeven in Figuur 16. De overige vermelde omstandigheden die al dan niet frequent zijn in triatlonwedstrijden, zijn gevisualiseerd in Figuur 17.



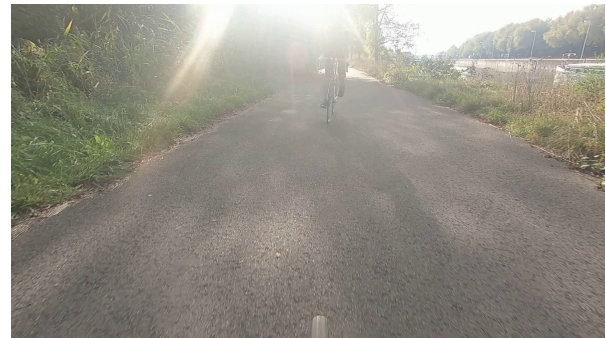
(a) Introductie van *motion blur*



(b) Regendruppels op de cameralens



(c) Het rijden in een tunnel



(d) Zonnestrallen op de cameralens zorgen voor een *flash* verblinding



(e) Het nemen van een bocht



(f) Hellende weg



(g) Op de trappers staan: beeld schommelt naar links



(h) Op de trappers staan: beeld schommelt naar rechts



(i) Interferentie door motor



(j) Interferentie door omstanders

Figuur 17: Verschillende voorkomende omstandigheden tijdens triatlonwedstrijden

Hoofdstuk 5

Objectdetectie

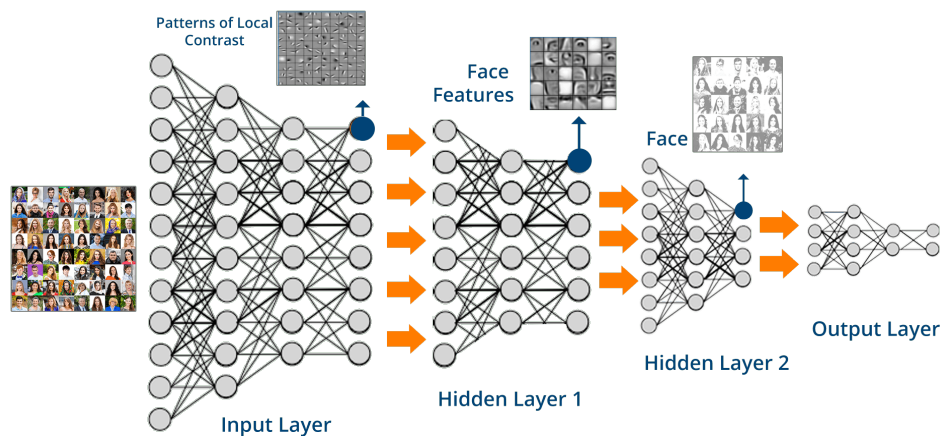
In dit hoofdstuk wordt objectdetectie behandeld beginnend met een overzicht van beschikbare mogelijkheden en algemene concepten uit de literatuur. In het eerste onderdeel worden enkele basisconcepten voor detectie besproken gevolgd door het *deep learning* concept in de beeldverwerking. Met oog op het detecteren van objecten, zoals een voetganger en een fietser, zijn er doorheen de jaren verschillende methoden ontwikkeld die ten opzichte van elkaar afgewogen worden. Op die manier wordt gekomen tot een selectie van het snelste, meest accurate en betrouwbare algoritme met oog op *realtime* draftingdetectie.

Aan de hand van de kennis opgedaan uit de literatuur met betrekking tot objectdetectie, wordt in een daaropvolgend onderdeel een verantwoorde keuze gemaakt voor een objectdetector. Meer bepaald zal deze gebruikt worden om realtime fietsers in beelden te detecteren. In een tweede paragraaf worden enkele objectdetectornetwerken, besproken in de eerste paragraaf, gebenchmarkt op testbeelden uit een triatlonwedstrijd. Het doel is om fietsers te detecteren tot op 20 meter om zodoende voldoende marge te hebben als de draftingafstand bijvoorbeeld op 15 meter wordt gekozen (wat door sommige wedstrijdorganisatoren wordt gehanteerd). Nadien wordt in een derde onderdeel een getraind netwerk voorgesteld en geëvalueerd.

5.1 Objectdetectie - literatuur

De hersenen zijn één van de meest complexe onderdelen van het lichaam. Met betrekking tot zicht kost het de mens geen moeite om een kat te onderscheiden van een hond, een teken te begrijpen of de gezichtsexpressie van een persoon te herkennen. Dit zijn lastige problemen om op te lossen met een computer. Het is hier dat de aspecten van *classifiers* en neurale netwerken een belangrijke rol spelen. Een *Deep Neural Network* (DNN) is een *Artificial Neural Network* (ANN) met meerdere lagen tussen de *input layer* en *output layer*. Zo'n DNN is schematisch weergegeven in Figuur 18 aan de hand van een conceptueel voorbeeld met betrekking tot gezichtsherkenning.

Deep learning is een onderdeel binnen AI en is één van de meest gebruikte technieken binnen de computervisie. Het is een relatief oude methode maar heeft door de steeds groeiende hoeveelheid aan data en toegenomen mogelijkheden in rekenkracht de laatste jaren zijn populariteit herwonnen. Het is een techniek waarbij een machine gevoed



Figuur 18: Voorbeeld van een voorstelling van een *Deep Neural Network* voor gezichts-herkenning [23]

wordt met een netwerkarchitectuur. Veelal is dit een neuraal netwerk en veel data waarbij de machine leert van trainingsdata. Het uiteindelijke deep learning algoritme wordt vervolgens getest om het patroon te identificeren in een compleet nieuwe dataset: testdata. Met betrekking tot computervisie bestaan er twee soorten deep learning: enerzijds is er *supervised learning* waarbij de data gelabeld is en in het andere geval spreekt men over *unsupervised learning* [24].

In de volgende onderdelen worden benaderingen aangehaald hoe objecten in een scène kunnen gedetecteerd worden.

5.1.1 Convolutioneel neuraal netwerk

Een voorbeeld van een DNN is een convolutioneel neuraal netwerk ofwel een *Convolutional Neural Network* (CNN) of *ConvNET*. Dit is een klasse van diepe, voorwaartse artificiële neurale netwerken en wordt meestal toegepast bij het analyseren van visuele informatie. De data stroomt van een input layer naar een output layer zonder enige vorm van feedback. De eerder vermelde Figuur 18 is een voorbeeld hiervan. Het netwerk wordt gezien als het meest krachtige en effectieve DNN om objecten te herkennen in beelden. Aan elk van de virtuele knopen van het virtueel netwerk, neuronen genoemd, worden gewichten toegekend die verbonden zijn met elkaar. De gewichten en ingangen worden vermenigvuldigd en geven een uitgang tussen 0 en 1. Indien het netwerk een bepaald patroon niet nauwkeurig herkent, past een algoritme de gewichten aan tijdens het trainen, zoals aangegeven in [25]. Op die manier kan het algoritme bepaalde parameters invloedrijker maken, totdat het de juiste wiskundige manipulatie bepaalt om de gegevens uiteindelijk volledig te verwerken.

Het doel van dit soort netwerken is om visuele patronen direct te herkennen uit de pixels van een afbeelding en dit met minimale voorbewerking [26]. In principe kunnen deze netwerken gezien worden als een soort neuraal netwerk dat veel exacte kopieën van eenzelfde neuron bevat en gebruikt. Dit is gelijksoortig bij het programmeren waarbij een functie op veel plaatsen aangeroepen kan worden, wat de kans op fouten vermindert en de efficiëntie verbetert.

In het volgend onderdeel wordt dieper ingegaan op objectdetectie met oog op het convolutioneel neuraal netwerk.

5.1.2 Objectdetectiealgoritmen

Bij objectdetectie wil men niet alleen weten of er een specifiek object al dan niet aanwezig is, ook de exacte positie van dit object in het beeld aangegeven door een *bounding box* is vereist. Objectherkenning maakt innovatieve systemen mogelijk, zoals autonome auto's en robots die afhankelijk zijn van *machine learning* en deep learning algoritmen. Het herkennen en lokaliseren van objecten zoals mensen, auto's, gezichten, ... in afbeeldingen of video's wordt daarom ook wel als het kernprobleem van computervisie beschouwd. Één beste benadering om dit probleem aan te pakken bestaat er echter niet. De keuze voor de juiste objectdetectiemethode hangt af van de opzet van het experiment en het probleem dat moet worden opgelost.

De detectiealgoritmes starten algemeen met het extraheren van een set van robuuste kenmerken van de ingangsafbeeldingen. De meest gekende kenmerkbeschrijvers betreffen: Haar-kenmerken voor gezichtsdetectie [27], *Scale Invariant Feature Transform* (SIFT) om lokale kenmerken te detecteren en te beschrijven in beelden [28], *Histogram of Oriented Gradients* (HOG) om objectherkenning op basis van de verdeling van lokale intensiteitsgradiënten te volbrengen [29] en convolutionele kenmerken [30].

Er worden vervolgens classifiers of *localizers* gebruikt om objecten te identificeren in de functionele ruimte. Deze classificeerders of localiseerders zitten vervat in de lagen van het neuraal netwerk waar ze aan classificatie, ingangsdata toewijzen aan een bepaalde categorie, of lokalisatie doen. Deze worden uitgevoerd ofwel aan de hand van een *sliding window* over de gehele afbeelding ofwel een deelverzameling van gebieden in het beeld. De twee aspecten van objectdetectie, objectclassificatie en -lokalisatie, worden aangegeven in Figuur 19. In de beschouwde figuur moet er een classificatie van het object uitgevoerd worden. In dit geval betreft dit een auto; andere mogelijke klassen zijn: achtergrond, voetganger, fiets, motor, ... Naast het detecteren van een bepaalde klasse in een afbeelding wordt aangegeven waar dit object zich precies bevindt in de afbeelding: lokalisering. Een detectiealgoritme zal daarbij een bounding box rond het object in de afbeelding plaatsen.

Een begrenzingskader wordt gedefinieerd door het convolutioneel netwerk door de



(a) Objectclassificatie van een auto



(b) Objectlokalisatie aan de hand van een bounding box

Figuur 19: Aspecten met betrekking tot objectdetectie [31]

parameters b_x en b_y dewelke het middelpunt van dit kader aangeven. De parameters b_h en b_w betreffen respectievelijk de hoogte en de breedte van de bounding box. De uitgang van een objectherkenningsalgoritme heeft conceptueel in dit voorbeeld een uitgang y onder de vorm van een matrix zoals aangegeven in formule (4).

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad (4)$$

P_c geeft aan of er een object in de afbeelding aanwezig is of niet. Indien deze de waarde ‘1’ heeft dan is er een object aanwezig. In het andere geval betreft deze waarde ‘0’ (er is alleen achtergrond aanwezig, alle andere waarden uit de matrix zijn dan *don't care*). Het symbool C geeft de klasse aan, meer bepaald in het geval van drie klassen: voetganger (C_1), auto (C_2) en fiets (C_3). In het geval er geen object is, zullen al deze de C -waarden ‘0’ zijn. In het andere geval bevat deze een probabiliteitscore die de kans aangeeft dat dit object tot elke klasse C_i behoort. Een bounding box omvat bijgevolg de coördinaten van een rechthoek die een object in een afbeelding volledig omsluit wanneer deze over een tweedimensionale achtergrond wordt geplaatst.

In de komende paragrafen worden enkele algoritmen besproken die objectdetectie mogelijk maken. Deze betreffen *Deformable Part Models* (DPM), de *Region-based Convolutional Neural Networks* (RCNN) familie en *You Only Look Once* (YOLO).

5.1.2.1 *Deformable Parts Model*

Deformable Parts Model, afgekort DPM, gebruikt een sliding window om objectdetectie te realiseren. Het maakt gebruik van een disjuncte pipeline om bepaalde taken te volbrengen. Deze taken betreffen: het extraheren van statische functies, classificeren van regio's en bounding boxes voorspellen voor gebieden met een hoge score met betrekking tot objectdetectie zoals aangegeven in [32].

Bij het principe van een sliding window ofwel glijdend venster, wordt er gestart met het nemen van een bepaald venster met een kleine grootte. Door het venster vervolgens in het convolutioneel netwerk overheen de afbeelding te laten 'glijden' wordt er een uitgang y bekomen. Deze uitgang geeft aan of er een object gedetecteerd is of niet. De classifier wordt bijgevolg uitgevoerd op gelijk van elkaar liggende locaties over de volledige afbeelding. Deze procedure wordt nadien herhaald, maar er wordt een groter rechthoekig venster gebruikt.

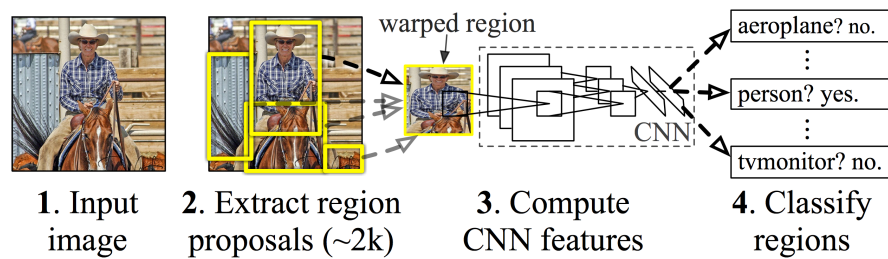
Het nadeel van sliding window detection is de berekeningskost. Er moeten zeer veel verschillende regio's overlopen worden in de afbeelding die elk onafhankelijk van elkaar verwerkt worden in een ConvNET. Als gevolg hiervan is deze methodiek erg traag en presteert niet goed in realtime omstandigheden. In onderdeel 5.1.2.3 wordt besproken hoe het YOLO-algoritme de voorspelling van een bounding box kan versnellen.

5.1.2.2 *Region-based Convolutional Neural Network familie*

- **RCNN**

Het *Region based Convolutional Neural Network* gebruikt *Selective Search* (SS) om potentiële begrenzingskaders te voorspellen in een afbeelding, zoals uiteengezet in [33]. Op elk van deze mogelijke bounding boxes wordt nadien een classifier uitgevoerd. Aansluitend wordt er *post-processing* gebruikt om door middel van een lineair model de bounding boxes te herdefiniëren. Deze herdefiniëring houdt naast het elimineren van dubbele detecties ook een herscoring in van de begrenzingskaders op basis van de overige gedetecteerde objecten in het beeld. Het elimineren van duplicaten gebeurt door het toepassen van een *non-maximum suppression* algoritme, wat het YOLO-algoritme uit onderdeel 5.1.2.3 ook gebruikt. Een overzicht van deze RCNN methodiek komt aan bod in [32] en is schematisch weergegeven in Figuur 20. Voor verdere details omtrent RCNN wordt verwezen naar [34].

Het nadeel van deze manier van werken is dat dit algoritme traag is en moeilijk te optimaliseren. De reden hiervoor is dat elke individuele component apart getraind moet worden. In het volgend onderdeel wordt er een snellere versie van RCNN besproken: *Fast RCNN*.



Figuur 20: RCNN schematisch voorgesteld [35]

In het onderdeel 5.1.2.3 wordt besproken hoe objectdetectie gerealiseerd kan worden door het probleem hiervan te herkaderen als een regressieprobleem, rechtstreeks van de beeldpixels naar de coördinaten van de bounding boxes.

- **Fast RCNN**

Bij Fast RCNN wordt de gehele afbeelding in één keer geconverteerd in tegenstelling tot RCNN waarbij de regio's gemapt worden in de corresponderende kenmerken. Deze snellere versie van RCNN maakt gebruik van een convolutionele implementatie van sliding windows om alle voorgestelde regio's te classificeren, wat in onderdeel 5.1.2.1 tevens is aangehaald. Voor verdere details betreffende dit algoritme wordt verwezen naar [34].

Het nadeel is dat de voorspelling van de regio's nog altijd te traag is om te gebruiken in realtime toepassingen. Om hieraan vooraf te gaan is er een andere groep voorgesteld, meer bepaald het *Faster RCNN*-algoritme.

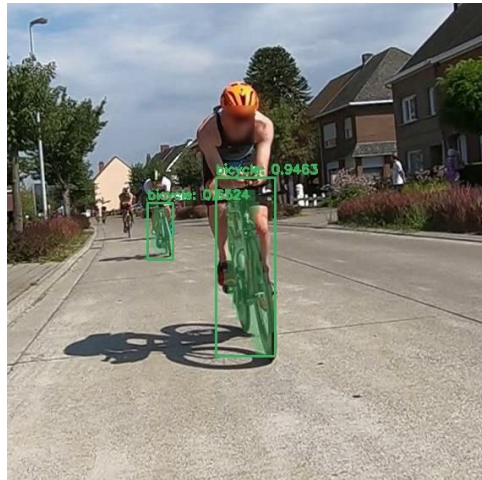
- **Faster RCNN**

Dit algoritme, voorgesteld in [36], gebruikt een convolutioneel netwerk om regio's te voorspellen. Het is bijgevolg een nieuw *Region Proposal Network* (RPN) voor objectlokalisatie. Voor verdere details wordt verwezen naar de literatuur [36].

- **Mask RCNN**

Een recente uitbreiding van Faster RCNN betreft *Mask RCNN* die een tak parallel toevoegt aan de detectiepipeline waardoor er naast een bounding box ook een masker van het object wordt voorspeld. Er wordt kortom aan objectdetectie en semantische segmentatie gedaan. Een voorbeeld toegepast op een beeld uit een triatlonwedstrijd is weergegeven in Figuur 21. Naast de extra overhead als nadeel is het volgens [37] wel handig om bijvoorbeeld menselijke posities waar te nemen, wat in deze scriptie minder van belang is.

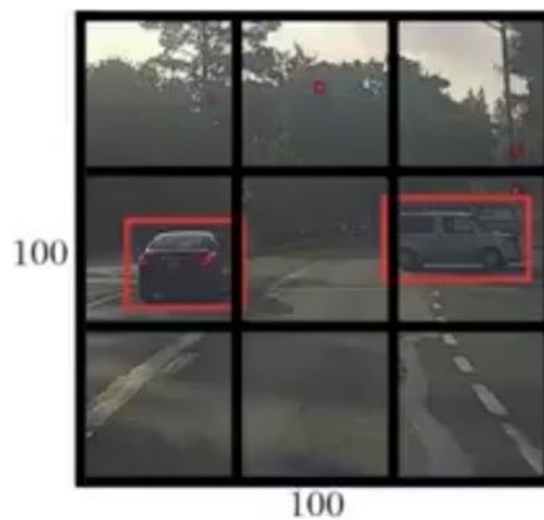
In het volgend onderdeel wordt het YOLO-algoritme uiteengezet. Deze presteert met oog op realtime doeleinden beter dan de tot nu toe eerder aangehaalde methoden voor objectdetectie.



Figuur 21: Mask RCNN toepassing op een beeld

5.1.2.3 *You Only Look Once*

In de overig aangehaalde methoden worden er altijd eerst voorgestelde regio's gezocht om vervolgens de objecten te classificeren. Het YOLO-algoritme kijkt slechts één keer, vandaar de naam "You Only Look Once". Het doel van YOLO is om voorspellingen voor de bounding box accurater te verkrijgen. De afbeelding wordt opgedeeld in een rooster van bijvoorbeeld 3×3^1 , zoals aangegeven in het conceptueel voorbeeld in Figuur 22.



Figuur 22: Voorbeeld van YOLO - 3×3 rooster [31]

Het classificatie en -lokalisatie algoritme wordt op elk van de negen cellen toegepast.

¹Veelal wordt er een veel groter rooster genomen, bijvoorbeeld 19×19 . Het 3×3 rooster is conceptueel makkelijker te begrijpen als voorbeeld.

Meer bepaald gaat YOLO voor elke roostercel twee bounding boxes voorspellen, ook wel *anchor boxes* genoemd. Het voordeel hiervan is dat het algoritme als gevolg van het roosterprincipe en de twee anchorboxes, in vergelijking met één of meerdere sliding window(s) over alle pixels van het beeld, erg snel is en geschikt is voor realtime toepassingen [32].

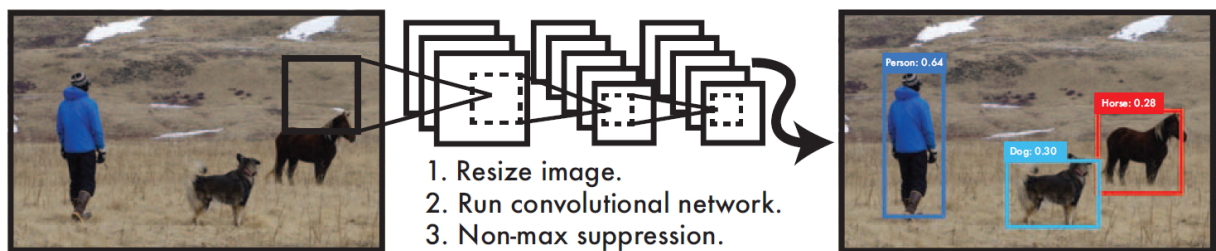
Aan elk begrenzingskader wordt een score toegekend die aangeeft met hoeveel waarschijnlijkheid het object in het begrenzingskader bijvoorbeeld een auto is. Dit wordt de *annotation* of *confidence score* genoemd. Deze vertrouwensscore geeft aan hoe zeker het model is dat het begrenzingskader een object bevat én hoe nauwkeurig het voorspelde kader is.

Om de nauwkeurigheid van een objectdetector op een gelabelde testset aan te geven kan de *Intersection over Union* (IOU) worden gebruikt. Het wordt gedefinieerd volgens formule (5).

$$\text{IOU} = \frac{\text{overlappingsgebied}}{\text{gebied}_{\text{voorspelde begrenzingskader}} \cdot \text{gebied}_{\text{grondwaarheid begrenzingskader}}} \quad (5)$$

Hoe groter de IOU-waarde, hoe accurater het begrenzingskader is. Een waarde groter dan of gelijk aan 0,5 wordt gezien als een “goede voorspelling”. Deze vier waarden van een bounding box, de x- en y-coördinaat en de breedte en hoogte, worden bij het YOLO-algoritme relatief aan de resolutie van het volledige beeld gekozen en bijgevolg niet aan een bepaalde cel.

Een voorbeeld van de output van een afbeelding door het YOLO-algoritme verwerkt betreft Figuur 23, die tevens de stappen van het algoritme omvat.



Figuur 23: Voorbeeld van YOLO - begrenzingskader met *annotation* [32]

Het YOLO-algoritme moet kunnen leren en doet dit op basis van trainingsdata. In dit geval zijn dit afbeeldingen met een grondwaarheid waarbij het algoritme de modelstructuur, het filter en de vervormingskosten leert. Voor de trainingsdata bestaan er verschillende *opensource* datasets zoals *KITTI* [38]. Er kan ook gebruikgemaakt worden van een eigen opgestelde dataset.

Het grote voordeel van YOLO is dat dit algoritme in vergelijking met de eerder

besproken methoden significant sneller is. De throughput van het systeem is 45 *frames per second* (fps), getest met de *NVIDIA GeForce Titan X Pascal Graphics processing unit* (GPU), in vergelijking met een fps van 7 bij Faster RCNN zoals aangegeven in [32]. De reden voor deze snelheidsverbetering is dat YOLO globaal redeneert over de afbeelding bij het maken van voorspellingen, in tegenstelling tot de DPM- en RCNN-technieken die alleen een bepaalde regio analyseren en bijgevolg niet het totale beeld in één keer. YOLO encodeert impliciet de samenhangende informatie over de klassen en hun voorkomen in de afbeelding. Kortom is het inzicht achter YOLO, in vergelijking met de eerder vernoemde netwerken, dat het predictienetwerk van de bounding box kan worden gecombineerd met het classificatienetwerk in één enkele netwerkarchitectuur.

Naast het recent YOLOv3-netwerk dat een afweging maakt tussen nauwkeurigheid en snelheid, bestaat er ook *Tiny YOLO* dat gericht is voor realtime applicaties voor *constraint devices* en tot 220 fps op de Titan X GPU kan halen volgens [39]. Het nadeel is daarentegen dat de nauwkeurigheid daalt. Bij testing op de *Common Objects in Context* (COCO) testset met de Titan X wordt een *mean Average Precision* (mAP) voor een IOU van 0,5 (mAP-50) van 23,7 % bekomen voor Tiny YOLOv3 in vergelijking met 57,9 % met een fps van 20 in het geval van YOLOv3 [40].

Een ander veelbelovend detectienetwerk voor apparaten met beperkte rekenkracht betreft *Single Shot Detector* (SSD) met een gelijkaardig detectieprincipe als YOLO. Voor meer uitleg en details wordt verwezen naar de aangehaalde literatuur [41].

5.2 Benchmarking van fietsdetectienetwerken

Er worden benchmarks uitgevoerd met Faster RCNN, Mask RCNN, YOLOv3, Tiny YOLOv3 en SSD op de testset². De performantie van een neuraal netwerk wordt beoordeeld door gebruik te maken van het mean Average Precision (mAP) criterium gedefinieerd in de *PASCAL Visual Object Classes* (VOC) *challenge*, die een *benchmark* is in herkenning en detectie van visuele objectcategorieën [42]. In deze applicatie is er slechts één klasse namelijk ‘bicycle’ en wordt bijgevolg alleen de AP beschouwd. Voor de verschillende netwerken is de Average Precision (AP) voor een IOU van 50 % uitgezet ten opzichte van de detectiesnelheid uitgedrukt in aantal frames per seconde (fps). Er wordt hiervoor Tabel 1 met bijhorende Figuur 24 beschouwd.³ De verschillende precision en recall curves voor de vijf gevallen bevinden zich in Bijlage C. De basisnetwerken hebben verschillende CNN-architecturen zoals: MobileNet (SSD),

²Deze betreft in dit geval de Swimkap triatlonwedstrijd bestand 49 van de dataset.

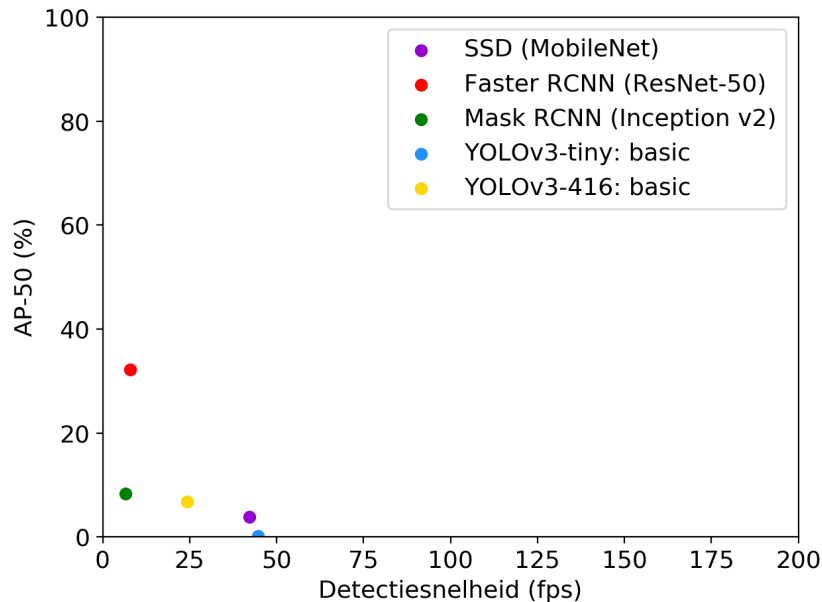
³Deze benchmarks en alle volgende objectdetectortesten zijn uitgevoerd op een Graphics Processing Unit (GPU) van het type *MSI GeForce GTX 1060 6GT*.

ResNet (Faster RCNN), Inception ResNet (Mask RCNN), Darknet (YOLOv3 en Tiny YOLOv3). Andere zijn VGGNET, DenseNet, AlexNet, ... [43]. MobileNet is een architectuur ontwikkeld voor constrained apparaten zoals een smartphone [44].

Tabel 1: AP-50 en detectiesnelheid voor verschillende basis objectdetectienetwerken

Detectienetwerk	AP-50 (%)	Detectiesnelheid (fps)
Single Shot Detector 300 (MobileNet)	3,85	42,25
Faster RCNN (ResNet-50)	32,20	7,85
Mask RCNN (Inception v2)	8,32	6,64
YOLOv3-tiny: basic	0,20	44,57
YOLOv3-416: basic ¹	6,84	24,19

¹ 416 wijst op de resolutie van het netwerk die door de beperking van de gebruikte GPU voor 416x416 werkt in plaats van de gebruikelijke 608x608. Het nadeel is dat de nauwkeurigheid daalt, maar er wordt een kleine snelheidswinst geboekt [40]. Voor YOLOv3-tiny wordt ook deze resolutie gebruikt.



Figuur 24: AP-50 versus detectiesnelheid voor verschillende basis objectdetectienetwerken

De gemiddelde precisie ofwel AP komt overeen met de oppervlakte onder de precision recall curve. De precisie wordt voor deze applicatie gedefinieerd als de fractie van de

gedetecteerde objecten die effectief een (triatlon)fiets zijn. In formulevorm wordt dit gegeven volgens formule (6), ook wel *Positive Predictive Value* (PPV) genaamd [45].

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

In deze beschouwde formule betreft TP het resultaat waar het detectiemodel correct een fiets heeft gedetecteerd, ofwel *True Positive*. *False Positive* (FP) is de fractie van het aantal gedetecteerde objecten die incorrect zijn voorspeld als een fiets.

Er kan daarnaast ook gekeken worden naar het aantal True Positives in vergelijking met het totaal aantal gedetecteerde objecten (TP + FN). FN, ofwel False Negatives, is het resultaat van het model waar deze incorrect de afwezigheid van een fiets heeft voorspeld. De verhouding van TP en (TP + FN) betreft de *recall* van de objectdetector, ook wel de True Positive Rate (TPR) genoemd, zie hiervoor formule (7) [45].

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

Zoals in de beschouwde Figuur 24 kan worden afgeleid, is de gemiddelde precisie voor de klasse ‘bicycle’ erg laag. In het geval de AP relatief hoog is, namelijk een maximum van 32,20 % bekomen voor Faster RCNN, dan blijkt dat de detectiesnelheid met 7,85 frames per seconde te laag is. Aangezien in deze applicatie de draftingdetectie in realtime moet gebeuren, is dit ongewenst. De beelden uit de dataset zijn opgenomen aan 30 fps, wat overeenkomt met de minimaal gewenste *throughput*. De gebruikte Single Shot Detector werkt op een model voor ingebedde applicaties, MobileNet, en heeft een detectiesnelheid van 42,25 fps maar heeft slechts een precisie van 3,85 %⁴. Een snellere mogelijkheid betreft Tiny YOLOv3 met een detectiesnelheid van 44,57 frames per seconde.

De *confusion matrix*, ofwel verwarringsmatrix, representeert de prestaties van het algoritme op de testset [46] zoals weergegeven in Tabel 2.

Er kan bovendien gekeken worden naar de *Log Average Miss Rate* (LAMR). Volgens [47] wordt deze berekend door de *Miss Rate* (MR) te berekenen over negen *False Positives Per Image* (FPPI) rates die even ver van elkaar liggen in de log-ruimte over het bereik van 0,01 tot 100. MR wordt gedefinieerd volgens formule (8).

$$\text{MR} = \frac{FN}{TP + FN} \quad (8)$$

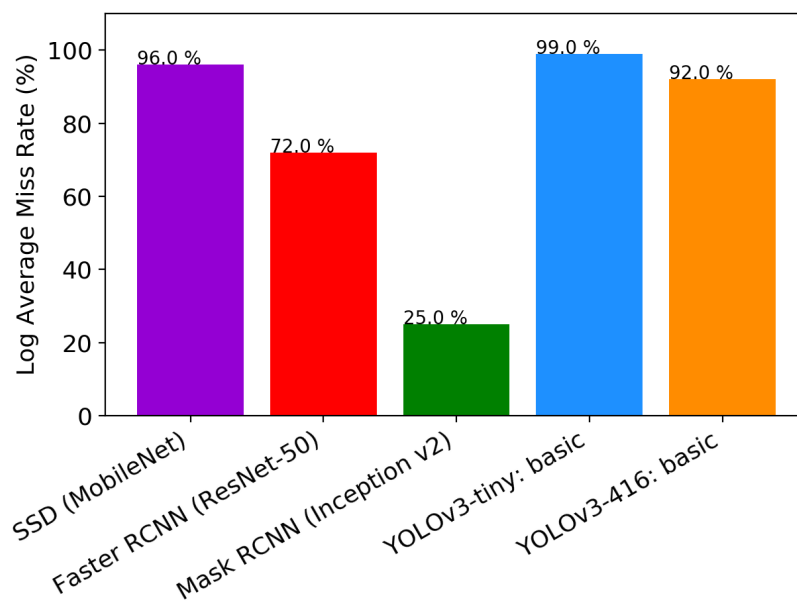
LAMR is gelijkaardig aan de eerder geëvalueerde AP, maar geeft volgens [47] een meer informatieve en stabiele schatting van de prestaties van een objectdetector. De reden hiervoor is dat in bepaalde applicaties zoals het voortdurend volgen van fietsers er een bovengrens is op het aantal False Positives per beeld onafhankelijk van de dichtheid van

⁴Ondanks dat SSD naast Tiny YOLOv3 ook realtime mogelijkheden biedt maar met een hogere precisie, kan deze dienst doen in een toekomstmogelijkheid zoals in Hoofdstuk 9 wordt beduid.

Tabel 2: Confusion matrix voor het detecteren van fietsen

		Voorspelling	
		Positief	Negatief
Werkelijk	Positief	TP Het model heeft correct de aanwezigheid van een fiets voorspeld.	FN Het model heeft incorrect de afwezigheid van een fiets voorspeld.
	Negatief	FP Het model heeft incorrect de aanwezigheid van een fiets voorspeld.	TN Het model heeft correct de afwezigheid van een fiets voorspeld.

de objecten (bijvoorbeeld fietsers die naast elkaar rijden). De LAMR vat met andere woorden de MR versus FPPI-curve samen in één waarde. Figuur 25 geeft de LAMR weer voor de verschillende eerder vermelde objectdetectoren. Uit het beschouwde



Figuur 25: Log Average Miss Rate voor verschillende basis objectdetectienetwerken

staafdiagram kan afgeleid worden dat de Mask RCNN-objectdetector voor de klasse 'bicycle' de laagste LAMR heeft, met name 25,0 %, wat impliceert dat deze relatief weinig False Negatives heeft (volgens de grondwaarheid is er een fiets doch wordt er geen fiets gedetecteerd). Zoals afgeleid kan worden uit Figuur 24 is de detectiesnelheid

echter te laag voor realtime doeleinden. Om een goede fietsdetector te verkrijgen is het doel om een netwerk te hebben met een grote (gemiddelde) precisie, minimale throughput van 30 fps, hoge recall en overeenkomstig een lage LAMR. Om dit te bewerkstelligen is er training nodig wat in het volgend onderdeel aan bod komt.

5.3 Benchmarking fietsobjectdetectie na training

In dit geval is er gekozen om het YOLOv3 detectienetwerk en zijn kleinere variant Tiny YOLOv3 te trainen. Enerzijds omdat beide relatief gemakkelijk kunnen getraind worden. Anderzijds is het voordelig voor toepassingen wegens de efficiëntie en real-time prestaties. Meer bepaald hebben deze respectievelijk in de basisversie een detectiesnelheid van 25,19 en 44,57 fps, wat het realtime gedrag benadert aangezien 20 fps volstaat voor de applicatie. Er is getraind aan de hand van beelden uit triatlonwedstrijden van de dataset aangezien een triatlonfiets er meestal anders uitziet dan een gewone (koers)fiets. Figuur 26 toont dit visueel aan.



(a) Triatlonfiets met een laag stuur, opzetstuur en vol achterwiel [48]



(b) Koersfiets met een normaal stuur en achterwiel [49]

Figuur 26: Twee voorbeelden van een triatlonfiets

Aangezien de camera achteraan het zadel is geplaatst, is de voorkant van een fiets zichtbaar en moet bijgevolg gedetecteerd worden zoals weergegeven in Figuur 27. Indien een fietser voorbijsteekt of wordt voorbijgestoken zoals in Figuur 16, moet deze ook gedetecteerd worden aangezien bijvoorbeeld in het laatste geval deze zich binnen de vijf seconden uit de draftingzone moet begeven.

Er wordt dienvolgens getraind op de voorkant en zijkant van een fiets waarbij het stuur en het voorwiel en eventueel achterwiel zichtbaar zijn. Eerst is YOLOv3 getraind met 20000 annotaties. Er is vervolgens overgegaan naar een trainingssessie met driemaal zoveel annotaties voor zowel YOLOv3 als de Tiny YOLOv3 variant. Dit valt onder



Figuur 27: Voorbeeld van een opgenomen beeld

supervised learning daar de trainingset is gelabeld.

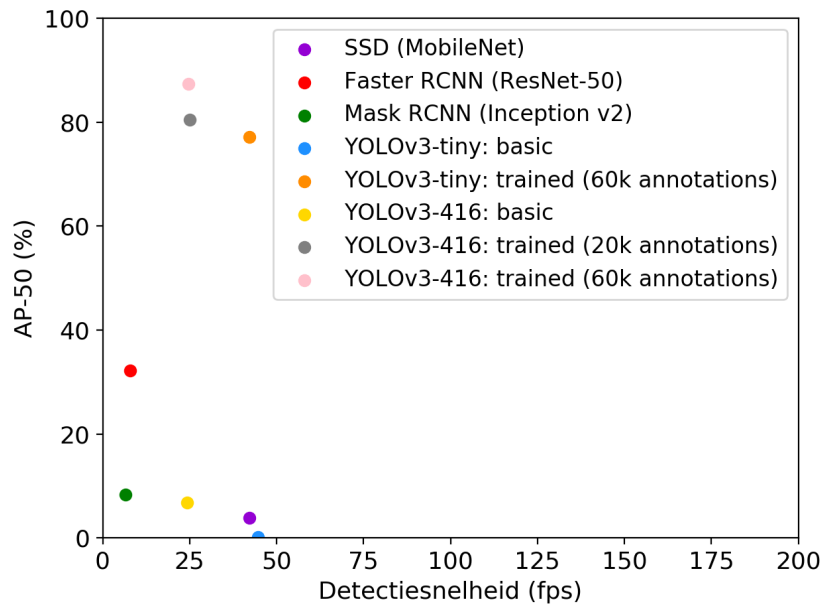
De bekomen waarden voor de detectiesnelheid en gemiddelde precisie voor een IOU van 50 voor de drie gevallen zijn weergegeven in Tabel 3. De nieuwe gemiddelde precisie curve versus de detectiesnelheid is weergegeven in Figuur 28 voor de drie getrainde netwerken.

Tabel 3: AP-50 en detectiesnelheid voor getrainde YOLOv3- en Tiny YOLOv3-objectdetectienetwerken

Detectienetwerk	AP-50 (%)	Detectiesnelheid (fps)
YOLOv3-tiny: trained (60k annotations)	77,19	42,25
YOLOv3-416: trained (20k annotations)	80,41	25,01
YOLOv3-416: trained (60k annotations)	87,35	24,57

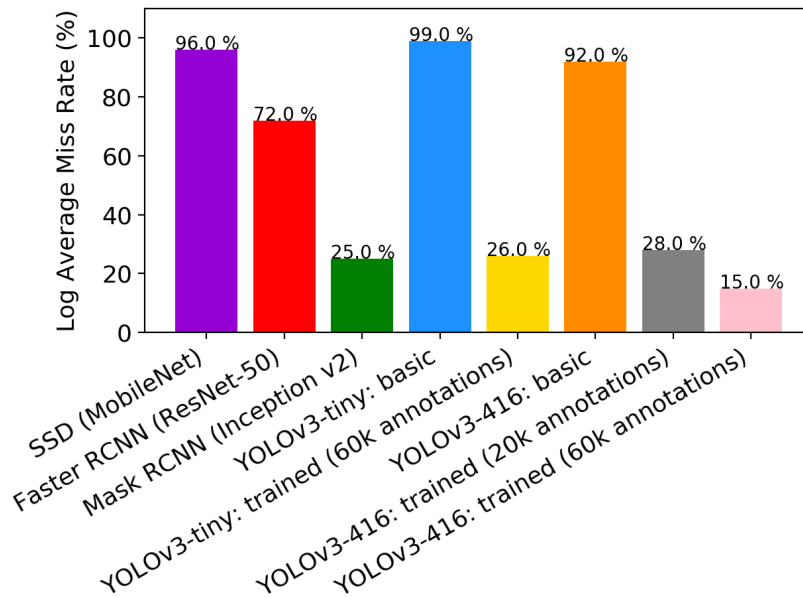
Uit de beschouwde tabel en figuur kan bemerkt worden dat de Average Precision na training is verbeterd, tot 87,35 % in het geval van YOLOv3 bij training met 60 000 annotaties ('YOLOv3-416: trained (60k annotations)').

Analoog aan de vorige paragraaf wordt gekeken naar de LAMR voor de verschillende objectdetectoren, zie hiervoor Figuur 29, dan kan er bemerkt worden dat de Log Average Miss Rate voor de getrainde versies met 60 000 annotaties van YOLOv3 en



Figuur 28: AP-50 versus detectiesnelheid voor verschillende objectdetectienetwerken samen met drie getrainde YOLO-netwerken

Tiny YOLOv3 gedaald is tot 15,0 % respectievelijk 26,0 %.



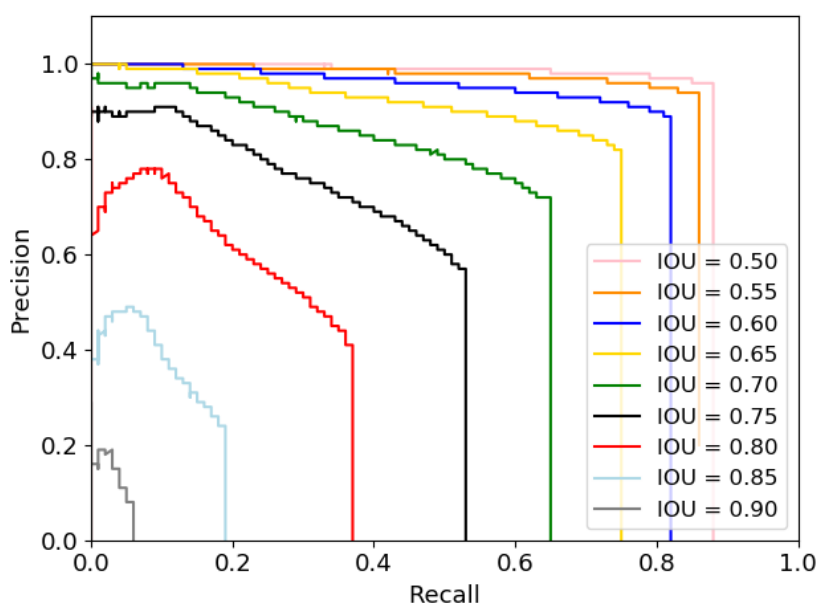
Figuur 29: Log Average Miss Rate voor verschillende basis objectdetectienetwerken samen met drie getrainde YOLO-netwerken

Bij benadering is de LAMR gelijk aan of kleiner dan deze van het basis Mask RCNN

netwerk, maar met een hogere detectiesnelheid van respectievelijk 42,25 % en 24,57 %. De precision recall curve van beide getrainde YOLO-netwerken in het geval van de trainingssessies met 60 000 annotaties, komt in de volgende subparagraaf aan bod.

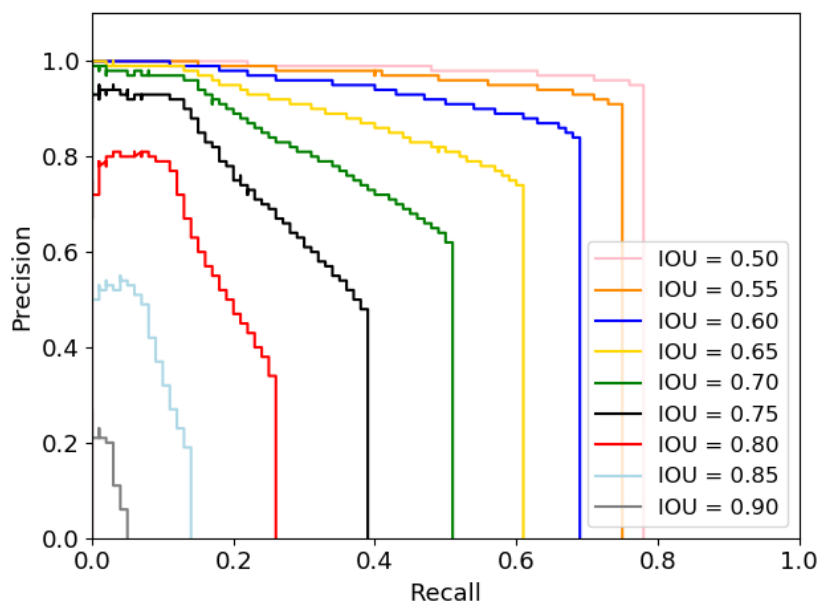
5.3.1 Precision recall curves

Eerst wordt een precision recall curve opgesteld voor YOLOv3 die weergegeven is in Figuur 30. Er worden verschillende IOU-grenzen beschouwd van 50 % tot en met 90 %. Voor Tiny YOLOv3 wordt tevens een precision-recall curve opgesteld voor verschillende



Figuur 30: Precision recall van een YOLOv3-netwerk voor verschillende IOU-waarden

IOU-waarden van 0,5 tot en met 0,9, zie hiervoor Figuur 31. Tabel 4 toont een vergelijking van de AP voor verschillende IOU's voor het YOLOv3 en Tiny YOLOv3 getraind netwerk. Figuur 32 betreft de combinatie van Figuur 30 en Figuur 31 om beide objectdetectoren visueel te vergelijken met betrekking tot hun prestaties: YOLOv3 in ononderbroken lijnen en Tiny YOLOv3 in stippelijnen. Voor een IOU-threshold van 50 % wordt een AP van 87,35 % en 77,19 % bekomen voor respectievelijk YOLOv3 en Tiny YOLOv3, een verschil van 10,16 %. Als er gekeken wordt naar de detectiesnelheid volgens Tabel 3 dan is de Tiny variant 1,72 keer sneller dan de gewone YOLOv3. Aangezien de keuze voor realtime detectie van cruciaal belang is, wordt gekozen om als fietsdetector de getrainde versie van Tiny YOLOv3 te gebruiken. Het verlies in LAMR van 11,0 % in vergelijking met de gewone YOLOv3 variant wordt gepoogd op te heffen door middel van robuuste objecttracking wat in het volgend hoofdstuk aan bod komt.

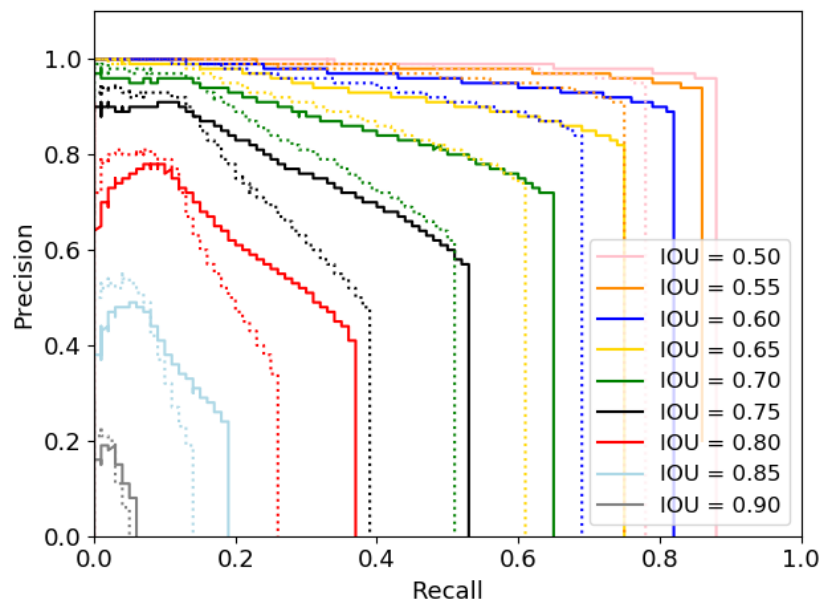


Figuur 31: Precision recall van een Tiny YOLOv3-netwerk voor verschillende IOU-waarden

Tabel 4: De AP voor YOLOv3 en Tiny YOLOv3 voor verschillende IOU-thresholdwaarden.

IOU-threshold	AP (%) YOLOv3	AP(%) Tiny YOLOv3	Vershil in AP (%)
0,50	87,35	77,19	10,16
0,55	84,69	72,88	11,81
0,60	79,08	65,52	13,52
0,65	70,18	54,70	15,48
0,70	56,78	42,67	14,11
0,75	41,18	29,83	11,35
0,80	23,62	16,98	6,64
0,85	7,62	6,05	1,57
0,90	0,92	0,81	0,11

Naast de AP, wat een samenvatting is van de precision versus recall curve, is de precisie in deze applicatie van groot belang. Meer bepaald wordt in Hoofdstuk 7 de hoogte van de bounding box, hetzij de ligging van de onderkant van de bounding box gebruikt om de afstand tot een achterligger te bepalen. De nauwkeurigheid van de bounding boxes wordt tevens gebenchmarkt in het volgend hoofdstuk. De reden hiervoor is dat de objecttracker tevens voor een bepaalde nauwkeurigheid instaat met betrekking tot het volgen van een bounding box. Er wordt alzo een grondige evaluatie uitgevoerd van de



Figuur 32: Precision recall van het getrainde YOLOv3- (ononderbroken lijnen) en Tiny YOLOv3-netwerk (stippellijnen) voor verschillende IOU-waarden

snelheid en de accuraatheid van de bounding box daar dit van belang is voor de afstandsbepaler in Hoofdstuk 7.

Hoofdstuk 6

Objecttracking en trackbeheer

In dit hoofdstuk wordt een verdere analyse gemaakt om realtime en met een hoge precisie fietsers te volgen. Daar geen enkele objectdetector perfect is en er bijgevolg valse positieven en valse negatieven optreden, wordt in dit hoofdstuk gepoogd om de dichtstbijzijnde gedetecteerde fiets te volgen in de tijd door een objecttracker te gebruiken. Bij objecttracking wordt het bewegingstraject van een object gevolgd. In een eerste onderdeel wordt het principe en de motivatie hiervan besproken met oog op het draftingdetectiesysteem. In een tweede onderdeel worden objecttrackingmethodieken uit de literatuur uiteengezet. Aansluitend worden in een derde paragraaf enkele objecttrackers gebenchmarkt op de testset die hiervoor kunnen gebruikt worden. De dichtstbijzijnde fietser wordt doorheen de verschillende scènes gevolgd. Om het bijhorende correspondentieprobleem op te lossen, wordt *trackmanagement* ofwel trackbeheer besproken.

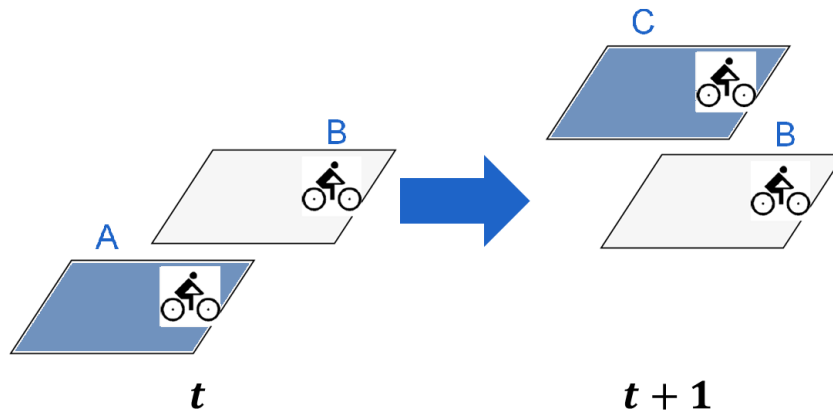
6.1 Motivatie voor objecttracking

Objecttracking komt neer op het volgen van één of meerdere gelokaliseerde objecten, in dit onderzoek voor een realtime toepassing. Het doel hierbij is om het traject van een object, met name de bounding box, te schatten in het afbeeldingsvlak terwijl dit object beweegt in de scène. De algoritmes die deze vorm van tracking implementeren, analyseren een reeks beelden en geven de beweging van doelobjecten in deze beelden als uitgang terug. Meer bepaald wijst een “tracker” consequent labels toe aan de getrackte objecten in de verschillende frames van een video.

Een eerste reden om aan objecttracking te doen, die in deze applicatie minder van belang is, betreft het volgen van meerdere fietsers. Er is gekozen om alleen de dichtstbijzijnde fietser te volgen, maar er kunnen ook meerdere gedetecteerd en getrackt worden wat verder besproken wordt in Hoofdstuk 9 met betrekking tot toekomst- en vervolgmogelijkheden.

Een tweede reden voor het gebruik van een objecttracker is om de miss rate te laten dalen. Door een continue monitoring te realiseren wordt op die manier getracht om de geschatte positie van de dichtstbijzijnde fietser doorheen de verschillende beelden te verbeteren. Zoals in Figuur 33 is weergegeven kan een atleet A die zich rechts van atleet B bevindt in frame (t) zich in een volgend frame ($t + 1$) niet opeens links van B

bevinden. Vermoedelijk gaat het in frame $(t + 1)$ over een andere atleet, C, die B aan het voorbijsteken is. Om die reden wordt elke gedetecteerde fiets uniek geïdentificeerd

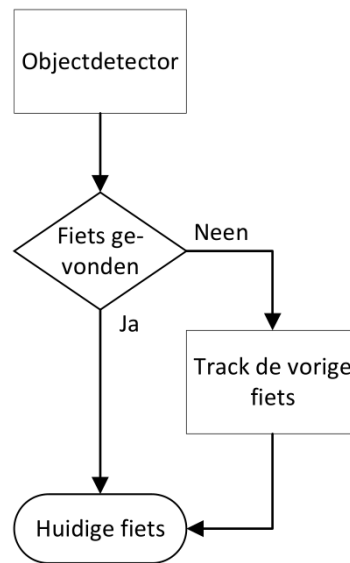


Figuur 33: Situatie waarbij atleet A en C niet dezelfde atleet zijn (eigen bewerking van [1])

door een ‘bicycle id’ aangezien elke fietser een andere draftinggeschiedenis heeft. Anders gezegd, atleet A krijgt een andere id dan atleet C in Figuur 33. Het is tevens van belang dat elke dichtstbijzijnde atleet zo goed mogelijk doorheen de scènes wordt gedetecteerd. Indien een atleet zich uit het beeld bevindt en/of niet gedetecteerd wordt in een bepaald frame, dan wordt deze *unregistered*, met andere woorden wordt zijn draftinghistorie en tijdsmeting (met name de draftingtijd) gereset.

Er wordt aan *visual objecttracking* [50] gedaan op momenten dat de objectdetector geen fiets detecteert, maar er in werkelijkheid wel een fietser aanwezig is in de scène: de False Negatives (FN). Meer bepaald wordt het principe van *tracking-by-detection* [51] van de bounding box toegepast zoals principieel getoond in Figuur 34. Indien er in het vorige frame $(t - 1)$ een fiets is gedetecteerd door de objectdetector ‘vorige fiets’ en in het huidige frame (t) niet, dan wordt deze geschat door deze laatst gedetecteerde te tracken. Met andere woorden, vanaf het moment dat de objectdetector geen fiets kan vinden in de opeenvolgende frames, wordt de bounding box van de laatst gevonden fiets telkens geschat. Het trackingalgoritme wordt hiervoor geïnitieerd met het huidige frame van de videosequentie en een bounding box om de locatie aan te geven van het object dat gevolgd moet worden. Indien de fiets zich echter op een afstand van bijvoorbeeld 25 meter begeeft of een voorbijsteekgedrag vertoont en zich uit de gefilmde scène bevindt, wordt de objecttracker gereset. Op die manier ontstaat er een continue monitoring. Dit zal resulteren in een beter trackmanagement en meer samples voor het bepalen van de afstand, wat aan bod komt in Hoofdstuk 7.

Voor de benodigde objecttracker voor het draftingdetectiesysteem zijn een aantal



Figuur 34: Tracking-by-detection principe voor het detecteren van fietsers

vereisten van fundamenteel belang, welke in het volgend onderdeel worden besproken.

6.1.1 Objecttrackervereisten

Om een goede keuze te maken van de objecttracker is het belangrijk dat deze toepasbaar is in de omgevingen en situaties van triatlonwedstrijden zoals eerder aangehaald in Hoofdstuk 4.

- Het in realtime volgen van fietsers doorheen de scènes is essentieel, bijgevolg is *offline learning* geen mogelijkheid wat wel het geval is voor *online learning*. Beide begrippen worden in het volgend onderdeel besproken.
- Aangezien de afstand tot de achterligger verandert door het vertragen of versnellen van de fietsers, verandert ook de grootte van de bounding box. Er is bijgevolg nood aan schaalvariatie.
- Zoals reeds vermeld kunnen er belichtingsveranderingen optreden in de scènes als gevolg van invallend zonlicht en het in- en uitrijden van een tunnel.
- De camera is voor deze applicatie gemonteerd aan het zadel van de fiets. Als gevolg hiervan treden er abrupte bewegingen op in het beeld door bijvoorbeeld het heen en weer schommelen van de fiets, een slecht wegdek, het nemen van een bocht, ... De tracker moet het object aldus nauwkeurig kunnen volgen in al deze gevallen.
- Bovendien is ook de nauwkeurigheid van de getrackte bounding box belangrijk aangezien de afstand wordt bepaald conform de hoogte of hoogteligging van deze rechthoek.

In een volgend onderdeel wordt een inleiding omtrent objecttracking gegeven naast de mogelijkheden en technieken om dit te implementeren.

6.2 Tracking van objecten - literatuur

In dit onderdeel wordt dieper ingegaan op het volgen of *tracken* van objecten in een reeks videoframes. Bovendien wordt er gefocust op het volgen van meerdere mogelijke objecten in de scène en in onderdeel 6.2.1 worden enkele uitdagingen aangehaald die invloed hebben op de robuustheid om een object te volgen.

Objecttracking is tegenwoordig een belangrijk onderdeel binnen de computervisie, mede mogelijk gemaakt door de proliferatie van high-end computers en de beschikbaarheid van goedkope camera's met een hoge kwaliteit. Er is bovendien ook een stijgende nood aan geautomatiseerde videoanalyse zoals in intelligente visuele bewakingssystemen ofwel *Intelligent Visual Surveillance* (IVS) besproken in [52]. Zo'n systeem realiseert automatische visuele monitoring die naast objectdetectie en -tracking het gedrag van objecten analyseert en interpreteert om visuele gebeurtenissen in het beeld te begrijpen. Het doel van beeldinterpretatie bestaat erin om op hoog niveau informatie van een visuele gebeurtenis uit een dynamisch beeld te extraheren. Naast toepassingen in de beveiliging speelt objecttracking ook een belangrijke rol in het realtime monitoren van het verkeer. In het volgend onderdeel wordt objecttracking besproken met oog op een bewegende camera, wat in dit onderzoek het geval is.

6.2.1 Uitdagingen objecttracking

Er zijn verschillende mogelijke redenen die het trackingresultaat niet nauwkeurig genoeg maken zoals: lage videokwaliteit, allerhande ruisinvloeden, een te klein object, abrupte of complexe objectbewegingen, de bewegingen van de camera (die in deze applicatie niet stationair is vooropgesteld), veranderingen in de belichting enzovoort. Er is daarenboven een verlies van informatie veroorzaakt door de projectie van het wereldcoördinatensysteem naar een 2D-afbeelding. Een overig belangrijk aspect waarmee rekening gehouden moet worden, beschreven in [53], is het effect van objectocclusie. Dit houdt in dat objecten in een beeld van elkaar onderscheiden moeten worden indien een object overlapt wordt door een ander object (bijvoorbeeld twee fietsers die zich achter elkaar bevinden, waarbij één fietser dus slechts gedeeltelijk zichtbaar is). Dit laatste speelt een belangrijke rol bij multitracking.

Er kunnen volgens [54] drie categorieën onderscheiden worden voor objecttracking: modelgebaseerde *backgroundsubtraction*, trajectclassificatie en objecttracking. In het eerstgenoemde wordt er een model opgesteld van de achtergrond door een bewegingscompensatie toe te passen. In het geval van trajectclassificatie wordt het

traject van een object bepaald door een clusterbenadering toe te passen van feature punten om de trajecten die tot hetzelfde object behoren af te trekken van een achtergrondmodel. Bij de laatste categorie wordt een object gemarkeerd door een objectdetector, om dit object nadien in de volgende frames van de videosequentie te volgen. Om het object te relateren in de opeenvolgende frames is informatie nodig over het object zoals kleur, textuur, histogram en/of statistische gegevens. Door gebruik te maken van een model van similariteit wordt de correspondentie bepaald in het volgend frame. In het geval van bewegende objecten wordt er gesproken over spatio-temporale correspondenties.

De meeste trackingalgoritmes veronderstellen, volgens [55], dat de beweging van een object niet abrupt verloopt. Volgens [56] is één van de grootste uitdagingen bij objecttracking om ruizige objectdetecties in een nieuw videoframe te relateren aan de hand van de vorige frames en vorig getrackte objecten. Zo wordt er een “levensduur” van een object gedefinieerd die verstrijkt als het object zich niet meer in de scène bevindt. Veelal is het tracken van een object gebaseerd op het analyseren van de beweging aan de hand van de vorige frames, wat een vertraging met zich meebrengt. Voor realtime toepassingen is dit echter niet gewenst. De grootste moeilijkheid hierin is dat ruizige objectdetecties kunnen geassocieerd worden in het huidige videoframe met de vorig getrackte objecten en dit zonder enige vertraging.

6.2.2 Feature selectie voor tracking

Voor objecttracking is het belangrijk dat objecten uniek onderscheiden worden in de zogenaamde *feature space* of letterlijk vertaald: kenmerkenruimte. Om dit te verwezenlijken worden verschillende features beschouwd of een combinatie ervan die nauw verbonden zijn met een objectvoorstelling. Enkele hiervan zijn: kleur, randen, *optical flow*, textuur.

6.2.2.1 Kleur

Dit is het meest gebruikte kenmerk en wordt volgens [55] beïnvloed door twee fysische factoren: de spectrale vermogensverdeling en reflectiekenmerken van het object. Het nadeel van dit kenmerk is dat deze niet resistent is tegen veranderingen in belichting van hetzelfde object.

6.2.2.2 Randen

Objectranden of edges komen voor als sterke veranderingen in intensiteiten in de afbeelding. Een randdetectiealgoritme kan bijgevolg gebruikt worden om deze veranderingen te identificeren, zoals de *Canny Edge detector* [57].

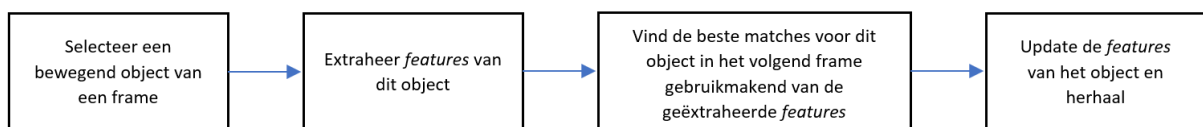
6.2.2.3 Optical flow

Bij *optical flow* wordt de translatie van elke pixel in een regio gedefinieerd aan de hand van vectoren. Het principe veronderstelt een constante helderheid van de corresponderende pixels in opeenvolgende frames [58]. Voor bewegingsdetectie spreekt men van bewegingsvectorenschatting, waarbij coherente bewegingsvectoren leiden tot de detectie van bewegende objecten. Het nadeel van optical flow is dan ook dat veranderingen in belichting het trackingresultaat negatief beïnvloeden [55].

6.2.2.4 Textuur

Bij textuur wordt de variatie in intensiteit van een oppervlak gemeten om zo kenmerken zoals “gladheid” en regelmatigheid te extraheren. Het voordeel is dat textuurkenmerken minder gevoelig zijn voor veranderingen in belichting in vergelijking met de kenmerken kleur en optical flow.

Figuur 35 toont het algemeen principe van objecttracking aan de hand van features.



Figuur 35: Algemeen concept van objecttracking (eigen bewerking van [54])

Veelal worden de kenmerken manueel gekozen door de gebruiker afhankelijk van de applicatie. In recente technieken wordt er gewerkt met leermethoden waarbij dit automatisch wordt bewerkstelligd. Om een object in een scène te volgen, zijn er verschillende methoden elk met hun voor- en nadelen, wat in het volgende onderdeel aan bod komt.

6.2.3 Methoden en technieken om een bewegend object te volgen

Door het wijde spectrum van trackingapplicaties in de computervisie hebben veel onderzoekers de laatste jaren verschillende methoden gepresenteerd die in dit onderdeel besproken zullen worden. Het probleem van objecttracking op basis van beweging kan volgens [59] in twee delen worden opgedeeld. Enerzijds is er de detectie van bewegende objecten in elk beeld. Anderzijds is er het associëren in de tijd van het corresponderend gedetecteerd object in elk frame.

6.2.3.1 Detectie van bewegende objecten

In het geval van een niet-statische camera is het relatief moeilijk om een video te verwerken, omreden dat de camerabeweging en de objectbeweging gemixt zijn. Beelden vastgelegd door een bewegende camera hebben een breed scala aan toepassingen binnen de robotica, autosector, enzovoort.

In [60] wordt een methode voorgesteld om objectdetectie en tracking te realiseren van een video genomen met een bewegende camera. Volgens de methodiek worden er eerst featurepunten gezocht in de frames. Deze punten worden vervolgens geclassificeerd als enerzijds voorgrondkenmerken of anderzijds achtergrondkenmerken. Door een integratie worden bewegende objectgebieden verkregen op basis van voorgrondkenmerken. Er wordt vervolgens een compensatieschema op basis van de beeldgeschiedenis van de continue bewegingscontouren verkregen uit drie opeenvolgende frames toegepast, om bijgevolg de gebieden met bewegende objecten te vergroten.

Bewegende objecten kunnen gedetecteerd worden op basis van video-objectsegmentatie waarbij het bewegend voorgrondobject uit elk frame geëxtraheerd wordt uit een videosequentie. In de literatuur zijn er veel methoden voorgesteld voor het segmenteren van objecten. Over het algemeen kunnen deze methoden grofweg geclassificeerd worden in twee types. Enerzijds de op achtergrondconstructiegebaseerde en anderzijds de op voorgrondextractiegebaseerde objectsegmentatie. In het eerste geval wordt de achtergrondinformatie eerst geconstrueerd en wordt een object in opeenvolgende frames verkregen door het toepassen van *background subtraction*. In het andere geval wordt de ruimtelijke informatie gebruikt om een object te verkrijgen en het object in opeenvolgende frames met behulp van bewegingsinformatie, veranderingsinformatie of een ander kenmerk te verkrijgen.

Background subtraction is het verschil tussen de achtergrond en het huidige frame. Kortom wordt er een scèneweergave opgebouwd, het *background model* genaamd, dat vergeleken wordt met inkomende videoframes om objecten te detecteren. Het voordeel volgens [59] is dat deze methode computationeel efficiënt is met het omgaan met ruis en veranderingen in de verlichting in de achtergronden. De meest recente modellen voor detectie maken hiervan gebruik. Het nadeel van deze methode is dat dit algoritme een statische camera vereist. Verbeteringen hiervoor betreffen een implementatie op basis van Gaussiaanse *mixture* modellen (GMM) [61] en adaptieve backgroundsubtraction [62]. Tot slot worden aan de hand van blobanalyse groepen verbonden pixels gedetecteerd, die waarschijnlijk overeenkomen met bewegende objecten [63].

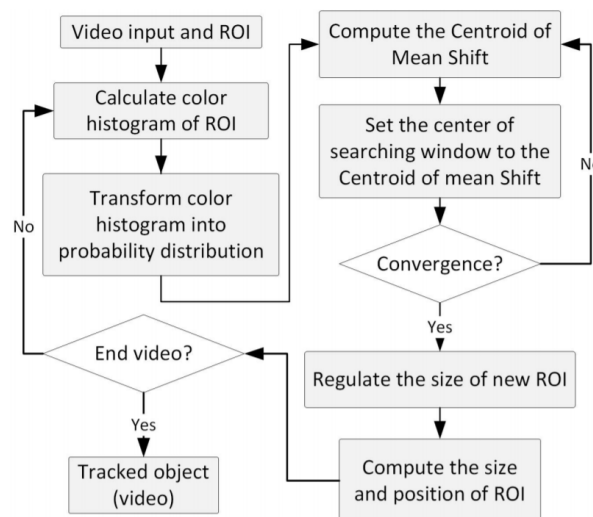
6.2.4 Associatie

De associatie van detecties met hetzelfde object is uitsluitend gebaseerd op beweging. De beweging van elke track kan geschat worden door bijvoorbeeld een Kalmanfilter. De filter wordt gebruikt om de locatie van het spoor van een object, gekenmerkt door bijvoorbeeld het middelpunt van zijn bounding box, in elk frame te voorspellen en de waarschijnlijkheid te bepalen dat elke detectie aan elk spoor wordt toegewezen. Er zijn verschillende mogelijkheden om de bewegingsdetecties met elkaar te associëren. Over het algemeen kan er een onderscheid gemaakt worden tussen filter- en leermethoden. Het eerstgenoemde probeert de kenmerken te selecteren op basis van algemene criteria. Bij het laatstgenoemde wordt er offline (op voorhand) of online (*on the fly*) “geleerd” van kenmerken van een object om deze dan te volgen.

6.2.4.1 Filtermethoden

- **CAMshift**

Continuously Adaptive Meanshift, afgekort CAMshift, is een verbetering van de *mean-shift* methode [64]. Deze methode kan gebruikt worden voor de algemene tracking van bewegende objecten en wordt uiteengezet in [65]. Het nadeel van deze methode is dat deze geen objecten kan detecteren die gehinderd worden door andere objecten binnen een frame, gedefinieerd als objectocclusie. Het principe van de CAMshift methode is weergegeven in Figuur 36.

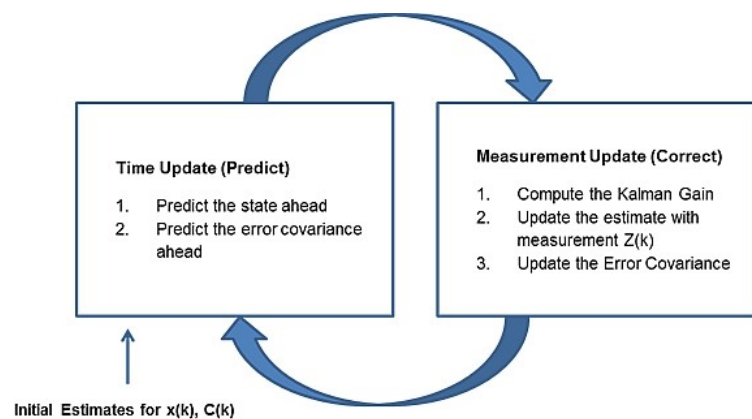


Figuur 36: Principe van de CAMshift methode [66]

- **Kalmanfilter**

Een Kalmanfilter wordt gebruikt voor het voorspellen van een lineair systeem en een Gaussiaans gedistribueerd systeem. Het is een statistische methode gebaseerd op puntracking en geeft daardoor een oplossing in de vorm van een voorspelling [67].

Eerst voorspelt dit filter de meest waarschijnlijke objectlocatie in het huidig frame gebaseerd op objecttracking van het vorig frame. De methode gaat vervolgens zoeken naar een doellocatie rond de vorige detectie. De primaire stappen van een Kalmanfilter bestaan uit voorspelling en correctie zoals weergegeven in Figuur 37.



Figuur 37: Kalmanfilter - discrete cyclus [67]

Het nadeel van deze aanpak is dat deze alleen gebruikt kan worden in lineaire of stabiele systemen met een Gaussiaans verdeelde ruis. Het voordeel van deze filter is dat deze objecten kan detecteren die tijdelijk gehinderd worden door een ander object binnen een frame, in tegenstelling tot de CAMshift methode.

- **Hybride CAMshift methode**

In [68] wordt een techniek voorgesteld om tegemoet te komen aan de problemen met betrekking tot lage videokwaliteit, systeemruis en/of een te klein object. Bij deze hybride aanpak wordt de CAMshift methode gebruikt als hoofdtracking en de Kalmanfilter als verfijning.

Andere mogelijkheden zijn mean-shift [64], *particle filtering* [69], ...

6.2.4.2 Leermethoden

De laatste jaren is er meer ingezet op het “leren”. Afhankelijk van de leermethode wordt er een onderscheid gemaakt tussen offline- en onlinemethoden met elk hun voor- en nadelen.

- **Offline leren**

Bij de offlineleermethoden wordt er geleerd voordat de eigenlijke tracking plaatsvindt. Er wordt daarbij gebruikgemaakt van een grondwaarheid zodoende een gelijkaardigheidsfunctie te leren tussen objectdetectie en objecttracking voor data-associatie. Het nadeel hiervan is dat het een statische benadering betreft en bijgevolg de dynamische status en het verleden van het object niet in rekening gebracht kan worden. Dit is belangrijk om dubbelzinnigheden op te lossen zoals gemiste objecten of objecten die zich uit het beeld begeben. *Generic Object Tracking Using Regression Networks* (GOTURN) [70] is een voorbeeld van een objecttracking algoritme dat gebaseerd is op *offline learning*.

- **Online leren**

In de onlineleermethoden wordt er geleerd tijdens het tracken. Een goede aanpak is om zowel positieve als negatieve trainingsvoorbeelden te construeren volgens de trackingsresultaten en dan een soortgelijke functie te trainen voor data-associatie. Het voordeel hiervan is dat deze manier van werken het mogelijk maakt om features te gebruiken die gebaseerd zijn op de huidige status en het verleden van het object. Het nadeel is echter dat er geen grondwaarheid beschikbaar is voor supervisie. De methode kan op die manier leren van mogelijk incorrecte trainingsdata. Als gevolg hiervan kunnen deze fouten zich accumuleren en vervolgens resulteren in een *tracking drift* [56]. *Multiple Instance Learning* (MIL) [71] en *Discriminative Scale Space Tracking* (DSST) [72] zijn voorbeelden van objecttrackers die hierop zijn gebaseerd .

Over het algemeen zijn er drie stappen om een bewegend object te volgen [55].

1. De detectie of het selecteren van de ROI van een bewegend object in een video of scène. Een objectdetector, vermeld in Hoofdstuk 5, kan hiervoor dienstdoen.
2. Het geselecteerd object volgen in de opeenvolgende frames.
3. De analyse van de beweging van het object om zodoende het gedrag van het object te analyseren en identificeren.

De eerste twee stappen vormen de basis van het draftingdetectiesysteem en worden in het volgend deel beschreven.

6.3 Objecttracker

Om een verantwoorde keuze te maken met betrekking tot de objecttracker, worden er acht objecttrackers uit de literatuur gebenchmarkt die eveneens geïmplementeerd zijn in

OpenCV¹ en/of DLIB². De verschillende trackers betreffen:

- *Channel and Spatial Reliability correlation filter Tracker* (CSRT) [75];
- *Kernelized Correlation Filter* (KCF) [76];
- *Boosting*: een realtime objecttracker gebaseerd op het online AdaBoost algoritme [77];
- *Multiple Instance Learning* (MIL): een trackingalgoritme dat een classifier online traint om het object te onderscheiden van de achtergrond [71];
- *Tracking, Learning and Detection* (TLD): een tracker gericht op langdurig tracken [78];
- *Medianflow*: een trackingalgoritme gebaseerd op optical flow [79];
- *Minimum Output Sum of Squared Error* (MOSSE): deze tracker werkt met grijswaarde beelden en maakt gebruik van adaptieve correlatiefilters [80];
- *Discriminative Scale Space Tracker* (DSST): een online learning algoritme met een adaptief correlatiefilter [72].

Elk van deze objecttrackers worden getest op detectiesnelheid, *successrate* en precision.

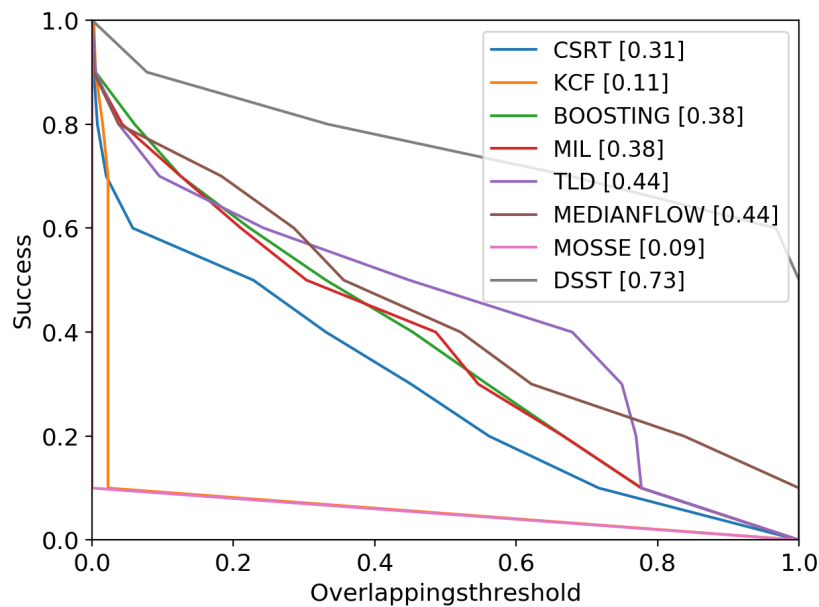
6.3.1 *Successplots*

Een eerste vergelijking is op basis van de *successplots* van de desbetreffende objecttrackers. Zo'n grafiek geeft het percentage weer van de frames (op de y-as) waarvan de IOU van de voorspelde en grondwaarheid bounding box groter is dan de overlappingsthreshold op de x-as [81]. Het resultaat na benchmarken op de testset is weergegeven in Figuur 38 waarbij in de legende de gemiddelde successwaarde over alle thresholds is beschouwd.

Het doel is om een zo hoog mogelijke *successrate* te bekomen en een zo groot mogelijke overlappingsthreshold. In de legende is de *Area Under the Curve* (AUC) van elke tracker te vinden. Uit deze AUC-waarde en de bijhorende curves kan afgeleid worden dat DSST, TLD en Medianflow de meest aangewezen keuzes zijn met respectievelijke AUC-scores: 0,73; 0,44 en 0,44.

¹*Open Source Computer Vision* (OpenCV) is een bibliotheek voor Python en C++ ontwikkeld door Intel en functies aanbiedt voor computervisie en beeldverwerkingstoepassingen [73]

²DLIB is een *Machine Learning* bibliotheek geschreven in C++ met ondersteuning voor Python met verschillende algoritmen voor beeldverwerking, datastructuren, lineaire algebra, ... [74]



Figuur 38: Successplots voor verschillende objecttrackers met in de legende de AUC-score voor elke overeenkomstige objecttracker

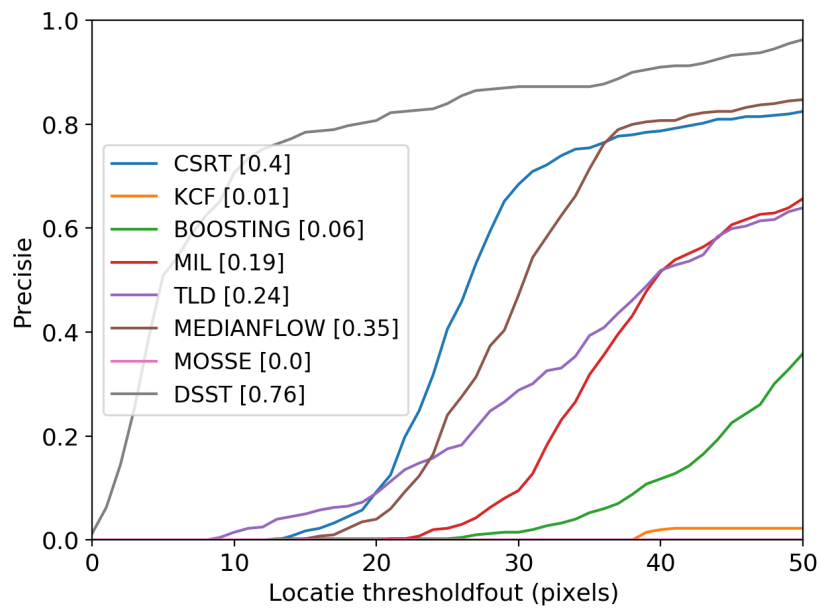
6.3.2 Precisieplots

Aangezien zoals eerder vermeld ook de precisie van de getrackte bounding boxes van belang is voor het bepalen van de afstand, zijn *precisionplots* opgesteld van elk van de acht objecttrackers. In deze context wordt volgens [81] een precisiegrafiek gedefinieerd als de gemiddelde euclidische pixelafstand tussen het centrum van de bounding box van het getrackte object en de grondwaarheid over alle frames in de reeks. Het toont aldus het percentage van de frames waarbij het centrum van de geschatte locatie zich binnen een bepaalde afstandsthresholdfoutmarge (*location error threshold* op de x-as) ligt van het centrum van de grondwaarheid bounding box. Deze benchmark is uitgevoerd op de testset waarvan de resultaten in Figuur 39 zijn weergegeven. Analoog is de gemiddelde preciescore weergegeven in de legende van de beschouwde figuur. Het doel is om een zo hoog mogelijke precisie te bekomen gecombineerd met een zo laag mogelijke location error threshold. Uit deze precisiebenchmark blijkt dat CSRT en DSST met een preciescore van respectievelijk 0,4 en 0,76 relatief goede resultaten opleveren.

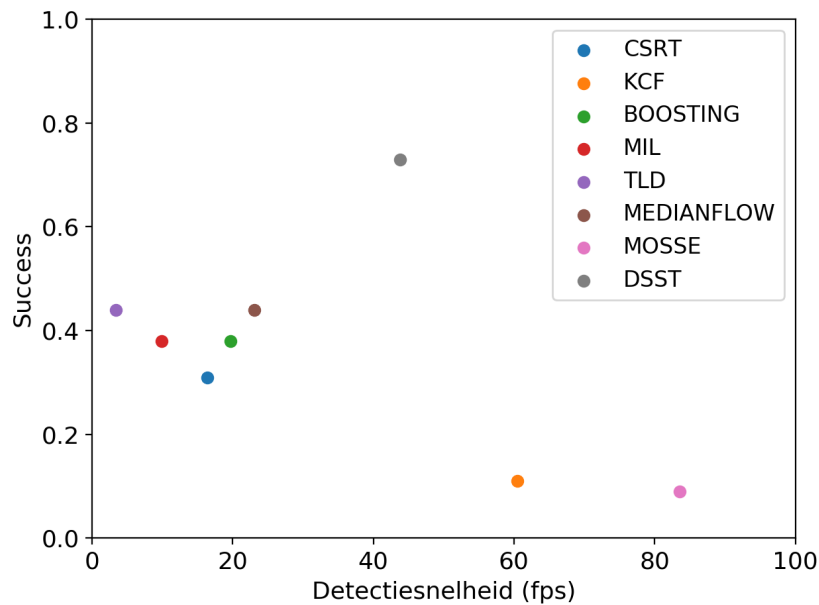
6.3.3 Successrate versus detectiesnelheid

Een laatste benchmark zet de successrate uit ten opzichte van de detectiesnelheid³. De resultaten van deze testen zijn weergegeven in Figuur 40.

³Deze benchmark is getest op een CPU van het type Intel i7 4710HQ



Figuur 39: Precisionplots voor verschillende objecttrackers waarbij de gemiddelde preciescore van elke objecttracker is weergegeven in de legende



Figuur 40: Successrate versus detectiesnelheid voor verschillende objecttrackers

Uit de drie benchmarks blijkt dat de DSST-objecttracker de meest aangewezen oplossing is. Deze is gebaseerd op MOSSE maar is robuuster in gevallen van schaalverandering. De DSST-tracker maakt namelijk gebruik van een multischaalpiramide om de schaal van een object te schatten. Dit maakt het mogelijk om objecten te tracken die veranderen in translatie (een fietser die bijvoorbeeld naar links opschuift in het beeld) en schaling (een



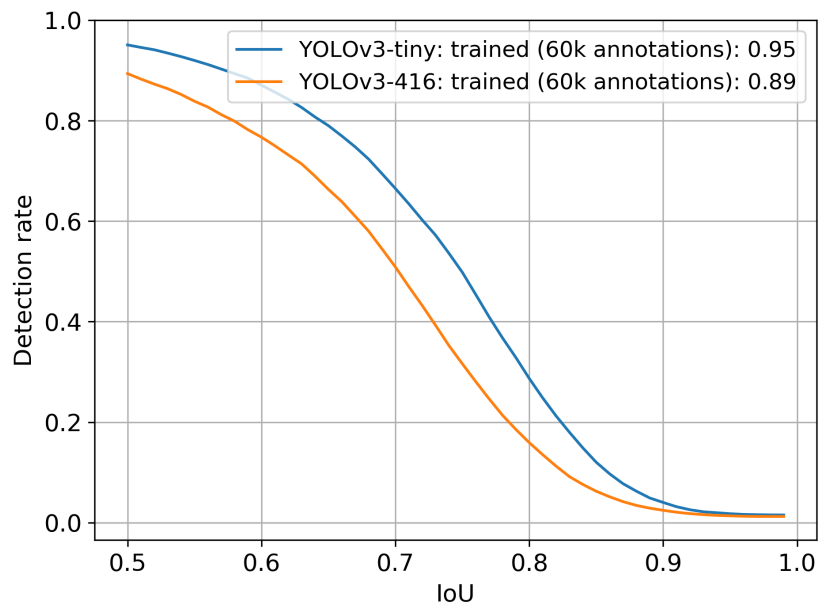
Figuur 41: Het ontstaan van trackingsdrift en lage precisie voor verschillende objecttrackers

fietser die dichterbij komt) [72].

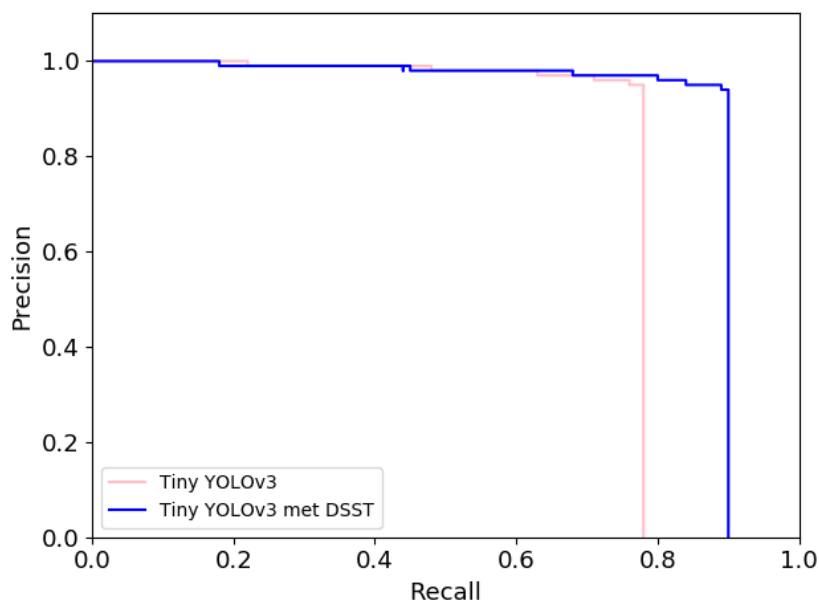
Naast het kunnen volgen van fietsen wordt de objecttracker ook gebruikt om de detection rate te verhogen. Zowel het resultaat van het getrainde YOLOv3-netwerk als deze van Tiny YOLOv3 worden beide getest in combinatie met de DSST-objecttracker met betrekking tot hun verbetering in de detectierate. Figuur 42 geeft de detectierate weer voor beide neurale netwerken en verschillende IOU-thresholds. Daar er initieel zonder objecttracker een gemiddelde detectierate ($1 - LAMR$) van 74 % is gekomen, is deze door het gebruik van de objecttracker gestegen tot 95,08 % in het geval van Tiny YOLOv3, in combinatie met een hogere detectiesnelheid. Voor de YOLOv3 bedraagt dit respectievelijk 85 % en 89,37 %. Nu er een hogere detectierate is gekomen door het gebruik van de objecttracker moet ook de hoogte en de hoogteligging van de bounding box worden geëvalueerd. Dit wordt in het volgend onderdeel behandeld.

6.3.4 Precision voor fietsdetectie

Aangezien de precisie van de detecteerde en/of getrackte bounding box van de fiets van belang is voor de afstandsbepaling, wordt de hoogte en de hoogteligging van de bounding box geëvalueerd voor de combinatie van het getrainde Tiny YOLOv3 en DSST-objecttracker. Daar initieel een AP voor Tiny YOLOv3 is gekomen van 77,19 %, wordt door het verhogen van de recall deze gemiddelde precisie verhoogd tot 88,27 %. De precision recall curve voor deze combinatie is weergegeven in Figuur 43 samen met het geval zonder de objecttracker.



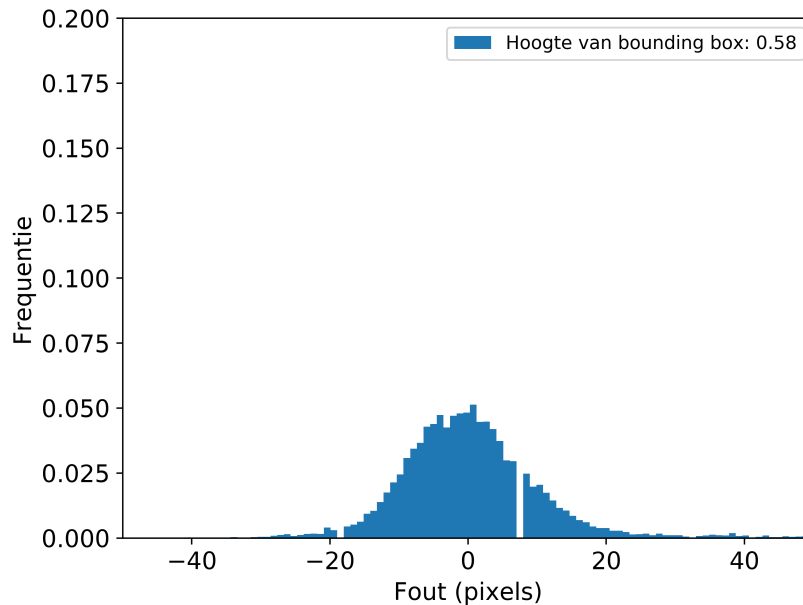
Figuur 42: Detection rate voor verschillende IOU's voor het getrainde (Tiny) YOLOv3-netwerk en de DSST-objecttracker, waarbij de legende de detection rate weergeeft voor een IOU van 0.5



Figuur 43: Precision recall curve voor Tiny YOLOv3 met de DSST-objecttracker

Voor de afstandsbeoordeling in Hoofdstuk 7 kan de afstand op twee manieren worden bepaald: de hoogte van de bounding box of de ligging van de bounding box ten opzichte van het grondvlak. Om een eerste indicatie te geven welke methode de grootste afwijking heeft, wordt in Figuur 44 de pixelfout op de hoogte van de bounding box en in Figuur

45 de hoogteligging ervan weergegeven. Een negatieve fout impliceert dat de waarde te groot is in vergelijking met de grondwaarheid en omgekeerd voor een positieve fout.

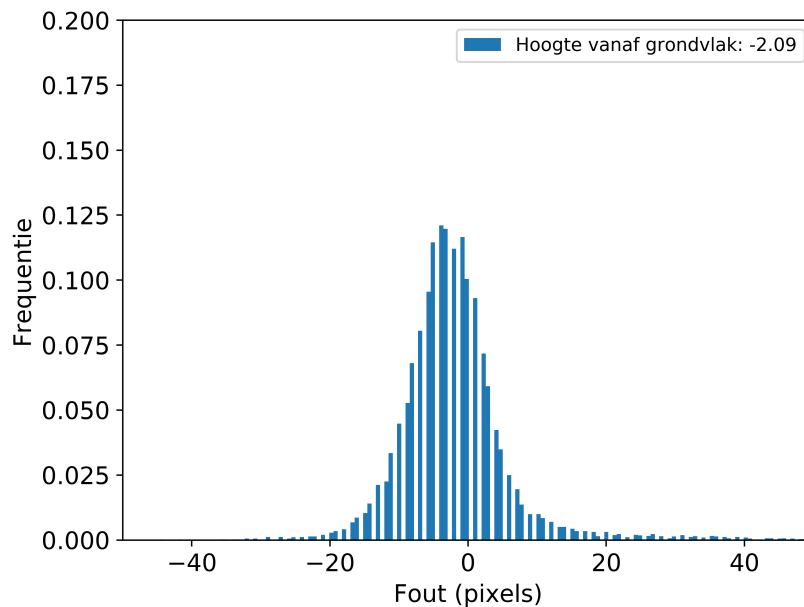


Figuur 44: Pixelfout op de hoogte van de bounding box, met in de legende de gemiddelde fout

De gemiddelde fout voor beide gevallen bedraagt respectievelijk 0,58 en -2,09 pixels, telkens weergegeven in de legende. In het eerste opzicht blijkt dat de fout op de hoogte van de bounding box lager is in vergelijking met de andere aanpak. De -2,09 pixels geeft reeds een indicatie dat de tilthoek van de camera niet correct is ingesteld en/of bepaald. De camera is naar onder getilt maar de tilthoek is te laag, wat de negatieve waarde verklaart. Dit wordt verder verduidelijkt in Hoofdstuk 7. De oorzaak van deze fout op de bounding boxes is het feit dat neurale netwerken erg onstabiel zijn voor kleine veranderingen in het beeld. Zo kan een verandering van één pixel volgens [82] een groot effect hebben op de uitkomst van het netwerk.

6.4 Trackbeheer

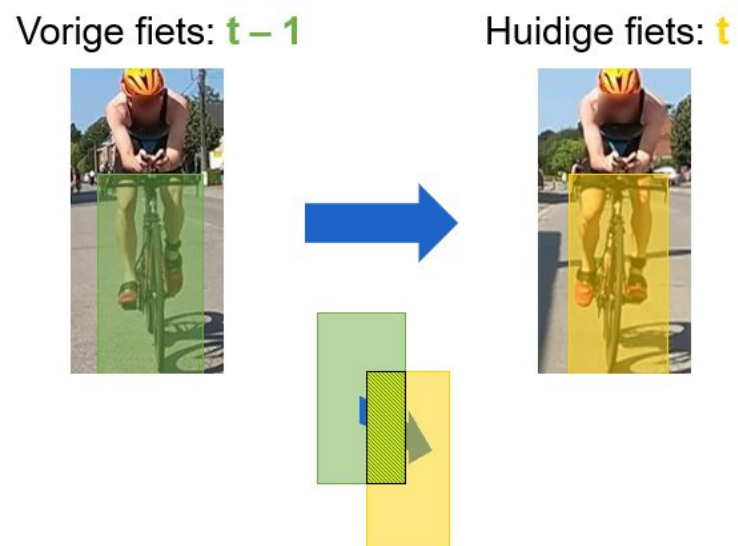
Om de gedetecteerde fietsers doorheen de verschillende scènes te volgen, wordt er trackmanagement verricht. Op die manier kan éézelfde fiets uniek geïdentificeerd worden door zijn ‘bicycle id’. Dit is vereist voor deze applicatie aangezien de tijd moet gemeten worden hoelang een bepaalde renner zich in de draftingzone bevindt. Meer bepaald mag deze zich bij het voorbijsteken maximaal 20 seconden in deze zone



Figuur 45: Pixelfout op de hoogteligging van de bounding box ten opzichte van het grondvlak, met in de legende de gemiddelde fout

begeven, zoals de rechterpersoon in Figuur 16. Als een renner voorbijgestoken wordt, zoals de linkerpersoon in Figuur 16, moet deze persoon zich volgens het reglement binnen de 5 seconden verwijderen uit de zone.

Als de bounding box van een gedetecteerde fiets in frame $(t - 1)$ overlapt met de bounding box van een fiets in frame (t) , dan wordt ervan uitgegaan dat dit dezelfde atleet betreft. De voorwaarde voor deze data-associatie is dat de IOU groter is dan een bepaalde threshold. Figuur 46 geeft het principe van dit correspondentieprobleem weer. De reden om een renner alleen tussen twee frames met elkaar te associëren en niet over een bepaald traject na x -aantal frames, is omwille van de bewegingen van de camera. Er is verondersteld dat de camera beweegt (al dan niet willekeurig) in een statische en niet-gecontroleerde omgeving. In de praktijk is deze omgeving echter niet statisch als gevolg van de beweging van wolken, beweging van bomen door de wind, de verandering in belichting, ... Het geselecteerde object, meer bepaald de bounding box ervan, wordt gevolgd in de opeenvolgende frames. Het voordeel hiervan is dat deze aanpak een relatief eenvoudige berekening met zich meebrengt. Het nadeel is daarentegen dat er wordt verondersteld dat er geen abrupte bewegingen optreden tussen twee frames, wat in de praktijk na visuele controle niet het geval is bij een framerate van ongeveer 15 tot 30 fps. Een eventuele oplossing voor dit mogelijke probleem is het gebruik van een schaalinvariante matcher, zoals SIFT [83], die het mogelijk maakt om de gedetecteerde bounding boxes aan elkaar te associëren. Het nadeel hiervan is dat dit voor een overhead zorgt met betrekking tot het aantal berekeningen en de verwerkingstijd.



Figuur 46: Correspondentieprobleem voor het associëren van fietsen

Hoofdstuk 7

Afstandsbepaling

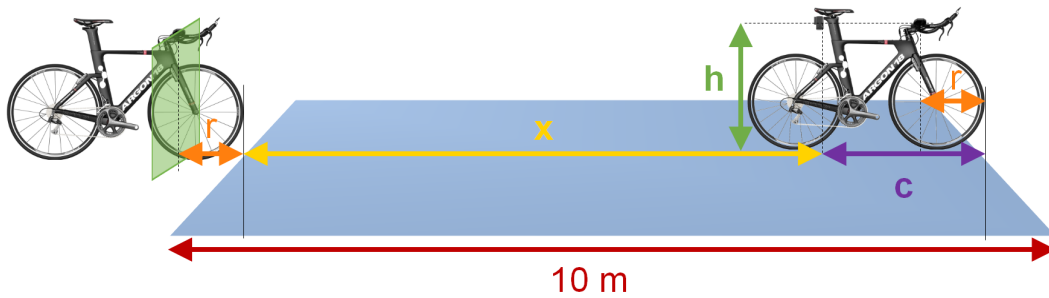
In Hoofdstuk 5 en 6 is besproken hoe fietsen gedetecteerd en gevolgd worden doorheen de beeldsequenties. Om daaropvolgend te bepalen of een atleet effectief aan het draften is en zich bijgevolg in de draftingzone bevindt, moet de afstand tot de gedetecteerde fiets worden bepaald. Het gedeelte van de afstandsbepaler komt aan bod in dit hoofdstuk. Aansluitend wordt in het volgende hoofdstuk het systeem onderworpen aan een reeks testen en benchmarks.

7.1 Berekenen van de afstand

Om de afstand tot de achterliggende triatleet te bepalen, wordt beroep gedaan op de bounding box van de overeenkomstig gedetecteerde fiets. Zoals eerder gesteld is het om die reden van belang dat de precisie, met de name de hoogte enerzijds en de hoogteligging anderzijds, van de bounding box relatief hoog moet zijn. Indien dit niet het geval is, zorgt dit voor een afwijking in de afstandsbepaling en mogelijk in het verkeerd toekennen van een draftingpenalty. Om een afstand te bepalen is eerst de focale lengte f nodig van de camera, in dit geval een GoPro Hero 6 met een verticale *Field Of View* (FOV) van ongeveer 70 graden. Deze bedraagt, volgens de technieken besproken in Hoofdstuk 4, ongeveer 995 pixels.

Er worden twee methoden voorgesteld om de afstand tot een bounding box aan de hand van een monoculaire cameraopstelling te bepalen, wat in de volgende twee subparagrafen wordt besproken. Beide methoden maken gebruik van relatief eenvoudige meetkundige technieken. Aangezien de camera achteraan het zadel is gemonteerd en volgens het reglement de draftingafstand berekend wordt van voorwiel tot voorwiel, moeten er een aantal afmetingen worden opgenomen van de fiets met de camera zoals aangegeven in Figuur 47. De afstand c in het paars op de beschouwde figuur komt overeen met de afstand van het voorwiel tot de plaats waar de camera hangt. De parameter x , in het geel uit de betreffende figuur, correspondeert met de afstand die de camera meet. Aangezien de objectdetector een fiets detecteert vanaf het stuur tot het contactpunt van het voorwiel en het grondvlak, is er een correctie nodig. Deze afstand hangt af van het fietstype zoals de wielstraal¹, de lengte van de stuurpen en het stuurtype. Als

¹Volgens het triatlonreglement [1] kan de wielstraal in theorie variëren van 27,5 cm tot 35 cm wat bijgevolg overeenkomt met een maximale afwijking van 7,5 cm.

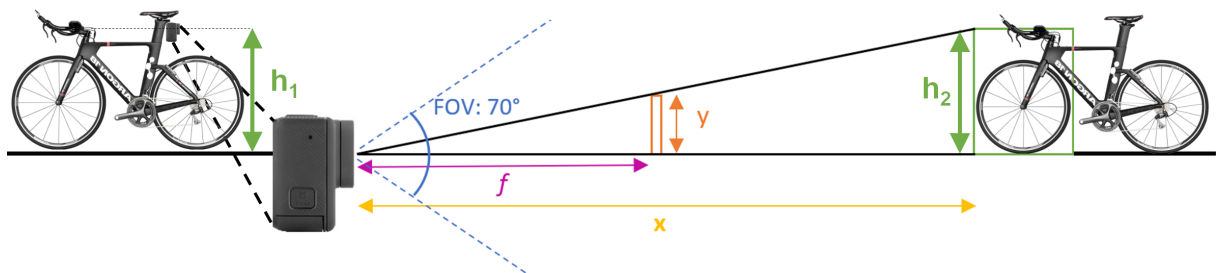


Figuur 47: Cameraplatsingscorrectie (paars) met stuurcorrectie (oranje) (eigen bewerking van [84])

benadering hiervoor wordt de afstand r beschouwd, gemeten vanaf het stuur tot het voorwiel zoals aangegeven in het oranje. Aangezien deze varieert van fiets tot fiets wordt als benadering de afstand r van de camerafiets (rechts in Figuur 47) gebruikt die verondersteld wordt gelijk te zijn aan deze van de achterligger (links in Figuur 47). Naast het afmeten van de afstand c en r , is voor beide afstandsmethoden in het volgend onderdeel ook de werkelijke hoogte van de camera vereist, met name h in Figuur 47.

7.1.1 Hoogtemethode

In deze eerste methode wordt verondersteld dat de hoogte van de camera op de camerafiets gelijk is aan de hoogte van het stuur van de achterliggende fiets. Deze hoogtes, respectievelijk h_1 en h_2 , zijn weergegeven in Figuur 48.



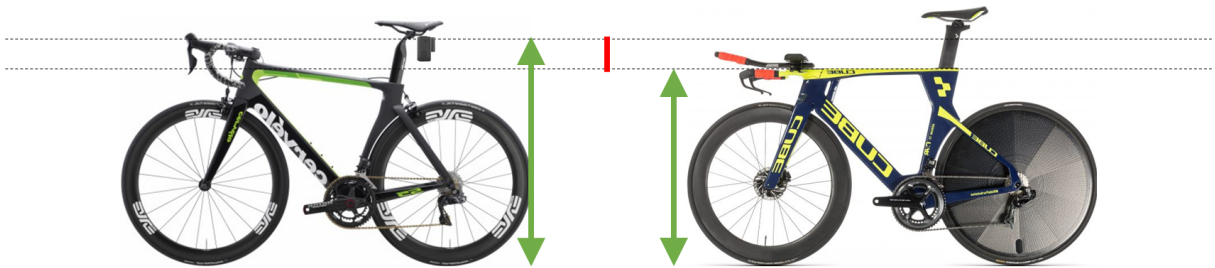
Figuur 48: Principe van de hoogtemethode voor afstandsbepaling (eigen bewerking van [84])

Op basis van de gedetecteerde bounding box wordt de hoogte, uitgedrukt in pixels, gerelateerd aan een afstand volgens het principe aangegeven in Figuur 48. In deze voorstelling komen f en y respectievelijk overeen met de focale lengte van de camera en de hoogte van de bounding box op de beeldsensor (beiden uitgedrukt in pixels). De afstand x in de figuur komt overeen met de afstand vanaf de camera tot aan het

voorwiel, die bijgevolg bepaald moet worden. De driehoek gevormd door de zijden f en y is gelijkzijdig met de driehoek gevormd door de zijden h_2 en x . Op die manier kan een relatie opgesteld worden om de afstand te bepalen zoals aangegeven in formule (9).

$$\begin{aligned} \frac{f}{x} &= \frac{y}{h_2} \\ x &= f \cdot \frac{h_2}{y} \end{aligned} \quad (9)$$

Het nadeel van deze methode is dat er wel degelijk een verschil tussen de camerahoogte en stuurhoogte kan zijn. Dit hoogteverschil is afhankelijk van het soort fiets en de wensen van de atleet voor een aerodynamische houding. Volgens [85] wordt voor een tourfiets een gelijke stuur- en zadelhoogte gekozen. In het geval van een koers- of triatlonfiets staat het stuur lager dan het zadel waarbij het verschil in de praktijk ongeveer 10 à 15 cm kan bedragen. In Figuur 49 is links een gewone racefiets weergegeven met in het rood het mogelijk hoogteverschil.



Figuur 49: Hoogtemethode verschil tussen een gewone racefiets en een typische triatlonfiets [48] en [49]

Het voordeel is dat deze methode eenvoudig is en de invloed van de tilthoek van de camera, en bijgevolg de ligging van de bounding box ten opzichte van de horizon, verwaarloosd kan worden. Een andere methode om de afstand te bepalen wordt in het volgende onderdeel behandeld en is meer afhankelijk van de tilthoek van de camera.

7.1.2 Grondvlakmethode

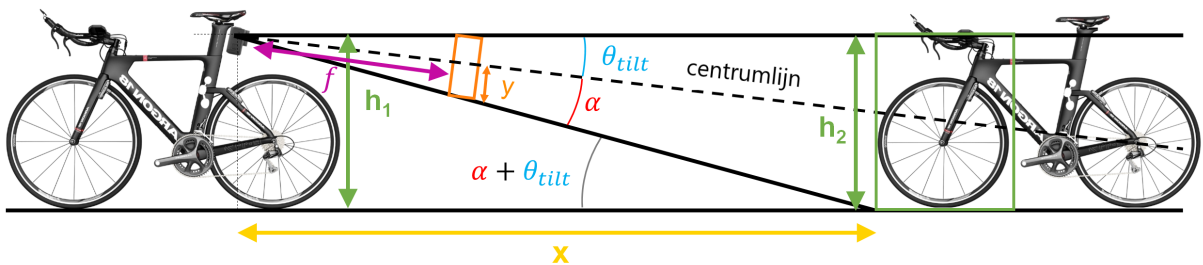
Een tweede mogelijkheid om de afstand te bepalen tot een bounding box betreft de grondvlakmethode. In het kort relateert deze aanpak de ligging van de onderkant van de bounding box aan het grondvlak vertrekkend van een gekende tilthoek en hoogte van de camera. In tegenstelling tot de hoogtemethode uit onderdeel 7.1.1 wordt bijgevolg niet de hoogte van de bounding box zelf beschouwd. Een visuele samenvatting van beide methoden is weergegeven in Figuur 50 waarbij y_1 de hoogte van de bounding box

betreft, nodig voor de hoogtemethode, en y_2 de ligging van de onderkant van de bounding box ten opzichte van het grondvlak.



Figuur 50: Principe van de hoogtemethode y_1 en de grondvlakmethode y_2

Concreet wordt de onderkant van de bounding box gekoppeld aan een hoek α waarbij ook een tilthoek θ_{tilt} in rekening gebracht wordt. De horizon verplaatst zich als het ware afhankelijk van de tilthoek. Figuur 51 geeft het principe van deze methodiek weer.



Figuur 51: Principe van de grondvlakmethode voor afstandsbeeping (eigen bewerking van [84])

Aan de hand van geometrische technieken kan de relatie opgesteld worden om tot een afstandsformule te komen. De hoek α kan bepaald worden volgens formule (10).

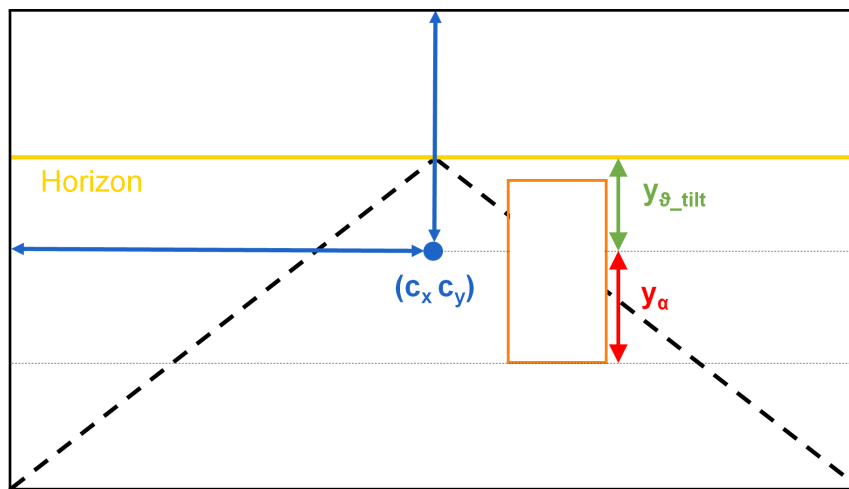
$$\alpha = \tan^{-1} \left(\frac{y}{f} \right) \tag{10}$$

Volgens het principe van verwisselende binnenhoeken is de hoek $\alpha + \theta_{tilt}$ gevormd door de driehoekszijden h_2 en x gelijk aan deze van de driehoek gevormd door de zijden h_1 en x . Op die manier kan een relatie opgesteld worden voor de afstandsbeeping x volgens

formule (11).

$$\begin{aligned} \tan(\alpha + \theta_{\text{tilt}}) &= \frac{h_1}{x} \\ x &= \frac{h_1}{\tan(\alpha + \theta_{\text{tilt}})} \end{aligned} \quad (11)$$

Als de camera perfect is gealigneerd, moet de verticale ligging van de horizon overeenkomen met de hoogte van het cameracentrum (c_x, c_y) . In dat geval moet $y_{\theta_{\text{tilt}}}$, de verandering van de horizonligging als gevolg van de tilthoek, in Figuur 52 0° zijn.



Figuur 52: Horizonverplaatsing afhankelijk van de tilthoek θ_{tilt} van de camera

Theoretisch is het voordeel van deze methode dat deze nauwkeuriger is dan de hoogtemethode, aangezien deze onafhankelijk is van de hoogte van de achterliggende fietser en er bijgevolg geen veronderstelling moet gemaakt worden over de gelijkheid van de camerahoogte en stuurhoogte. Het grote nadeel van deze methode is dat deze sterk afhankelijk is van θ_{tilt} , de tilthoek van de camera. Zo moet de initiële tilthoek gekend zijn voor elk systeem. Er zijn bovendien situaties waarbij het systeem door een productiefout niet perfect gealigneerd is. Een andere mogelijkheid is dat de camera bij de huidige opnamemogelijkheid kan verdraaien door de schokken die deze ondervindt. Een gelijkaardig gedrag wordt bemerkt als er een verschil in helling is door bijvoorbeeld een opwaartse helling of neerwaartse helling van een brug zoals weergegeven in Figuur 53.

Om de invloed van de tilthoek op de afwijking die deze introduceert op de afstandsberekening in acht te nemen, worden in het volgende hoofdstuk benchmarks uitgevoerd. Er is bovendien nood aan een automatische camerakalibratie om een tilthoekverandering op te vangen, wat in het volgende onderdeel aan bod komt.



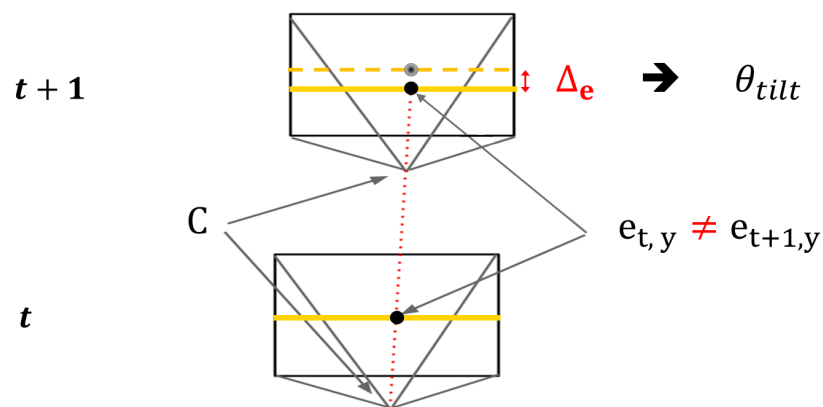
(a) Oplopende helling

(b) Aflopende helling

Figuur 53: Helling van een brug

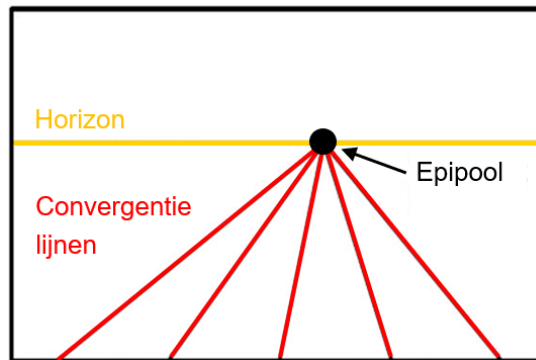
7.2 Tilhoekverandering

De invloed van de tilhoek wordt bij de hoogtemethode verwaarloosd, maar is bij de grondvlakmethode van cruciaal belang. Echter kan deze tilt afwijken door de montage van het draftingdetectiesysteem op de fiets of het verkeerd afmeten ervan. Bovendien kan er een mogelijkheid zijn dat de camera verdraaid is tijdens een wedstrijd. Om deze tilhoek te bepalen kan de ligging van de horizon worden gebruikt. In het ideale geval dat de tilhoek perfect 0° bedraagt, moet de horizon op dezelfde hoogte liggen als het centrum C van de camera: c_y . Als de camera meer naar boven is gericht, zal de horizon bijgevolg lager komen te liggen en vice versa. Dit principe is geïllustreerd in Figuur 54 waarbij de verplaatsing van de horizon Δ_e ($\Delta_e = e_{t,y} - e_{t+1,y}$) overeenkomt met een bepaalde tilhoek θ_{tilt} .

Figuur 54: Ligging van de epipool als de tilhoek initieel 0° bedraagt

Om de ligging van de horizon te bepalen, wordt er beroep gedaan op het bepalen van een verdwijnpunt (ofwel vanishing point) wat tussen twee frames in overeenkomt met de

epipool e . De notaties $e_{t,y}$ en $e_{t+1,y}$ komen bijgevolg overeen met de verticale ligging van de epipool voor respectievelijk frame (t) en $(t + 1)$. Meer bepaald zullen parallelle lijnen in een perspectieve scène convergeren in dit vluchtpunt zoals weergegeven in Figuur 55 [12].

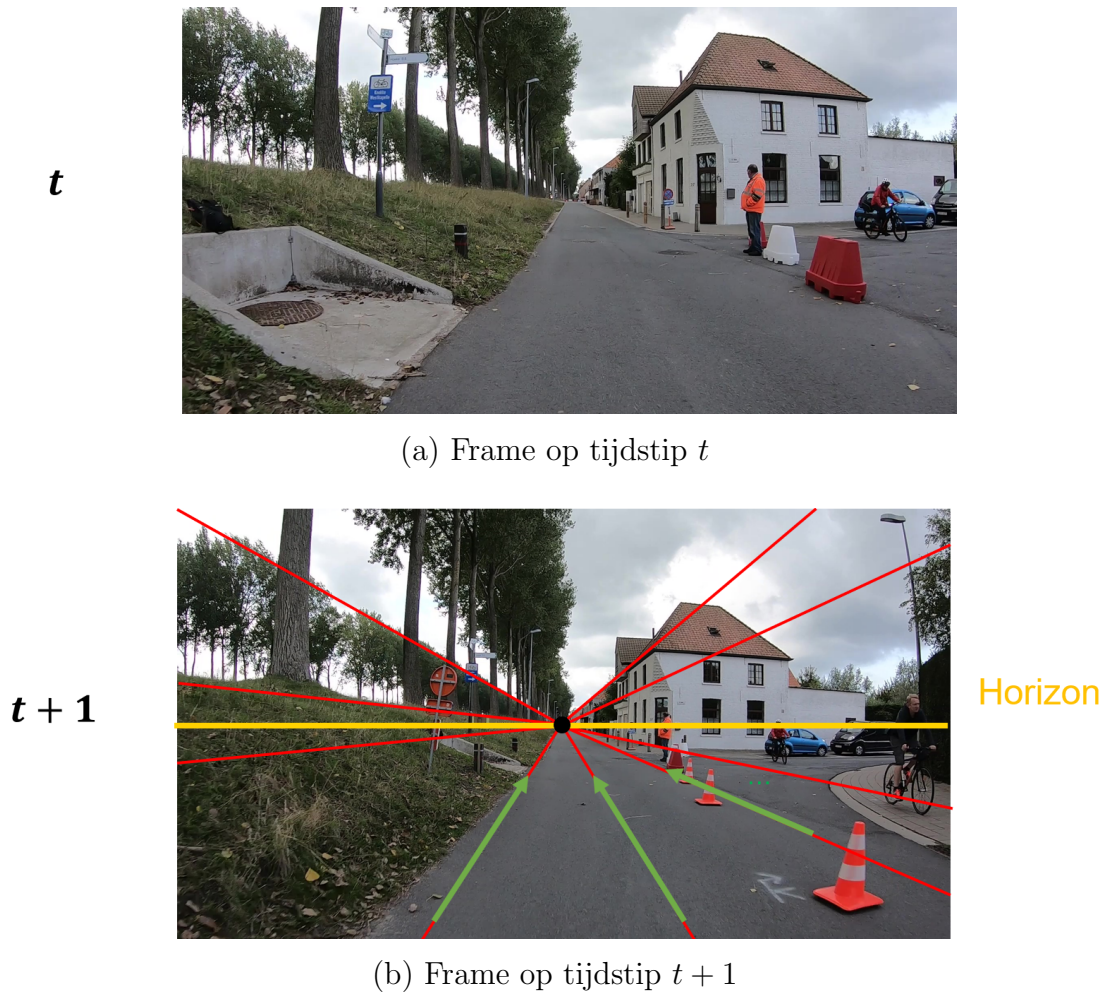


Figuur 55: Verticale epipoolligging bepalen

Een eerste stap is bijgevolg om deze convergentielijnen te bepalen. Er kan hiervoor randdetectie uitgevoerd worden om deze kandidaten te selecteren. Het nadeel is echter dat het aantal convergentielijnen voor bijvoorbeeld een stedelijke omgeving rijkelijk is in tegenstelling tot een open ruimte met weinig kenmerkpunten en bijgevolg kandidaten. Een tweede mogelijkheid is om het principe van optical flow aan te wenden om de ligging van de epipool te bepalen zoals weergegeven in Figuur 56.

Beschouw een bepaalde scène op tijdstip t en een beeld op een tijdstip $t + 1$ later zoals in Figuur 56 is weergegeven, dan zullen de optical flow vectoren zich richten volgens de convergentielijnen. Deze vectoren respectievelijk lijnen zijn aangeduid in het groen en rood in de overeenkomstige illustratie. Op die manier zullen deze uitsluitsel geven over de locatie van het vanishing point en bijgevolg de ligging van de horizon.

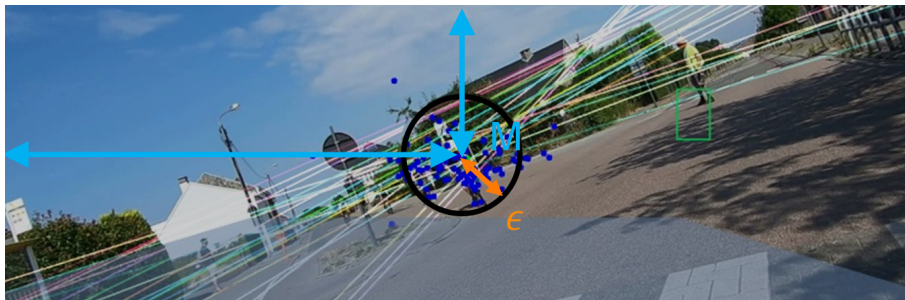
Naast het verdraaien van de camera zorgt een wisselende helling zoals een brug of een heuvel in de Vlaamse Ardennen ook voor een verplaatsing van de horizon in het beeld. De mate waarin de ligging van de horizon wordt gecontroleerd, moet bijgevolg met voorbedachte rade gekozen worden. Intuïtief geldt: hoe meer relatief goede kandidaten voor de lijnsegmenten, hoe beter het uiteindelijk vanishing point maar hoe langer de berekeningstijd. De mogelijkheid bestaat dat de cameratilt of de helling van het wegdek verandert terwijl deze kalibratiecontrole bezig is, wat afhankelijk is van de *refresh rate* van de checker. Veronderstel dat er 300 frames nodig zijn om een nauwkeurig resultaat te bekomen, wat bij een 30 fps camera overeenkomt met 10 seconden. Dit heeft als gevolg dat als er een triatleet aan het draften is, het draftingdetectiesysteem 10 seconden lang niet kan controleren. Kortom heeft de keuze van de refresh rate van de kalibratiechecker



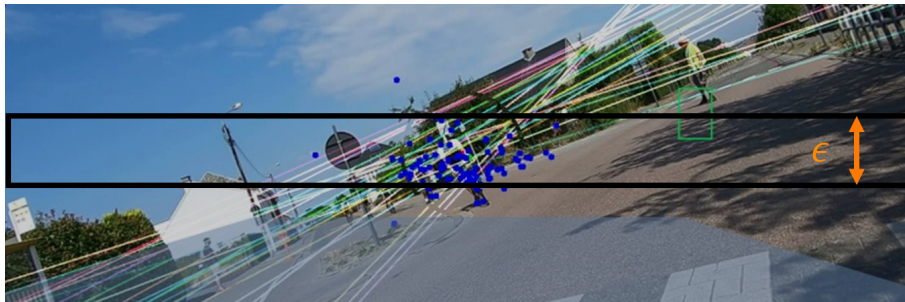
Figuur 56: Ligging van epipool aan de hand van optical flow vectoren (groen) volgens de convergentielijnen (rood)

invloed op het nauwkeurig bepalen van de horizonligging en de continue monitoring voor draftingdetectie. Deze is echter afhankelijk van de omgeving, de benodigde berekeningstijd en de rechtstreekse invloed op de tijdsmeting voor draftingdetectie. Het grote nadeel van deze methode is dat enerzijds optical flow erg gevoelig is voor belichtingsveranderingen en anderzijds dat de camera beweegt waarbij situaties voorkomen zoals in Hoofdstuk 4 is behandeld. Vooral de invloed van het heen en weer bewegen van de fiets, het nemen van een bocht en de invloed van dichtbij gelegen omstaanders blijkt in de praktijk een nefaste invloed te hebben op het bepalen van het vanishing point. Een flowchart van het algoritme dat hiervoor is gehanteerd, is weergegeven in Figuur 95 in Bijlage E. Om de kandidaatlijnsegmenten te verbeteren worden beperkingen opgelegd op de regio's in het beeld waar er mag gezocht worden naar het vanishing point. Er wordt vanuit gegaan dat het initiële vanishing point overeenkomt met de coördinaten van het cameracentrum. Er wordt met andere woorden

iteratief naar het ‘echte’ vanishing point toegewerkt. Een eerste benadering veronderstelt dat het nieuwe vanishing point zich in een cirkel $c(M, \epsilon)$ bevindt met M overeenkomend met het cameracentrum. Dit wordt de cirkelmethode genoemd. Een tweede mogelijkheid is om een horizontale band te beschouwen waarin dit te zoeken punt kan liggen met breedte ϵ , voor de smalle rechthoekmethode, of 2ϵ , voor de brede rechthoekmethode. Deze twee methoden zijn conceptueel weergegeven in Figuur 57 waarbij de optical flow vectoren zijn getekend als gekleurde lijnen naast enkele berekende vanishing points in het blauw.



(a) Cirkelmethode



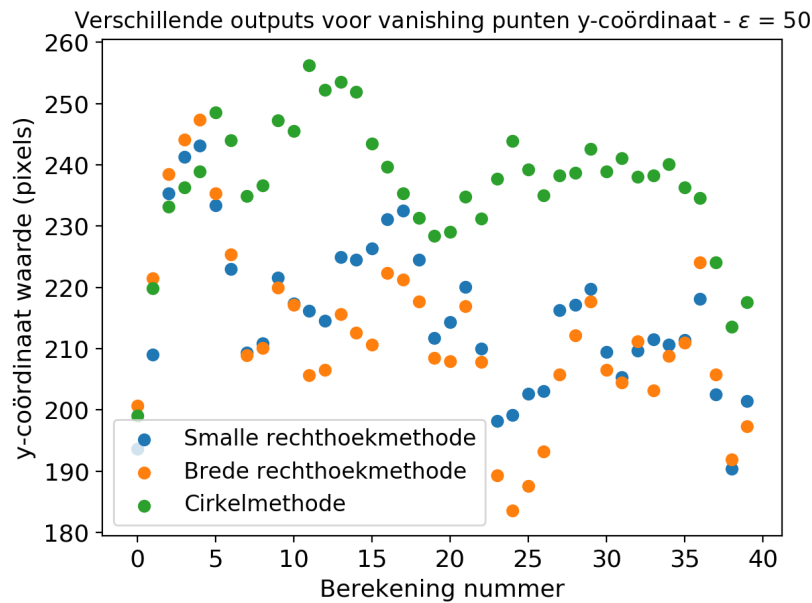
(b) Rechthoekmethode

Figuur 57: Kalibratiechecker twee strategieën

Voor ϵ wordt een waarde gekozen van 50 pixels, waarvan de resultaten zijn weergegeven in Figuur 58 voor de cirkelmethode en rechthoekmethode (voor ϵ en $2 \cdot \epsilon$).

Voor de smalle rechthoekmethode met een breedte van 50 pixels varieert de hoogte van het gevonden vanishing point van 200 tot 245 pixels, wat neerkomt op een onzekerheid van 45 pixels. In het ideale geval moeten al deze vanishing points zich namelijk op dezelfde pixellocatie bevinden. De overige methoden geven ook een weinig accuraat resultaat. Als gevolg hiervan zal dit zich accumuleren in een significante afstandsfout indien de grondvlakmethode wordt gebruikt.

Indien er wordt aangenomen dat door een foute keuze van de tilthoek de absolute fout op de hoogteligging van de bounding box 5 pixels bedraagt, is volgens formule (10) de



Figuur 58: Kalibratiechecker test

afwijking op de tilthoek ongeveer 0,29 graden bij een focale lengte van 990. Bovendien komt bij een camera met een verticale openingshoek van 70° voor een 1920×1080 beeld een verschil van 1° bijgevolg overeen met een verschil van 15,43 pixels.

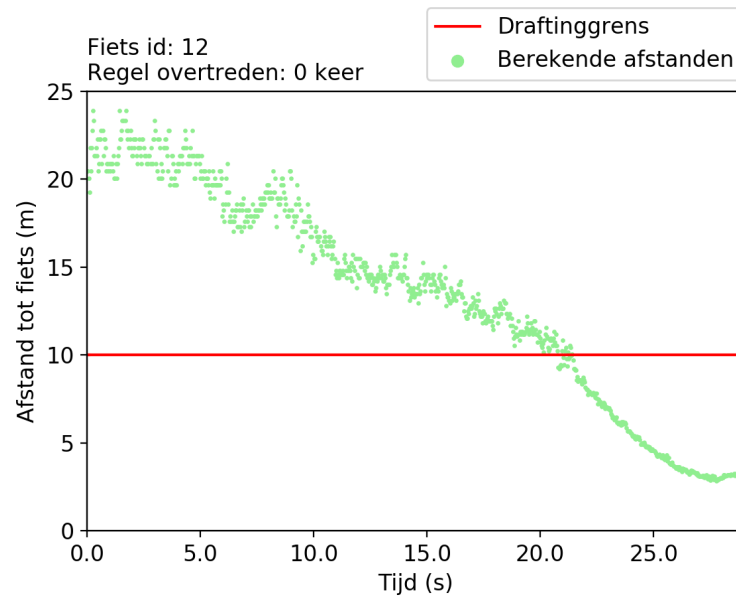
Een verdere evaluatie en testing van beide methoden voor de afstandsbepaling, wordt in het volgend hoofdstuk besproken.

7.3 Filteren van de afstandsberekeningen

In de vorige onderdelen met betrekking tot afstandsbepaling is er telkenmale uitgegaan van een afstand berekend in elk beeld. Dit resulteert in een ruzig signaal zoals weergegeven is in het groen in Figuur 60. In deze grafiek wordt op de y-as de berekende afstand getoond in relatie tot de tijd op de x-as. De rode constante betreft de afstandslimiet voor drafting die in dit geval op 10 meter is geplaatst. Er kan uit de betreffende figuur worden afgeleid hoeveel seconden deze triatleet aan het draften is, wat in deze situatie reeds 8,17 seconden bedraagt.

Door de informatie uit meerdere frames te combineren kan de robuustheid van de afstandsestimatie verbeterd worden. Het afstandssignaal wordt hiervoor gefilterd met als doel eventuele *outliers* en ruis te verwijderen. Er zijn verschillende mogelijke filters om dit te realiseren:

- Gemiddelde-waardefilter
- Particlefilter [86]

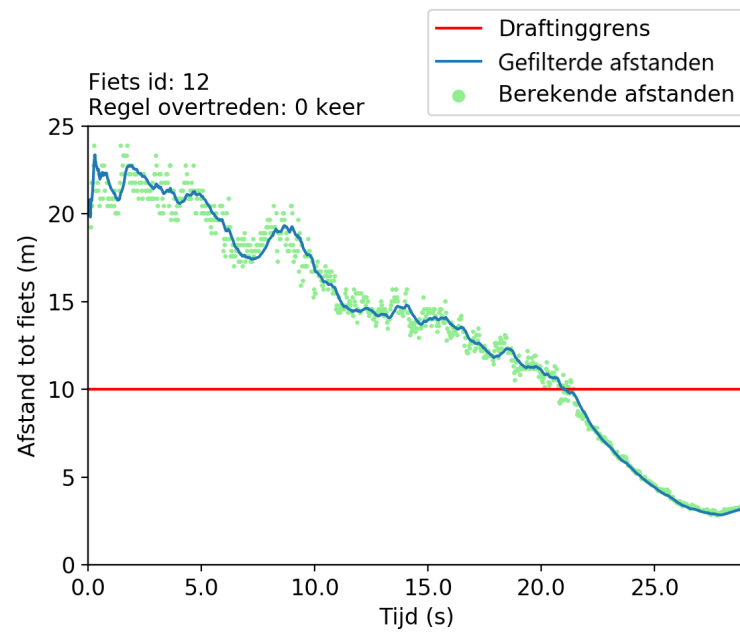


Figuur 59: Berekende afstand in functie van de tijd - zonder filtering

- Mediaanfilter
- Kalmanfilter [67]

In deze implementatie wordt er gekozen voor de laatstvermelde filter. Het voordeel is dat deze in realtime op basis van de vorig berekende staat en zijn covariantiematrix (prior) een uitkomst kan bepalen [87]. Deze aanpak kan echter alleen gebruikt worden in lineaire of stabiele systemen met een Gaussiaans verdeelde ruis zoals besproken in onderdeel 6.2.4.1. Aangezien de fluctuatie in snelheid en bijgevolg de afstand gedurende een triatlonwedstrijd nagenoeg miniem is, is het gebruik van deze filter te verantwoorden. Het resultaat na filtering is in het blauw weergegeven in Figuur 60. Voor de output van enkel situaties tijdens een triatlonwedstrijd met de bijhorende afstandsgrafiek, wordt verwezen naar Bijlage F.

In het volgend hoofdstuk worden de resultaten besproken na testing en wordt er een gefundeerde keuze gemaakt voor één van de twee methoden voor de afstandsbeplating.



Figuur 60: Berekende afstand in functie van de tijd - na toepassen van het Kalmanfilter

Hoofdstuk 8

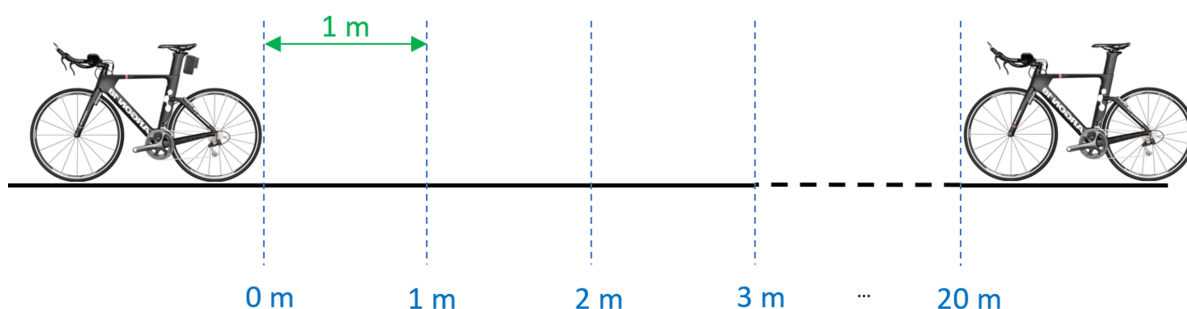
Testing en resultaten

In de voorbije hoofdstukken is besproken welke bouwstenen er nodig zijn om tot een computervisiegebaseerd draftingdetectiesysteem te komen. Voor de afstandsbepaler zijn er twee strategieën voorgesteld: de hoogtemethode en de grondvlakmethode. Beide methoden worden in dit hoofdstuk verder geanalyseerd en gebenchmarkt om bijgevolg een gefundeerde keuze te maken met oog op robuustheid en nauwkeurigheid. Om dit te realiseren zijn twee soorten testopnames gemaakt en besproken: een statische en een dynamische testopname. Na een grondige analyse van het detecteren van drafting over een bepaalde tijdsperiode wordt de robuustheid van het systeem beduid en worden enkele tekortkomingen aangehaald.

8.1 Nauwkeurigheid afstandsberekening

8.1.1 Statische situatietest

Een eerste analyse om de nauwkeurigheid van de berekende afstand te bepalen betreft een statische situatie voor verschillende gekende referentieafstanden. Er worden hiervoor afstanden beschouwd van 0 meter tot en met 20 meter met een verschil van één meter, deze vormen de grondwaarheid van dit experiment. Er is met andere woorden een reeks van beelden opgenomen waarbij een fiets telkens wordt verplaatst zoals geïllustreerd in Figuur 61¹. Aan de hand van de bekomen beelden opgenomen door de camerafiets (links



Figuur 61: Statische opstelling - principe met een camerafiets en een achterliggende fiets voor de statische situatietest (eigen bewerking van [84])

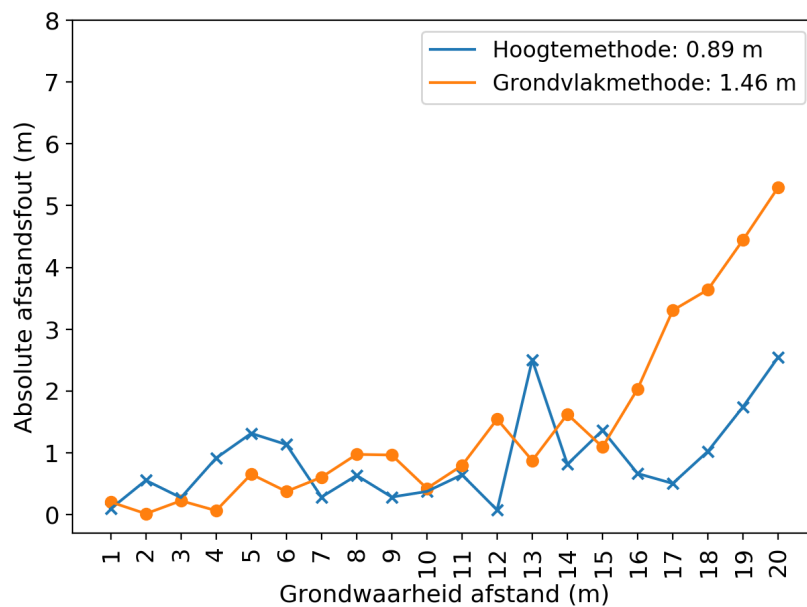
¹Een voorbeeld van de beeldregistratie van deze test is weergegeven in Figuur 100 in Bijlage G

in Figuur 61), wordt de afstand tot de achterligger bepaald (rechts in Figuur 61). In de huidige implementatie wordt er één meting per frame verricht. Voor deze statische test wordt vervolgens voor elke grondwaarheidsafstand een gemiddelde genomen van de gemeten afstand over een 100-tal frames. De resultaten van de hoogtemethode en de grondvlakmethode zijn weergegeven in Tabel 5.

Tabel 5: Statische opmeting - afstandsmetingen voor de hoogte- en grondvlakmethode voor grondwaarheidsafstanden van 0 tot 20 meter

Grondwaarheidsafstand (m)	Berekende afstand (m)		Absolute afstandsfout (m)	
	Hoogtemethode	Grondvlakmethode	Hoogtemethode	Grondvlakmethode
1	1,10	0,80	0,10	0,20
2	2,56	1,99	0,56	0,01
3	2,73	2,78	0,27	0,22
4	4,92	4,07	0,92	0,07
5	6,32	4,35	1,32	0,65
6	7,14	6,38	1,14	0,38
7	6,72	6,40	0,28	0,60
8	7,37	7,03	0,63	0,97
9	8,72	8,04	0,28	0,96
10	10,38	10,43	0,38	0,43
11	11,65	10,21	0,65	0,79
12	12,08	10,46	0,08	1,54
13	15,51	13,88	2,51	0,88
14	14,82	12,38	0,82	1,62
15	16,37	13,91	1,37	1,09
16	16,67	13,97	0,67	2,03
17	16,50	13,70	0,50	3,30
18	16,98	14,36	1,02	3,64
19	17,26	14,56	1,74	4,44
20	17,45	14,71	2,55	5,29

Deze resultaten kunnen in een grafiek uitgezet worden met op de x-as de grondwaarheidsafstand en op de y-as de absolute fout op de afstand, zoals in Figuur 62 weergegeven. In de legende is het gemiddelde van de absolute fout over alle grondwaarheidsafstanden weergegeven die voor de hoogte- en grondvlakmethode



Figuur 62: Statische opmeting - absolute afstandsfout voor de hoogtemethode en grondvlakmethode, waarbij in de legende het gemiddelde van de absolute fout over alle grondwaarheidafstanden is beschouwd voor beide methoden

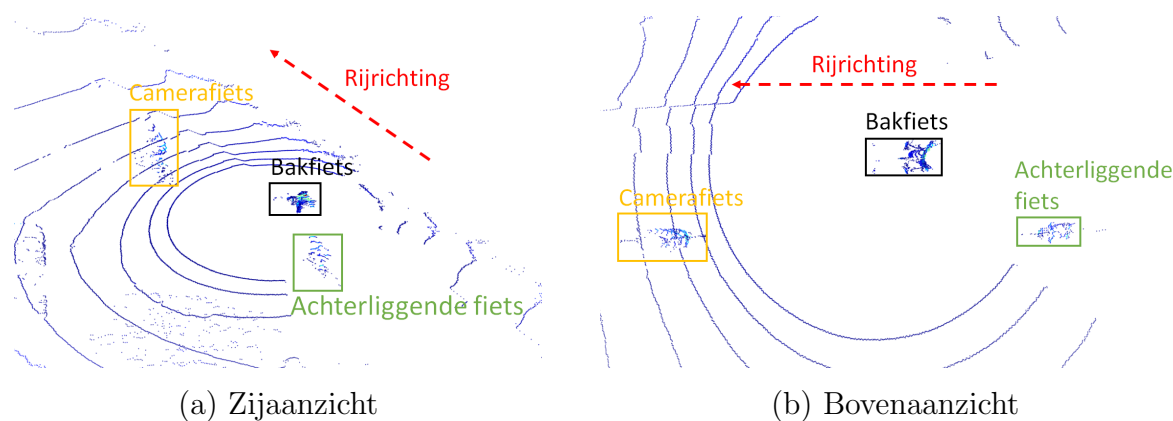
respectievelijk 0,89 m en 1,46 m bedraagt. Uit deze grafiek kan worden afgeleid dat voor afstanden kleiner dan of gelijk aan 6 meter beter de grondvlakmethode wordt gehanteerd daar de absolute afstandsfout er gemiddeld gezien kleiner is. In de andere gevallen geeft de hoogtemethode een relatief lagere absolute afstandsfout als gevolg van een duidelijke invloed op de gevoeligheid van de tilthoek voor de grondvlakmethode. Deze metingen zijn echter gerealiseerd met een gekende tilthoek, bijgevolg is er geen tilhoekafwijking verondersteld. Als de tilthoek 1° graad verschilt, dan komt dit neer op een gemiddelde absolute afstandsfout over alle grondwaarheidafstanden van 2,54 meter voor de grondvlakmethode.

Deze situatietest geeft een goede indicatie over de afwijking op de gemeten afstand van het draftingdetectiesysteem. Uit deze test blijkt dat de hoogtemethode een betere keuze is dan de grondvlakmethode. Enerzijds geeft deze betere resultaten en anderzijds vanuit praktisch oogpunt is deze onafhankelijk van de tilthoek.

Het nadeel van deze statische opmeting is dat deze opnameopstelling verschillend is van wat er gebeurt tijdens een triatlonwedstrijd. Bovendien is een evaluatie voor de zone van ongeveer 8 tot 12 meter van belang daar de draftingregel 10 meter voorschrijft als limiet. De gehele referentieafstanden vormen daarenboven een beperking aangezien er telkens sprongen van één meter worden gemaakt. Om deze belemmeringen uit de weg te gaan, wordt een dynamische situatietest besproken in het volgend onderdeel.

8.1.2 Dynamische situatietest

Om het algoritme in een meer dynamische omgeving te testen, wordt gebruikgemaakt van een LIDAR-systeem opgesteld op een bakfiets om een wedstrijdssituatie te simuleren². In deze testopstelling wordt een vlakke weg beschouwd zonder hellingen. De LIDAR van het type *Velodyne VLP 16* levert een 3D-puntenwolk op van de omgeving waardoor de gemeten afstand van het draftingdetectiesysteem kan worden vergeleken met de LIDAR-afstand. Deze afstanden zijn manueel vanaf het bovenaanzicht gemeten aan de hand van de *Veloview* tool die in deze test als grondwaarheid wordt gebruikt³. De opstelling is weergegeven in Figuur 63 waarbij ‘Camerafiets’ de fiets is met de camera aan het zadel gemonteerd. ‘Achterliggende fiets’ is de fiets die achter de camerafiets rijdt en bijgevolg al dan niet aan het draften is. De rijrichting van de drie fietsers is ter verduidelijking aangegeven met een rode pijl in de beschouwde illustratie.



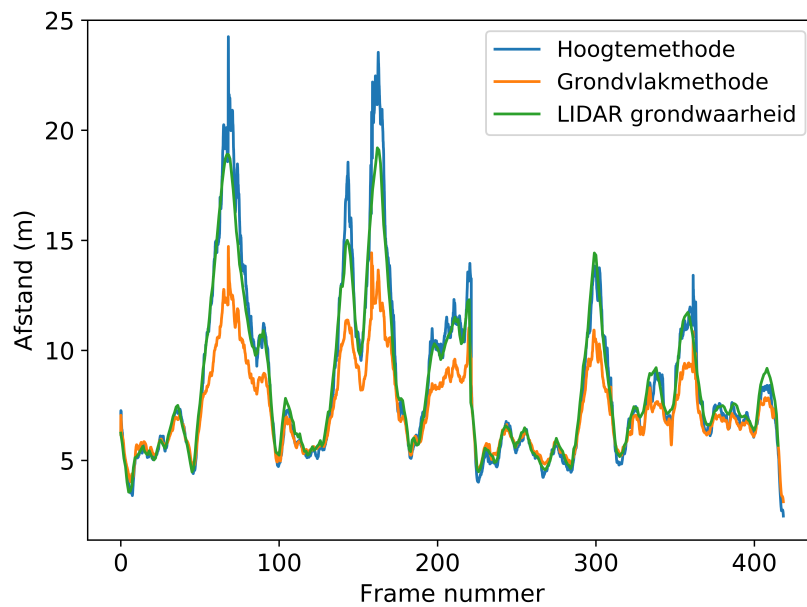
Figuur 63: Dynamische opmeting - LIDAR-opstelling met een bakfiets, camerafiets en een achterliggende fiets voor de dynamische situatietest

De gebruikte LIDAR draait rond aan 10 Hz zodoende een horizontale FOV van 360 ° te bedekken. Er worden bijgevolg 10 afstandsmetingen per seconde beschouwd over 416 frames zoals weergegeven in Figuur 64.

In deze figuur kan afgelezen worden hoe de afstand varieert van ongeveer 3 meter tot 19 meter in vergelijking met de grondwaarheid. Er kan bovendien duidelijk opgemerkt worden dat de grondvlakmethode een grotere afwijking heeft ten opzichte van de grondwaarheid in vergelijking met de hoogtemethode. Echter is dit voor de grondvlakmethode in het geval van een perfecte tilthoek wat in de praktijk uitgesloten is, zoals ook is gebleken uit de analyse met betrekking tot de horizonligging in Hoofdstuk 7.

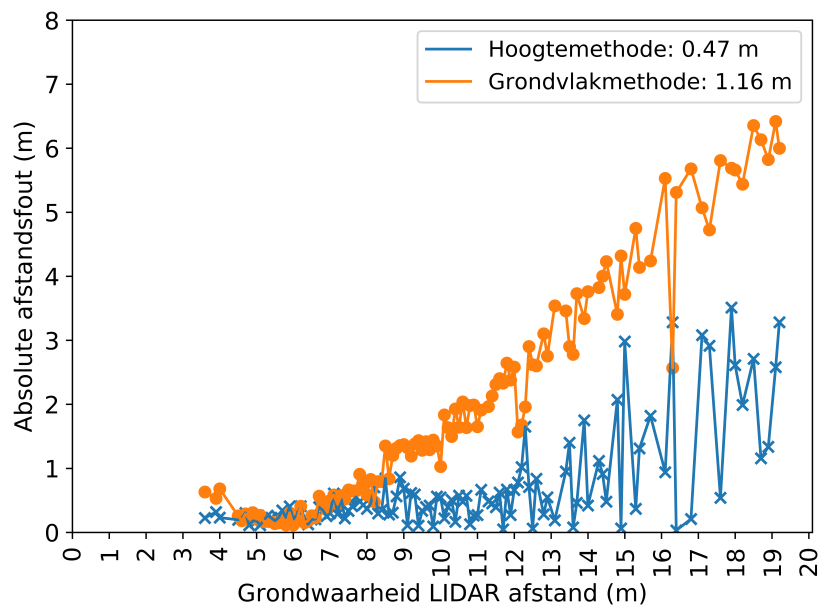
²Een voorbeeld van de beeldregistratie van deze test is weergegeven in Figuur 101 in Bijlage G

³Aangezien de afstand wordt gemeten vanaf de camera tot aan het stuur van de achterliggende fiets, is de grondwaarheidsafstand van de LIDAR ook op die manier aangeduid.



Figuur 64: Dynamische opmeting - LIDAR-afstanden in vergelijking met de afstanden van de hoogtemethode en grondvlakmethode

Figuur 65 geeft de absolute afstandsfout weer berekend over alle grondwaarheidsafstanden tot op 0,10 meter nauwkeurig voor zowel de hoogtemethode als de grondvlakmethode.

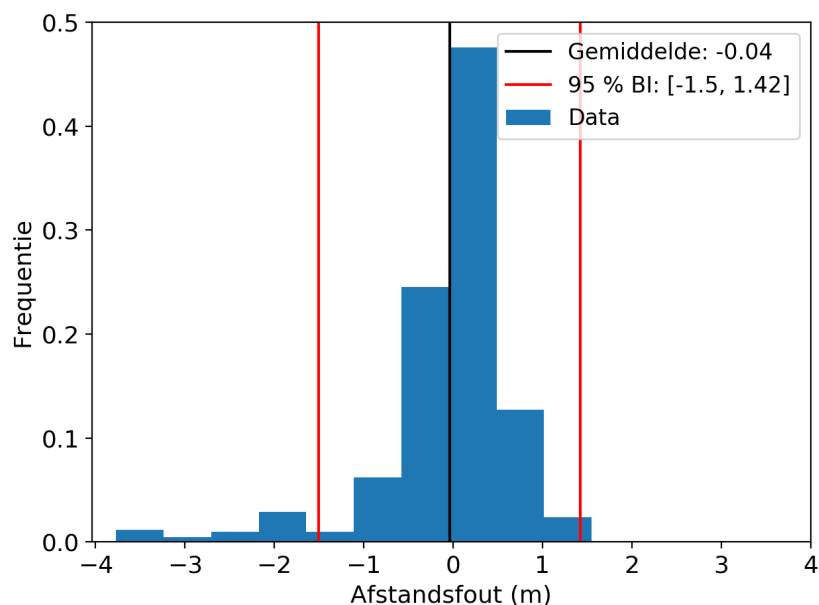


Figuur 65: Dynamische opmeting - absolute afstandsfout voor de hoogtemethode en grondvlakmethode, waarbij in de legende het gemiddelde van de absolute fout over alle grondwaarheidsafstanden is beschouwd voor beide methoden

In de legende is de gemiddelde absolute afstandsfout weergegeven over alle grondwaarheidafstanden. Daaruit blijkt dat de hoogtemethode wederom nauwkeuriger is dan de grondvlakmethode met respectievelijk een gemiddelde absolute fout van 0,47 meter en 1,16 meter. In het geval er een tilthoekafwijking wordt gekozen van één graad, blijkt dat de gemiddelde afstandsfout over alle grondwaarheidafstanden bijgevolg 1,96 meter bedraagt voor de grondvlakmethode. Met andere woorden, ook in de dynamische situatietest blijkt de hoogtemethode de aangewezen keuze.

Zoals eerder is gesteld, is het vooral de zone rondom de grens van 10 meter van de draftingszone die van belang is. Als er gekeken wordt naar de zone van 8 meter tot 12 meter gezien vanaf het voorwiel van de camerafiets, dan blijkt dat de gemiddelde absolute afstandsfout over dit bereik van grondwaarheidafstanden 0,42 bedraagt voor de hoogtemethode.

Over alle afstanden heen kan er een 95%-betrouwbaarheidsinterval (BI) beschouwd worden zoals weergegeven in Figuur 66. Volgens de betreffende figuur ligt 95 % van de



Figuur 66: Dynamische opmeting - histogram van gemiddelde afstandsfouten met 95%-betrouwbaarheidsinterval

waarnemingen in het interval $[-1,50; 1,42]$. Het gemiddelde ligt op $-0,04$ meter, er worden dienvolgens over het algemeen meer negatieve dan positieve fouten gemaakt wat veroorzaakt wordt door de imperfectie van objectdetector. Het negatieve komt in dit geval overeen met een afstand die te groot is ingeschat in vergelijking met de

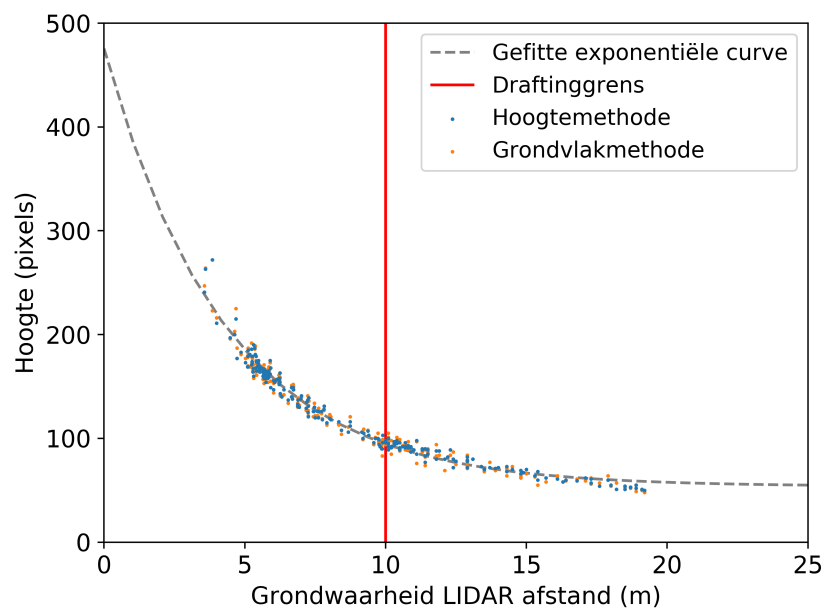
grondwaarheid. Deze grenzen kunnen in een eindimplementatie gebruikt worden als hysteresisgrenzen. Anders gezegd, de tijd die in de draftingzone wordt verbleven, wordt pas gemeten als er op een minimum afstand van 8,50 meter wordt gereden.

In de verdere analyse van het draftingdetectiesysteem wordt een gemiddelde absolute afstandsfout van 0,47 meter beschouwd, wat een verbetering is van minimaal 0,85 meter ten opzichte van de relatief gemeten afstandsfout in een implementatie op basis van GPS zoals besproken in Hoofdstuk 2.

Om de relatie tussen de absolute fout op de afstand te relateren aan deze van de bounding box wordt de formule voor de hoogtemethode (9) theoretisch afgeleid zoals in formule (12).

$$\frac{\partial x}{\partial y} = -\frac{f \cdot h_2}{y^2} \quad (12)$$

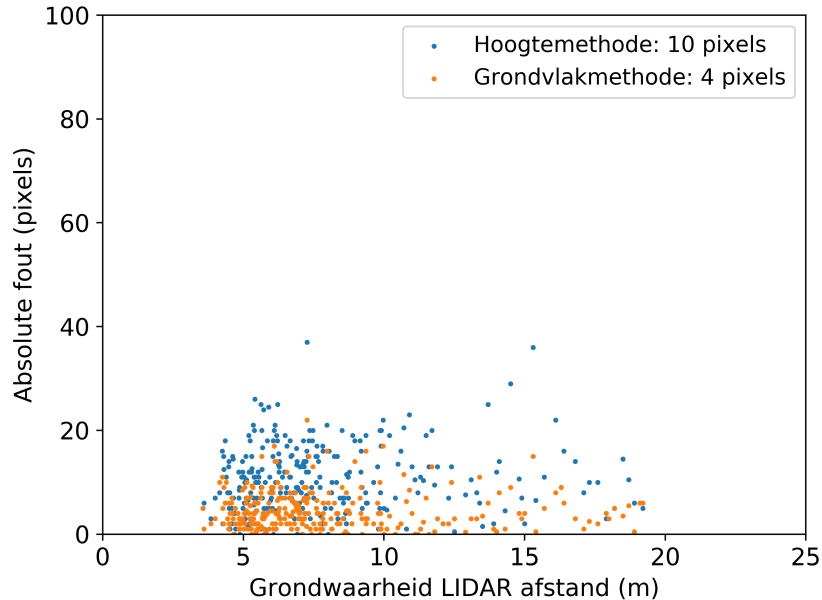
Er kan hieruit afgeleid worden dat hoe kleiner de hoogte van de bounding box wordt, hoe groter $\frac{\partial x}{\partial y}$ en bijgevolg hoe belangrijker de afwijking wordt op de afstand. Dit stemt tevens overeen met het feit dat hoe verder een fiets is verwijderd, hoe moeilijker het algoritme dit nauwkeurig kan detecteren. Dit kan ook afgeleid worden uit Figuur 67 en Figuur 68.



Figuur 67: Hoogte in pixels van de bounding box voor verschillende grondwaarheidsafstanden voor de hoogte- en grondvlakmethode

Figuur 67 geeft de pixelhoogte weer in functie van de grondwaarheidsafstand voorzien door de LIDAR-sensor voor zowel de hoogte- als grondvlakmethode. In het rood is de

draftinggrens van 10 meter aangegeven. Hoe verder een fiets zich bevindt, des te kleiner zijn bounding box is; de grijze stippellijn geeft dit exponentieel dalend gedrag aan. Als gevolg hiervan is bij grotere afstanden de invloed van de hoogte(ligging) van de bounding box veel gevoeliger in vergelijking met kleinere afstanden. In Figuur 68 is voor



Figuur 68: Absolute fout op de y-coördinaat voor verschillende afstanden voor de hoogte- en grondvlakmethode, waarbij in de legende de gemiddelde absolute fout over alle grondwaarheidafstanden van beide methoden is weergegeven

verschillende grondwaarheidafstanden op de x-as de absolute fout in pixels op de hoogte van de bounding box (hoogtemethode) en de hoogteligging ervan (grondvlakmethode) weergegeven. In de legende is de gemiddelde absolute fout van de pixelhoogte over alle grondwaarheidafstanden berekend, weergegeven die respectievelijk 10 en 4 pixels bedraagt voor de hoogtemethode en de grondvlakmethode.

Een gelijkaardige redenering kan gemaakt worden om de invloed van de tilthoek op de grondvlakmethode te analyseren. De partieel afgeleide van formule (11) wordt gegeven door formule (13).

$$\frac{\partial x}{\partial(\alpha + \theta_{tilt})} = -\frac{h_1}{\tan^2(\alpha + \theta_{tilt}) \sin^2(\alpha + \theta_{tilt})} \quad (13)$$

Uit deze formule blijkt: hoe groter de afwijking op de tilthoek, een negatieve en positieve afwijking, hoe belangrijker de afwijking wordt op de afstand x .

Samengevat kan er besloten worden dat gemiddeld gezien de grondvlakmethode een

kleinere absolute fout voor de hoogteligging van de bounding box ondervindt indien de tilthoek perfect is gekozen. Aangezien een tilthoekbepaling moeilijker nauwkeurig kan gebeuren en behouden blijven gedurende een wedstrijd, biedt de hoogtemethode uiteindelijk betere resultaten in de praktijk met betrekking tot de afstand.

In het volgend onderdeel wordt er dieper ingegaan op het detecteren van draften voor meerdere metingen in de tijd. Meer bepaald wordt een score bepaald hoe goed het systeem werkt om drafting te detecteren bekeken voor 20 seconden, daar dit de minimale tijd is die een triatleet zich in de draftingzone mag begeven.

8.2 Draftingdetectiescore voor 20 seconden

Om de uiteindelijke score van het draftingdetectiesysteem te bepalen, wordt de kans bepaald zoals weergegeven is in formule (14).

$$P(\text{draftingregel overtreden} | \text{renner in draftingzone over een bepaald tijdsinterval}) \quad (14)$$

Aan de hand van de errorfunctie (erf of *error function*) kan de kans bepaald worden dat de echte afstand kleiner is dan 10 meter en de atleet zich bijgevolg in de draftingzone bevindt, zoals aangegeven is in formule (15).

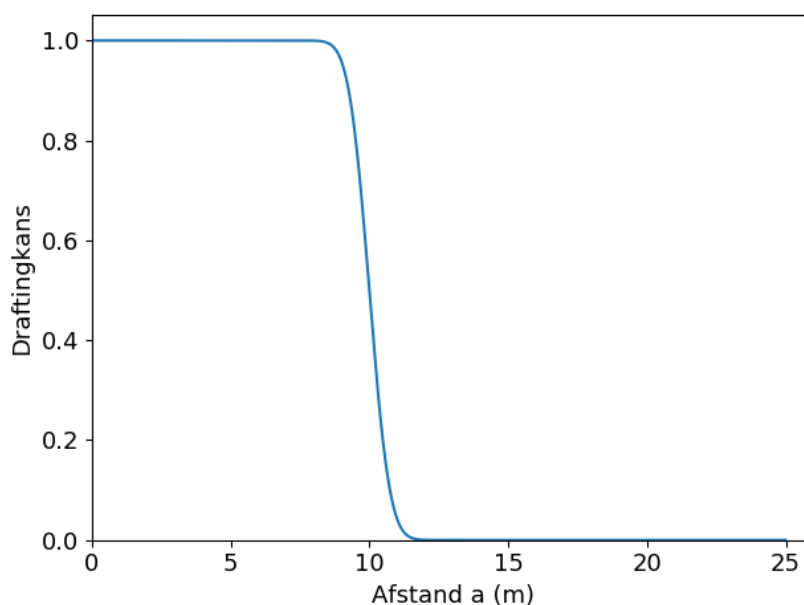
$$P(a, \sigma) = \frac{\sigma - \sigma \operatorname{erf}\left(\frac{-10+a}{\sqrt{2}\sigma}\right)}{2\sigma} \quad (15)$$

In deze formule betreft σ de standaardafwijking op de afstand wat in dit geval 0,58 meter bedraagt. Figuur 69 geeft de kansverdeling weer van deze functie voor een triatleet die zich in de draftingzone bevindt als de gemeten afstand a betreft.

Deze kans geeft aanleiding tot een binomiale kansverdeling met het verschil dat:

1. Het overtreden van de draftingregel overeenstemt met het verblijven van minimum 20 opeenvolgende seconden in de draftingzone. De metingen zijn bijgevolg afhankelijk van elkaar, er is een correlatie tussen de metingen.
2. De kans om zich in de draftingzone te bevinden, hangt af van de gemeten afstand a .

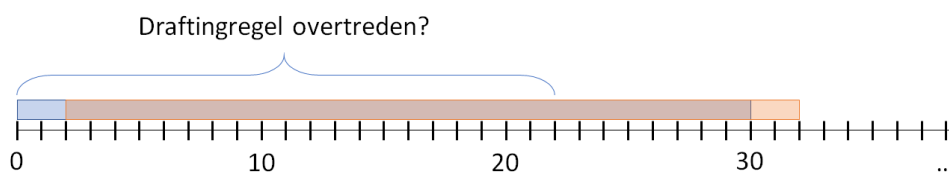
De afstand wordt bijvoorbeeld berekend om de 2 seconden over een periode van 20 seconden, zodat er 1 meting per seconde wordt volbracht. Met andere woorden, de draftingregel is overtreden als er minimum 11 opeenvolgende instanties zijn waar de triatleet zich in de draftingzone bevond. Een andere mogelijkheid is om de afstand om de 4 seconden te berekenen over een periode van 40 seconden. In dit geval heeft de



Figuur 69: De verdeling van de kans dat een triatleet zich in de draftingzone van 10 meter bevindt voor een gemeten afstand a

fietser de draftingregel overtreden indien hij of zij zich in de draftingzone bevond voor minimum 6 opeenvolgende instanties.

Kortom kan er gevarieerd worden in het aantal afstandsmetingen per seconde (bijvoorbeeld 1 meting om de 2 seconden) en de grootte van het venster waarover wordt gekeken (bijvoorbeeld 30 seconden)⁴. De draftingregel is bijgevolg overtreden als er 11 van de 15 samples opeenvolgend in de draftingzone liggen. Dit venster kan dan telkens bijvoorbeeld om de 2 seconden worden opgeschoven zoals principieel weergegeven in Figuur 70.



Figuur 70: Principe om de draftingdetectiescore over 20 seconden te bepalen gemeten over een venster van 30 seconden en een afstandsmeting om de 2 seconden

⁴In Hoofdstuk 9 wordt een benadering aangehaald om de venstergrootte en het aantal samples per seconde met oog op de schatting van de probabiteit te bepalen.

Om de aanwezigheid in de draftingzone te bepalen, wordt gebruikgemaakt van draftingpatronen elk met een lengte van 15 eenheden. Deze patronen zijn opgebouwd door alle combinaties te beschouwen van 15 bits waarvan er minimum 11 opeenvolgende bits '1' moeten zijn om een geldig patroon te vormen. Op die manier kunnen er 104 mogelijke draftingpatronen opgesteld worden. De eerste tien patronen zijn weergegeven in formule (16).

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
 \end{bmatrix} \tag{16}$$

Het eerste patroon geeft aan dat de atleet zich in de draftingzone bevond voor de laatste 11 samples, en niet in de zone voor de eerste 4 instanties. Dit komt overeen met een geldig patroon, de drafting regel is in dat geval overtreden. Een ongeldig patroon kan zijn: [010101011111000].

Om de kans te bepalen dat de draftingregel is overtreden, wordt een reeks van 15 gesampled afstanden beschouwd (bijvoorbeeld: 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99; 8,79; 8,61; 8,13; 8,06; 7,91; 7,64; 7,56; 7,39). De probabilliteit dat het eerste draftingpatroon voorkomt, bedraagt 0,059 %. Daarnaast komt het patroon [111111111111111] overeen met de grootste probabilliteit, namelijk 14,25 %.

Bijgevolg komt drafting voor als minimum één van de draftingpatronen voorkomt. Aangezien deze draftingpatronen mutueel exclusief zijn, kunnen de probabilliteiten van alle patronen opgeteld worden om de uiteindelijke probabilliteit te vinden dat minimum één van deze, , ongeachte welk patroon, is voorgekomen. In het geval van het gegeven voorbeeld komt dit overeen met een probabilliteit van 88,36 %.

Samengevat wordt voor elke 30 seconden voor elke renner om de 2 seconden de kans bepaald dat de draftingregel is overtreden. De resultaten voor enkele situaties, waarbij de afstand om 2 seconden is gesampled, zijn weergegeven in Tabel 6.

Op basis van de draftingscore, berekend over het beschouwd tijdsinterval, kan de toekenning van een draftingpenalty worden gerealiseerd. Deze aanpak heeft als voordeel dat met oog op het gebruik van apparaten met een beperkte rekenkracht er minder

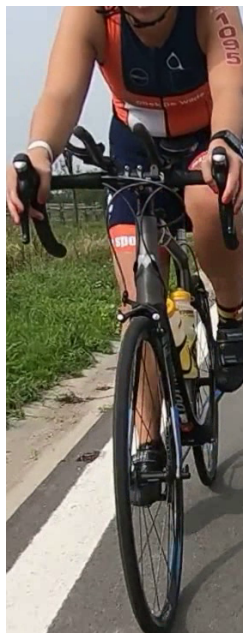
Tabel 6: Gesamplede afstanden met de kans dat de draftingregel is overtreden

Gesamplede afstanden (m)	Drafting-score
9,37; 9,82; 10,16; 10,84; 10,67; 10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69	0,0070
9,82; 10,16; 10,84; 10,67; 10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88	0,0081
10,16; 10,84; 10,67; 10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87	0,0130
10,84; 10,67; 10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97	0,0291
10,67; 10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30	0,2352
10,98; 10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,3; 9,39	0,6394
10,48; 10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14	0,8170
10,83; 10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98	0,8303
10,42; 9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62	0,8324
9,59; 9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93	0,8289
9,07; 8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19	0,8038
8,56; 8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99	0,6163
8,45; 7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99; 8,79	0,6312
7,83; 7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99; 8,79; 8,61	0,8235
7,69; 7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99; 8,79; 8,61; 8,13	0,8697
7,88; 7,87; 7,97; 8,30; 9,39; 9,14; 9,98; 9,62; 9,93; 9,19; 8,99; 8,79; 8,61; 8,13; 8,06	0,8794
...	...
9,98; 9,62; 9,93; 9,19; 8,99; 8,79; 8,61; 8,13; 8,06; 7,91; 7,64; 7,56; 7,39; 7,72; 9,05	0,9565

berekeningen moeten uitgevoerd worden. Het gedeelte met betrekking tot deze apparaten wordt in Hoofdstuk 9 als toekomstmogelijkheid behandeld.

8.3 Het nemen van een screenshot

Indien een triatleet zich langer dan 20 seconden in de draftingzone bevindt of langer dan 5 seconden indien de persoon is voorbijgestoken, dan is de draftingregel overtreden en wordt er een draftingpenalty toegekend. In de huidige implementatie wordt een screenshot opgeslagen waarvan Figuur 71 een voorbeeld is. Andere invullingen voor het



Figuur 71: Screenshot van een triatleet die de draftingregel heeft overtreden

toekennen van straf tijd of andere meldingen voor een eindontwerp zijn ook mogelijk zoals: het knipperen van een led, het draadloos doorsturen van een signaal, ...

8.4 Robuustheid testing

Naast het nauwkeurig kunnen detecteren van fietsen en bijgevolg de afstand tot deze te bepalen, moet het systeem tevens robuust kunnen werken in realistische situaties zoals vermeld in Hoofdstuk 4. Op een paar uitdagingen na werkt het systeem naar behoren na visuele testing op de verschillende beschikbare wedstrijdbeelden van de dataset:

- **Hellingen: Figuur 17 (f)**

Voor het draftingdetectiesysteem wordt de meest nauwkeurige afstandsmethode gehanteerd, namelijk de hoogtemethode. Het voordeel is dat naast een gemiddeld kleinere absolute afstandsfout in vergelijking met de grondvlakmethode, deze onafhankelijk is van een verandering van de tilthoek als gevolg een helling, een brug of het verdraaien van de camera.

- **Belichtingsverandering door een tunnel of *flash* cameraverblinding als gevolg van zonnestrallen: respectievelijk Figuur 17 (c) en (d)**

Door het realiseren van een continue monitoring met betrekking tot fietsdetectie, worden de False Negatives die ontstaan in deze situatie opgevangen.

- **Kortstondige bewegingsonscherpte door een drempel of een oneffen wegdek: Figuur 17 (a)**

Aangezien deze situatie veelal van korte duur is en in beperkte mate voorkomt tijdens de wedstrijden uit de dataset, vormt dit geen heikel punt voor het systeem als gevolg van de continue monitoring. Door de EIS-functionaliteit van de gebruikte camera, is dit bij de opnames tot een minimum beperkt kunnen worden.

- **Het heen en weer bewegen van de fiets: Figuur 17 (g) en (h)**

Indien er op de trappers heen en weer wordt gefietst, zal het beeld ook abrupt van links naar rechts bewegen en omgekeerd. Dit vormt in de praktijk geen probleem als het achterliggend gedetecteerd object zich niet verder dan ongeveer 15 meter van de camera bevindt.

- **Het nemen van een bocht: Figuur 17 (e)**

In de huidige implementatie wordt een fiets ook in een bocht gevolgd en gecontroleerd op drafting. Aangezien er volgens het triatlonreglement wel mag gedraft worden in deze situatie, worden in Hoofdstuk 9 enkele mogelijke oplossingen hiervoor aangereikt.

- **Omstaanders: Figuur 17 (j)**

Aangezien alleen gefocust is op (triatlon)fietsen bij het trainen, vormt dit voor de objectdetector met bijhorende controlelogica geen hindernis.

De drie overige situaties zoals regendruppels op de cameralens, eventuele valse detecties door een motor en het monitoren van meerdere fietsers doorheen de frames, worden beschouwd als tekortkomingen en worden in de volgende paragrafen besproken.

8.4.1 Cameralensverblinding

Als er in de regen wordt gereden met een koersfiets kunnen er waterdruppels op de lens van de camera terechtkomen zoals in Figuur 17 (b). Als gevolg hiervan kan de objectdetector minder betrouwbaar een fiets detecteren zoals bemerkt kan worden in Figuur 72.



Figuur 72: Onnauwkeurige detectie van een fiets als gevolg van regendruppels op de cameralens

Als gevolg hiervan lijkt het dat deze fietser zich verder bevindt volgens de afstandsmeting met de hoogtemethode dan in werkelijkheid. Een oplossing hiervoor kan zijn door de camera te voorzien van een “kuissysteem” zoals dit ook bij wagens wordt gebruikt. In dit laatstgenoemde wordt de waterdruppel als het ware met perslucht weggeblazen of weggeveegd van de cameralens.

8.4.2 Het volgen van de dichtste fietser

In de huidige implementatie wordt alleen de dichtste fietser gevolgd en gecontroleerd op drafting. In de praktijk zijn er echter situaties dat er meerdere personen tegelijk aan het draften zijn zoals in Figuur 16. In Hoofdstuk 9 wordt hiervoor een oplossing besproken.

8.4.3 Valse detecties

In sommige gevallen blijkt dat het netwerk motors van scheidsrechters, zoals in Figuur 17 (i), als fietsen classificeert. In de meeste gevallen vormt dit geen probleem aangezien deze zich in wedstrijd situaties niet langer dan 20 seconden in beeld begeven. In de

huidige implementatie wordt alleen de dichtste fietser gevolgd waardoor de valse detectie van bijvoorbeeld de motor kan zorgen voor het tijdelijk wegvallen en resetten van een fietser die aan het draften was. Een gelijkaardig effect ontstaat als een andere fietser de huidig gemonitorde fietser in de scène voorbijsteekt.

In het volgende en laatste hoofdstuk wordt een uiteenzetting gegeven over mogelijke uitbreidingen en vervolgmogelijkheden van deze proof of concept voor de detectie van drafting.

Hoofdstuk 9

Toekomst en vervolg

Het draftingdetectiesysteem, besproken in de vorige hoofdstukken van deze scriptie, kan naast de vermelde implementaties nog uitgebreid worden, wat in dit hoofdstuk aan bod komt. In een eerste onderdeel worden enkele opties besproken om een meer nauwkeurige camerakalibratie te bekomen. Nadien wordt de aanzet gegeven voor de mogelijkheid om een bochtdetector en locatiebepaler te integreren gevolgd door een uiteenzetting om meerdere fietsers te volgen. In het laatste onderdeel worden enkele methoden voor het gebruik van apparaten met een beperkte rekenkracht vermeld.

9.1 Automatische camerakalibratie

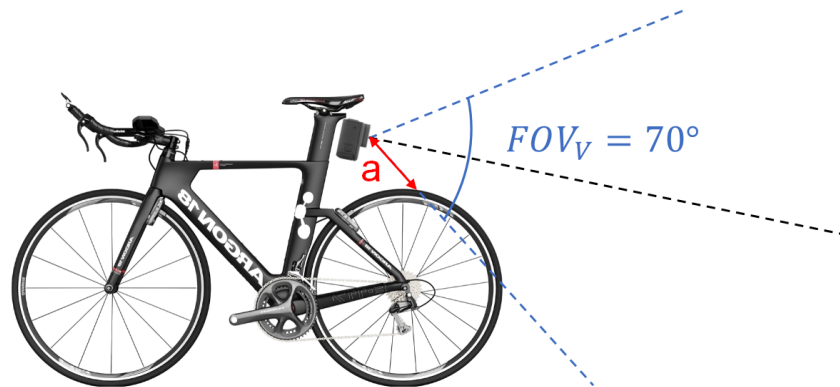
Het grote nadeel van de grondvlakmethode is dat de tilthoek niet nauwkeurig kan bepaald worden. Er is hiervoor in Hoofdstuk 7 een algoritme besproken dat de tilthoek tracht te schatten. Deze aanpak blijkt niet nauwkeurig genoeg, met een grote afwijking voor de afstand als gevolg. Eens deze kalibratie is gerealiseerd, kan de camera door trillingen eventueel verdraaien of kan de horizon verplaatsen als gevolg van een helling, waardoor dit regelmatig gecontroleerd moet worden. De tilthoek moet in deze gevallen vervolgens geüpdatet worden.

Een andere mogelijkheid om de tilthoek te schatten wordt besproken in het volgend subonderdeel en maakt gebruik van het achterwiel dat mee in beeld kan worden genomen.

9.1.1 Vaste afmetingen in de scène

Zoals aangehaald in Hoofdstuk 4 is voor bepaalde beelden van de dataset het achterwiel mee in beeld genomen. De afstand van de camera tot het achterwiel, a in Figuur 73, is normaliter een constante. Dit geldt ook voor het oppervlak aan ‘band’ dat zichtbaar is in beeld zoals in Figuur 73. Het achterwiel kan kortom als ankerpunt gebruikt worden om de verandering van de tilthoek te monitoren. Deze waarden blijven slechts constant als de camera niet verspringt. Op die manier kan er een onderscheid gemaakt worden tussen een camera die verspringt of een helling. Als het gedeelte van de achterband dat in beeld is, verandert, dan moet er een update gebeuren van de camerakalibratie.

Een andere mogelijkheid om deze te tracken is gebruikmaken van periodieke herkenningspunten op het wegdek zoals wegmarkeringen, straatroosters, ... om deze



Figuur 73: Afstand van de camera tot het achterwiel

vervolgens te tracken. Aangezien het wegdek vlak wordt verondersteld, kunnen de herkenningspunten gefit worden aan het grondvlak. Het nadeel is dat deze aanpak niet schaalbaar is in de praktijk door een afhankelijkheid van de specifieke omgeving.

9.2 Situaties waar drafting is toegelaten

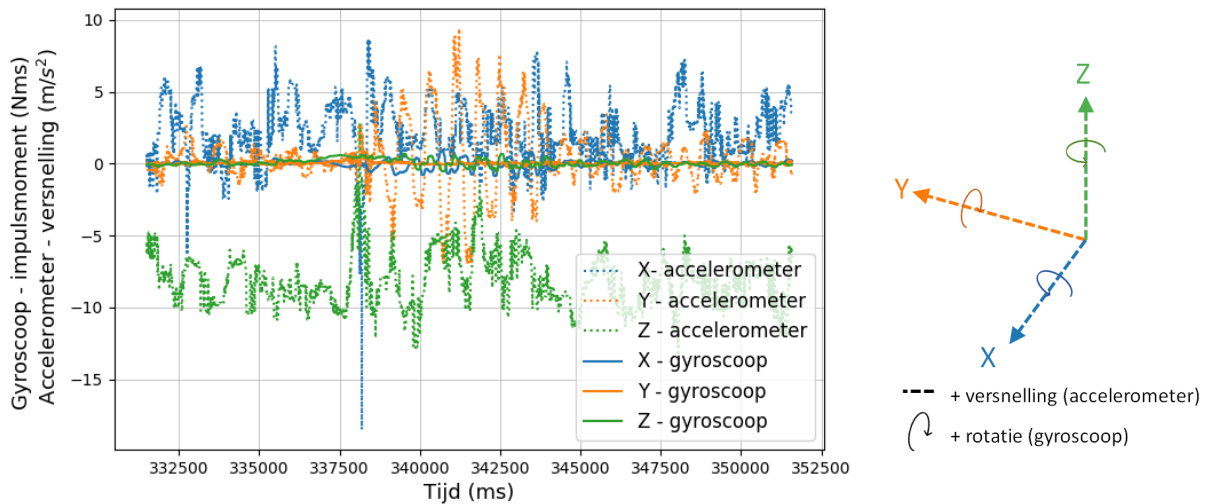
Volgens het triatlonreglement zijn er situaties waar er wel mag gedraft worden in non-drafting wedstrijden [1]:

- *Een scherpe bocht;*
- *Wanneer de atleet een draftingzone betreedt en binnen de 20 sec de zone passeert bij het inhalen;*
- *Omwille van veiligheidsredenen;*
- *100 m voor of na een bevoorradingspost;*
- *100 m voor of na de wisselzone;*
- *Wanneer scheidrechters dit op een bepaald traject toelaten wegens: een te smalle doorgang, wegenwerken, omlleidingen of andere veiligheidsredenen.*

In de volgende paragrafen worden oplossingen aangereikt met betrekking tot deze regels: bochtdetectie, locatiebepaling.

9.2.1 Bochtdetector

Volgens het reglement mag er gedraft worden in een scherpe bocht. Meestal wordt in de praktijk bij elke (merkbare) bocht niet gecontroleerd door de scheidsrechters. Het detecteren van de afstand tot de bounding box zorgt bovendien in een bocht ook voor



Figuur 74: Gyroscoop en accelerometer data in het geval van een bocht

een randeffect, de achterliggende renner lijkt dichter te rijden dan de realiteit. Dit laatste is het gevolg van het parcours dat geen rechte lijn is, waar voor de afstandsbeplanning van het draftingdetectiesysteem wordt vanuit gegaan. Dit geeft aanleiding tot de nood aan een bochtdetector, waarvoor enkele oplossingen worden aangereikt.

- Een eerste mogelijkheid is om aan de hand van de 200 Hz gyroscoop- en accelerometerdata, die bij de opnames zijn opgeslagen, af te leiden dat er een bocht wordt genomen zoals weergegeven in Figuur 74. In deze beschouwde figuur betreffen de stippellijnen de accelerometerdata en de volle lijnen de data van de gyroscoop.

De rotatie rond de Y-as is quasi constant gedurende het hele traject. Als er heen en weer wordt geschommeld met de fiets om snelheid te maken, zal de versnelling volgens deze as corresponderen met een periodiek patroon met stijgende amplitude zoals het geval is tussen secondes 33,75 en 34,50. De X-as komt overeen met een versnelling in de voorwaartse (positief) of achterwaartse (negatief) richting. In het geval van sterk afremmen in de bocht zal er bijgevolg een negatieve piek optreden. Een rotatie volgens deze as komt overeen met een terugkerend periodiek patroon als gevolg van de typische fietsbewegingen en het eventueel op de trappers gaan staan waardoor de rotatie rond deze as tijdelijk groter wordt in amplitude. Indien er een bocht wordt genomen naar rechts zal dit periodiek patroon bij benadering constant blijven met een negatieve waarde. Een bocht naar links stemt overeen met een positieve gyroscoopwaarde volgens de X-as. Een gelijkaardig patroon wordt waargenomen voor rotatie rond de Z-as. De versnelling volgens de Z-as komt overeen met de typische trillingen als gevolg van het wegdek. Bij het indraaien van een bocht zal er bovendien een piek optreden.

Kortom is het mogelijk om patronen uit de gyroscoop- en accelerometerdata te herkennen van typische situaties die zich voordoen tijdens het fietsen zoals het nemen van een bocht in dit geval.

- Een tweede mogelijkheid is om beroep te doen op het Fourier spectrum dat bij het nemen van een bocht zal “draaien” over een bepaalde hoek.
- Een derde mogelijkheid is om rechte lijnen te detecteren met de *Hough transform* en hun draaiing te monitoren. Na evaluatie van het eventueel gebruik van de kalibratiechecker blijkt echter dat deze oplossing niet schaalbaar is wegens de verscheidenheid aan omgevingen: open landschap versus stedelijke omgevingen met veel gebouwen en bijgevolg veel features.
- Een laatste mogelijkheid is om beroep te doen op de richtingsverandering van de optical flow vectoren, gelijkaardig aan de implementatie van de kalibratiechecker. Een bocht wordt bijgevolg gekenmerkt door het naar links richten van de vectoren over een bepaalde tijdsspanne.

9.2.2 Locatiebepaling

Voor de overige geciteerde regels uit het reglement waar drafting is toegestaan, kan er beroep gedaan worden op locatiegebaseerde methoden zoals GPS, wat een nauwkeurigheid van 1 à 2 meter kan hebben. Het nadeel is dat GPS een goede *Line Of Sight* (LOS) vereist en met andere woorden tekort schiet aan accuraatheid in een stedelijke omgeving waar er veel reflecties kunnen optreden. Een andere mogelijkheid is door gebruik te maken van monoculaire visuele odometrie ontwikkeld binnen de onderzoeksgroep [88].

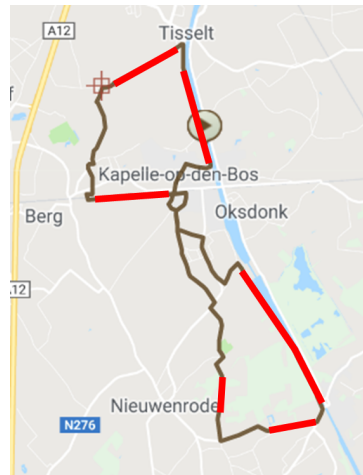
Aan de hand van een combinatie van de vermelde technieken kunnen er zogenaamde non-draftingzones worden ingevoerd, in het rood aangeduid in Figuur 75, waardoor draftingcontrole over een bepaald traject mogelijk is. De organisatie kan deze hotspots dan bepalen of ingeven voor de start van de wedstrijd.

9.3 Uitbreidingsmogelijkheden

In dit onderdeel worden twee uitbreidingsmogelijkheden besproken, respectievelijk het detecteren van meerdere fietsers en een laterale afstandsbepaling.

9.3.1 Meerdere fietsers detecteren en tracken

De draftingdetectie kan ook intrede doen in het wielrennen aangezien wedstrijdmotoren 10 meter afstand moeten houden op achterliggende wielrenners. In dat geval kan het



Figuur 75: Non-draftingzones aangeduid in het rood [89]

draftingdetectiesysteem op de motor worden geplaatst en moeten er bijgevolg meerdere fietsers gedetecteerd en gevolgd worden zoals in Figuur 76 het geval is.



Figuur 76: Het volgen van meerdere fietsers die in een peloton rijden

In de huidige implementatie wordt alleen de dichtste fietser gevolgd doorheen de scènes. Het voordeel van deze aanpak is dat het nauwkeurig volgen van een bepaalde fiets eenvoudig en relatief betrouwbaar is. Het nadeel is echter dat de controle voor draftingdetectie omzeild kan worden door met twee fietsen afwisselend gedetecteerd te worden als de dichtste renner. De objecttracker wordt bovendien gebruikt om de detectierate voor fietsdetectie te verhogen. Indien er meerdere objecten dienen gedetecteerd en gevolgd te worden, moet het neurale netwerk voor objectdetectie enerzijds beter getraind worden op deze situaties en anderzijds moet aan elk object een tracker toegekend worden. Deze laatste stap heeft als gevolg dat de verwerkingssnelheid

met een factor daalt evenredig met het aantal geïnstantieerde objecttrackers en bijgevolg het aantal fietsen in een bepaalde scène. Door gebruik te maken van *threads* kan de verwerking van elke objecttracker overgeheveld worden naar een eigen CPU-core.

9.3.2 Laterale afstandsbepaling

Doorheen deze scriptie is telkens de longitudinale afstand beschouwd van 10 meter, maar een uitbreiding voor de laterale afstand van 3 meter, zoals in Figuur 2 is aangegeven, is tevens mogelijk.

9.4 Apparaten met beperkte rekenkracht

In het geval van een ingebed systeem of constraint device, moet er zuinig omgesprongen worden met rekenkracht, vermogen, throughput, ... Daar CNN's grote computationele eisen stellen moet bijgevolg een goede afweging gemaakt worden tussen deze factoren. Formule (17) definieert de *Figure of Merit* (FoM) en wordt uitgedrukt in aantal correct afgeleide beelden per Joule [90].

$$\text{FoM} = \text{Nauwkeurigheid} \cdot \frac{\text{Throughput}}{\text{Vermogen}} \quad (17)$$

Bijlage H geeft een overzicht met betrekking tot de throughput (in frames per seconde), het energieverbruik en de prijs.

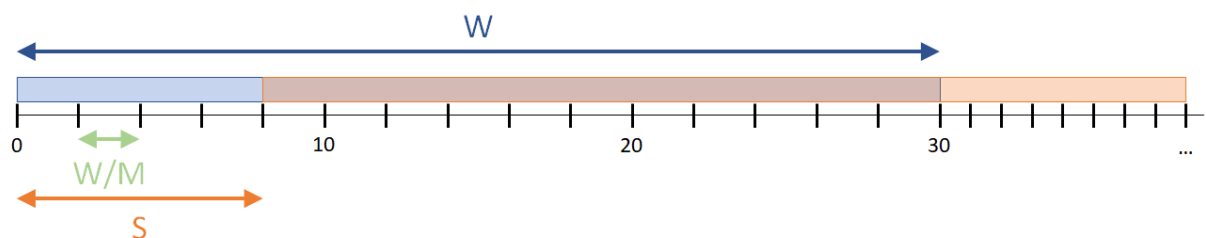
Door het gebruik van een ingebed systeem brengt dit enkele beperkingen met zich mee met betrekking tot de snelheid en het aantal berekeningen per seconde. In onderstaande opsomming worden enkele benaderingen besproken die hiervoor dienst kunnen doen.

- Om de throughput te verminderen kan het aantal fps verlaagd worden waardoor er minder beelden moeten verwerkt worden per seconde.
- In de huidige implementatie wordt de afstand tot een fiets berekend voor elk frame. Dit kan, zoals eerder aangehaald bij de uiteenzetting van de draftingscore, bijvoorbeeld verricht worden om de twee seconden.
- Een tweede mogelijkheid om het aantal berekeningen voor de afstand te minimaliseren betreft het bepalen van de relatieve snelheid van de achterligger ten opzichte van de eigen snelheid. Er wordt nadien geschat wanneer deze ongeveer in de buurt van de draftingzone komt, waarna het aantal afstandsberekeningen weer opgevoerd kan worden. Hetzelfde geldt voor afstanden ver onder de draftinggrens. Er wordt wel verondersteld dat de snelheid van de fietser relatief constant blijft en niet plots kan veranderen. Dit is kortom een uitbreiding van de implementatie van het Kalmanfilter.

De relatieve snelheid kan tevens gebruikt worden om een beter uitsluitsel te geven over mogelijke valse detecties door bijvoorbeeld omstaanders met een fiets die stilstaan of een erg lage snelheid hebben.

- Een andere mogelijkheid om het aantal berekeningen voor de afstand te minimaliseren, is om een ruwe draftingdetectie met GPS te bepalen zoals in een eerdere implementatie van het systeem besproken in Hoofdstuk 2. Vervolgens kan een meer nauwkeurige analyse gemaakt worden met het draftingdetectiesysteem op basis van computervisie indien deze atleet in de buurt van de afstandsgrens voor drafting komt. Dit kan uitgebreid worden door ook de camera niet te laten opnemen wanneer dit niet nodig blijkt om bijgevolg te besparen op de batterij.
- Om de kans te bepalen dat de draftingregel is overtreden, wordt in de huidige implementatie de afstand elke 2 seconden bemonsterd. Vervolgens wordt de probabilliteit bepaald dat er gedraft is voor minstens 20 seconden bekeken over venster van 30 seconden. Er kunnen bijgevolg drie belangrijke bemonsteringsparameters beschouwd worden:
 - W : de lengte van het interval, meer specifiek het aantal seconden waarover de afstand wordt gesampled, kortom het bemonsteringsvenster;
 - M : het aantal samples dat is genomen over het interval W ;
 - S : de stapgrootte tussen twee intervallen in;

Een overzichtsschema van de drie parameters is weergegeven in Figuur 77.



Figuur 77: Overzichtsschema van de drie belangrijke bemonsteringsparameters W , M en S met betrekking tot de bepaling van de draftingscore

Om een goede schatting van de beschouwde probabilliteit te komen, moeten in het ideale geval de parameters W en M groot zijn terwijl S klein is gewenst. Zeer grote waarden voor W en M en kleine waarde voor S leiden echter tot onaanvaardbaar lange rekestijden. Er kan vervolgens aan de hand van simulaties bepaald worden wat redelijke waarden zijn voor de drie bemonsteringsparameters.

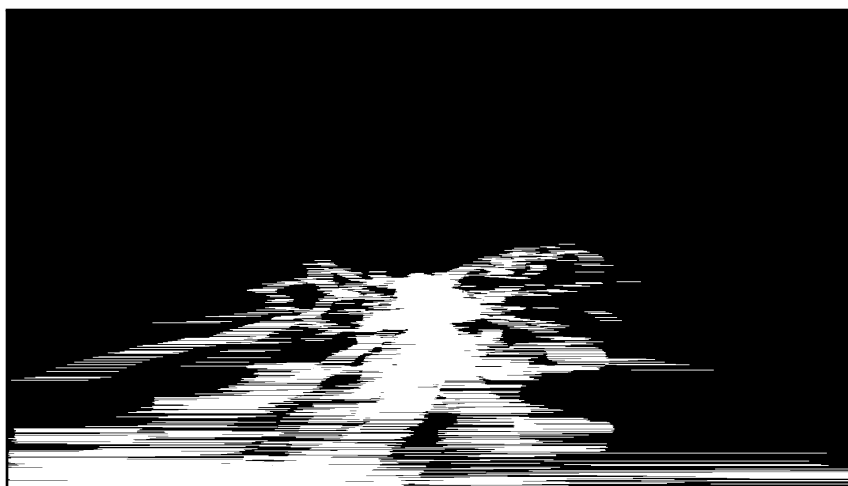
- In een implementatie waarbij er meerdere fietsers moeten gevolgd worden, zorgt dit voor een veelvoud aan CPU-belasting. Dit laatste kan een probleem vormen bij apparaten met een beperkte rekenkracht.
- Aangezien de objectdetector de grootste bottleneck vormt qua performantievereisten, kan eventueel de objecttracker meer ingezet worden om de bounding box van een fietser in een frame te schatten en te volgen. Om onafhankelijk te zijn van het verschil in stuurhoogte en de hoogte van de camera kan er ook een marker op de voorkant van elk stuur geplaatst worden om deze eenvoudig, maar robuust te detecteren. Deze marker moet bijgevolg voor alle fietsen op dezelfde hoogte worden aangebracht.
- Indien de tilthoek relatief goed geschat wordt, kan er tevens een selectie uitgevoerd worden van pixels in het beeld waar zich een fiets kan bevinden. Met andere woorden, er zijn regio's in het beeld waar er nooit een fietser zal worden gedetecteerd. Idealiter worden er alleen fietsers gedetecteerd in de blauwe RIO van Figuur 78, de onderkant van de bounding box kan alleen op het wegdek liggen. In



Figuur 78: Ideaal detectiemasker

de praktijk komt dit na testing overeen met Figuur 79 voor de onderkant van de bounding box en Figuur 80 voor de gehele bounding box.

In de huidige implementatie van het draftingdetectiesysteem is gebruikgemaakt van Tiny YOLOv3, maar SSD met het MobileNET *framework* biedt ook een veelbelovende mogelijkheid voor gebruik in ingebelde systemen.



Figuur 79: Detectiemasker opgebouwd op basis van de onderkant van de gedetecteerde bounding boxes uit de testset



Figuur 80: Detectiemasker opgebouwd op basis van de gedetecteerde bounding boxes uit de testset

Conclusie

In deze thesis is een proof of concept voor een draftingdetectiesysteem in triatlon uiteengezet. Het draftingconcept en de verschillende situaties tijdens een triatlonwedstrijd zijn in een eerste deel behandeld. Met deze kennis is het systeem ontwikkeld dat is samengesteld uit drie belangrijke bouwstenen: objectdetectie, objecttracking en afstandsbepaling.

Het detecteren en aansluitend volgen van de dichtstbijzijnde fietser doorheen de verschillende scènes, zorgt voor een continue monitoring in de tijd. Op die manier wordt voor de testset een gemiddelde precisie, AP, bekomen van 88,27 % voor fietsdetectie. De nauwkeurigheid van de bounding box van de gedetecteerde fietser is immers van belang voor de afstandsbepaler. Voor dit laatstgenoemde zijn er twee methoden behandeld die aansluitend ten opzichte van elkaar zijn afgewogen: de hoogte- en grondvlakmethode. De hoogtemethode blijkt uit de resultaten het meest nauwkeurig, daar de grondvlakmethode te gevoelig is voor een tilthoekverandering en -schatting. In een statische situatietest bedraagt de gemiddelde absolute fout 0,89 meter en 1,46 meter voor respectievelijk de hoogtemethode en de grondvlakmethode. In dit geval is de tilthoek, waarvan de grondvlakmethode sterk afhankelijk is, echter ideaal verondersteld (wat in de praktijk uitgesloten is). Om een meer reële wedstrijdssituatie te benaderen, is er een dynamische test uitgevoerd met LIDAR-afstanden als grondwaarheid. Uit dit experiment blijkt dat de hoogtemethode een gemiddelde absolute fout heeft van 0,47 meter. Voor de grondvlakmethode bedraagt deze fout 1,16 meter in het geval van een perfecte tilthoek en 1,96 meter in het overige geval. De hoogtemethode blijkt bijgevolg een verantwoorde keuze.

In vergelijking met een eerder ontwerp uit de literatuur op basis van GPS en mobiele netwerken waar de fout of de relatief gemeten afstand 1,32 meter tot 2,64 meter bedraagt, is er met andere woorden een verbetering tot 82,20 % bekomen. Daarnaast is locatiebepaling aan de hand van GPS erg plaats afhankelijk waardoor dit erg betrouwbaar is in de open ruimte, maar veel minder in een stedelijke omgeving. Er is bovendien een draftingdetectiescore opgesteld om de kans, bekeken over 30 seconden, te bepalen dat de draftingregel is overtreden bij een afstandsberekening om de 2 seconden. Deze regel is geschonden indien een atleet zich langer dan 20 seconden in de draftingzone van 10 meter op 3 meter begaf. In het kader van een eindproduct voor dit concept zijn in het laatste hoofdstuk van deze scriptie enkele uitbreidingsmogelijkheden en benaderingen besproken met oog op apparaten met beperkte rekenkracht. Het draftingdetectiesysteem met een gemiddelde absolute afstandsfout van 0,47 meter kan beslist intrede doen in triatlon- en duatlonwedstrijden om bijgevolg een eerlijker wedstrijdverloop te verkrijgen.

Referentielijst

- [1] B. T. & D. Federatie, “BTDF Sportreglementen 2018”, 2018.
- [2] B. Mavis, “Implementing Challenge’s 20-Meter Drafting Rule – Triathlete”, 2015. [Online]. Beschikbaar: https://www.triathlete.com/2015/03/race-coverage/implementing-challenges-20-meter-drafting-rule_113485
- [3] Eurosport. (2019) Gent-Wevelgem 2019 Highlights - Cycling. [Online]. Beschikbaar: <https://www.youtube.com/watch?v=chLNhU2-XrY>
- [4] B. Blocken, T. V. Druenen, Y. Toparlar, F. Malizia, P. Mannion, T. Andrianne, T. Marchal, G.-j. Maas, en J. Diepens, “Journal of Wind Engineering & Industrial Aerodynamics Aerodynamic drag in cycling pelotons : New insights by CFD simulation and wind tunnel testing”, *Journal of Wind Engineering & Industrial Aerodynamics*, vol. 179, no. May, pp. 319–337, 2018. [Online]. Beschikbaar: <https://doi.org/10.1016/j.jweia.2018.06.011>
- [5] I. Fister, D. Fister, S. Fong, en I. Fister, “Widespread mobile devices in applications for real-time drafting detection in triathlons”, *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 3, pp. 310–321, 2013.
- [6] I. Fister en I. J. Fister, “Concept of drafting detection system in Ironmans”, *Electrotechnical Review*, vol. 78, no. 4, pp. 217–222, 2011.
- [7] P. Ramjee en R. Marina, *Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems [Hardcover]*. Artech House, 2005. [Online]. Beschikbaar: <https://lib.ugent.be/catalog/rug01:001050355>
- [8] Redactie Garmin, “Varia™ Achteruitkijk Fietsradar”. [Online]. Beschikbaar: <https://buy.garmin.com/nl-BE/BE/p/518151/pn/010-01509-10>
- [9] T. Solc, T. Gale, en C. Fortuna, “Optimization of ultra-narrowband wireless communication: An experimental case study”, in *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2017*. IEEE, may 2017, pp. 523–528. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/8116432/>
- [10] H. Esmaili en H. Thwaites, “Virtual photogrammetry”, in *Proceedings of the 2016 International Conference on Virtual Systems and Multimedia, VSMM 2016*. IEEE, oct 2016, pp. 1–6. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/7863153/>

- [11] T. Tietz, “Édouard-Gaston Deville and the Perfection of Photogrammetry - SciHi BlogSciHi Blog”, 2017. [Online]. Beschikbaar:
<http://scihi.org/edouard-gaston-deville-photogrammetry/>
- [12] A. Z. Richard Hartley, *Multiple View Geometry*, 2003, vol. 53, no. 9.
- [13] Z. Zhang, “A Flexible New Technique for Camera Calibration (Technical Report)”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.
- [14] K. Simek, “The Intrinsic Matrix”, 2013. [Online]. Beschikbaar:
<http://ksimek.github.io/2013/08/13/intrinsic/>
- [15] Z. Zhang, “Camera calibration with one-dimensional objects”, vol. 26, no. 7, pp. 892–899, 2004.
- [16] R. Y. Tsai, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, aug 1987. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/1087109/>
- [17] G. P. Stein, “Accurate Internal Camera Calibration using Rotation, with Analysis of Sources of Error”, in *Fifth International Conference on Computer Vision (Iccv'95)*, pp. 230 – 236, 1995. [Online]. Beschikbaar:
<http://ieeexplore.ieee.org/document/466781/>
- [18] J.-O. Eklundh en R. I., “Computer Vision - ECCV '94”, in *Proceedings of the third European conference on Computer vision (vol. 1)*, vol. 800. Springer-Verlag, 1994, pp. 471–478. [Online]. Beschikbaar: <https://dl.acm.org/citation.cfm?id=189583>
- [19] D. Liebowitz en A. Zisserman, “Metric rectification for perspective images of planes”, *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 482–488, 1998. [Online]. Beschikbaar:
<http://ieeexplore.ieee.org/document/698649/>
- [20] B. Caprile en V. Torre, “Using vanishing points for camera calibration”, *International Journal of Computer Vision*, vol. 4, no. 2, pp. 127–139, mar 1990. [Online]. Beschikbaar: <http://link.springer.com/10.1007/BF00127813>
- [21] M. Engineering, “Positioning of Drug Carriers using Permanent Magnet-Based Robotic System Master Thesis of cand . M . Sc . Abdelrahman Hassan Elfar Application number 22-0694”, 2017.

- [22] G. E. T. S. For, “HERO5 Black Field of View (FOV) Information”. [Online]. Beschikbaar: https://gopro.com/help/articles/question_answer/hero6-black-field-of-view-fov-information
- [23] S. S. Grigsby, “Augmented Cognition: Intelligent Technologies”, vol. 10915, no. June, 2018. [Online]. Beschikbaar: <http://link.springer.com/10.1007/978-3-319-91470-1>
- [24] S. Nagaraj, B. Muthiyar, S. Ravi, V. Menezes, K. Kapoor, en H. Jeon, “Edge-Based Street Object Detection”, *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)*, pp. 1–4, 2017.
- [25] R. D. Hof, “Deep Learning - MIT Technology Review”, 2013. [Online]. Beschikbaar: <https://www.technologyreview.com/s/513696/deep-learning/>
- [26] Y. LeCun, “LeNet-5, convolutional neural networks”, Tech. Rep., 2015. [Online]. Beschikbaar: <http://yann.lecun.com/exdb/lenet/>
- [27] C. Papageorgiou, M. Oren, en T. Poggio, “A general framework for object detection”, *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, no. February 1998, pp. 555–562, 2014. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/710772/>
- [28] D. Lowe, “Object recognition from local scale-invariant features”, in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999, pp. 1150–1157 vol.2. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/790410/>
- [29] Grace Tsai, “Histogram of oriented gradients”, pp. 1–17, 2010. [Online]. Beschikbaar: <https://pdfs.semanticscholar.org/presentation/342b/c8b55d46f822e1574e4c6fcaa0b8bfa5d3b.pdf>
- [30] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, en T. Darrell, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition”, 2013. [Online]. Beschikbaar: <http://decaf.berkeleyvision.org/http://arxiv.org/abs/1310.1531>
- [31] A. Kumar, “Object Localization & Object Detection”, 2017. [Online]. Beschikbaar: <http://www.ashukumar27.io/Object-Localization/>

- [32] J. Redmon, S. Divvala, R. Girshick, en A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, 2015. [Online]. Beschikbaar: <http://arxiv.org/abs/1506.02640>
- [33] K. E. A van de Sande, J. R. R Uijlings, T. Gevers, en A. W. M Smeulders, “Segmentation as Selective Search for Object Recognition”, Tech. Rep. [Online]. Beschikbaar: <https://www.koen.me/research/pub/vandesande-iccv2011.pdf>
- [34] S. Ren, K. He, R. Girshick, J. Sun, J. Donahue, T. Darrell, J. Malik, W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, R. Girshick, D. Impiombato, S. Giarrusso, T. Mineo, O. Catalano, C. Gargano, G. La Rosa, F. Russo, G. Sottile, S. Billotta, G. Bonanno, S. Garozzo, A. Grillo, D. Marano, G. Romeo, D. Erhan, C. Szegedy, A. Toshev, en D. Anguelov, “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 794, pp. 1–15, 2015. [Online]. Beschikbaar: <https://arxiv.org/abs/1311.2524>
- [35] R. Girshick, J. Donahue, T. Darrell, J. Malik, en U. C. Berkeley, “Rich feature hierarchies for accurate object detection and semantic segmentation”, 2012.
- [36] R. P. Networks, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, vol. 39, no. 1096, pp. 1137–1149, 2015. [Online]. Beschikbaar: <https://arxiv.org/abs/1506.01497>
- [37] K. He, G. Gkioxari, P. Dollar, en R. Girshick, “Mask R-CNN”, *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 2980–2988, 2017.
- [38] A. Geiger, P. Lenz, C. Stiller, en R. Urtasun, “The KITTI Vision Benchmark Suite”, 2013. [Online]. Beschikbaar: <http://www.cvlibs.net/datasets/kitti/>
- [39] R. Mehta en C. Ozturk, “Object detection at 200 Frames Per Second”, 2018. [Online]. Beschikbaar: <http://arxiv.org/abs/1805.06361>
- [40] J. Redmon en A. Farhadi, “YOLOv3: An Incremental Improvement”, 2018. [Online]. Beschikbaar: <http://arxiv.org/abs/1804.02767>
- [41] D. Impiombato, S. Giarrusso, T. Mineo, O. Catalano, C. Gargano, G. La Rosa, F. Russo, G. Sottile, S. Billotta, G. Bonanno, S. Garozzo, A. Grillo, D. Marano, en G. Romeo, “SSD: Single Shot MultiBox Detector Wei”, *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 794, pp. 185–192, 2015. [Online]. Beschikbaar: <http://arxiv.org/abs/1512.02325>

- [42] L. Van Gool, M. Everingham, A. Zisserman, J. Winn, en C. K. I. Williams, “The Pascal Visual Object Classes (VOC) Challenge”, *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2009.
- [43] A. Briot, P. Viswanath, en S. Yogamani, “Analysis of efficient CNN design techniques for semantic segmentation”, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, 2018, pp. 776–785. [Online]. Beschikbaar:
http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w12/Briot_Analysis_of_Efficient_CVPR_2018_paper.pdf
- [44] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, en H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, 2017. [Online]. Beschikbaar:
<https://arxiv.org/pdf/1704.04861.pdf>
- [45] D. POWERS, “Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation”, *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Beschikbaar:
<http://david.wardpowers.info/BM/index.htm>.
- [46] T. Fawcett, “An introduction to ROC analysis”, *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Beschikbaar:
www.elsevier.com/locate/patrec
- [47] P. Dollár, C. Wojek, B. Schiele, en P. Perona, “Pedestrian Detection: The State of the Art”, *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012. [Online]. Beschikbaar: www.vision.caltech.edu/Image
- [48] Redactie Bicycled, “Team Wanty – Groupe Gobert naar de Tour met nieuwe Cube Aerium TT C:68 — Bicycled Magazine”, 2018. [Online]. Beschikbaar:
<https://www.bicycled.be/magazine/team-wanty-groupe-gobert-naar-de-tour-met-nieuwe-cube-aerium-tt-c68/>
- [49] Redactie Tweewieler, “Merida pakt op Bike MOTION uit met limited edition Reacto Disc YC”, 2019. [Online]. Beschikbaar:
https://www.tweewieler.nl/winkel/nieuws/2019/02/merida-pakt-op-bike-motion-uit-met-limited-edition-reacto-disc-yc-10137626?vakmedianet-approve-cookies=1&_ga=2.122429451.10414806.1555007469-752785019.1555007469

- [50] Seung-Hwan Bae en Kuk-Jin Yoon, “Robust Online Multiobject Tracking With Data Association and Track Management”, *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2820–2833, jul 2014. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/6807759/>
- [51] S. Roth, “People-Tracking-by-Detection and People-Detection-by-Tracking”. [Online]. Beschikbaar: https://www.mpi-inf.mpg.de/fileadmin/inf/d2/andriluka/andriluka_cvpr08.pdf
- [52] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, en S. G. Kong, “Intelligent visual surveillance - A survey”, *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 926–939, 2010.
- [53] S. M. Khan en M. Shah, “Tracking multiple occluding people by localizing on multiple scene planes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, mar 2009. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/4497204/>
- [54] M. Yazdi en T. Bouwmans, “New trends on moving object detection in video images captured by a moving camera: A survey”, pp. 157–177, may 2018. [Online]. Beschikbaar: <https://www.sciencedirect.com/science/article/pii/S1574013716301794>
- [55] A. Yilmaz, O. Javed, en M. Shah, “Object tracking”, *ACM Computing Surveys*, vol. 38, no. 4, pp. 13–es, 2006. [Online]. Beschikbaar: <http://portal.acm.org/citation.cfm?doid=1177352.1177355>
- [56] Y. Xiang, A. Alahi, en S. Savarese, “Learning to Track : Online Multi- Object Tracking by Decision Making Multi-Object Tracking”, 2015.
- [57] Canny J., “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [58] M. Lucena, J. Fuertes, J. Gomez, N. Perez de la Blanca, en A. Garrido, “Tracking from optical flow”, in *3rd International Symposium on Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the*, vol. 2. IEEE, 2004, pp. 651–655. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/1296357/>
- [59] A. Keivani, J.-R. Tapamo, en F. Ghayoor, “Motion-based moving object detection and tracking using automatic K-means”, *Africon, 2017 Ieee*, pp. 32–37, sep 2017. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/8095451/>

- [60] W. C. Hu, C. H. Chen, T. Y. Chen, D. Y. Huang, en Z. C. Wu, “Moving object detection and tracking from video captured by moving camera”, *Journal of Visual Communication and Image Representation*, vol. 30, pp. 164–180, 2015. [Online]. Beschikbaar: <http://dx.doi.org/10.1016/j.jvcir.2015.03.003>
- [61] Y. M. Song, S. Noh, J. Yu, C. W. Park, en B. G. Lee, “Background subtraction based on Gaussian mixture models using color and depth information”, *2014 International Conference on Control, Automation and Information Sciences, ICCAIS 2014*, no. Iccais, pp. 132–135, 2014.
- [62] N. Shah, A. Pingale, V. Patel, en N. V. George, “An adaptive background subtraction scheme for video surveillance systems”, *2017 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2017*, pp. 13–17, 2018.
- [63] T. Surgailis, A. Valinevicius, en M. Zilyys, “Traffic image processing systems”, *2nd International Conference on Advances in Circuits, Electronics and Micro-electronics - CENICS 2009*, pp. 61–66, 2009.
- [64] C. Peter, A. IEEE, N. O. Machine, en V. O. L. Intelligence, “Dorin Mean Shift: Approach Toward Feature Space Analysis.” *on Pattern Analysis and 5MAY*, vol. 24, no. 5, pp. 603–619, 2002.
- [65] S. Huang en J. Hong, “Moving object tracking system based on camshift and Kalman filter”, in *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*. IEEE, apr 2011, pp. 1423–1426. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/5769081/>
- [66] G. Swalaganata, D. R. Sulistyaningrum, en B. Setiyono, “Super-resolution imaging applied to moving object tracking”, in *Journal of Physics: Conference Series*, vol. 893, no. 1, 2017.
- [67] Robotix, “Kalman Filter”, 2018. [Online]. Beschikbaar: https://2018.robotix.in/tutorial/advanced/kalman_filter/
- [68] G. Swalaganata, “Moving Object Tracking Using Hybrid Method”, pp. 606–610, 2018.
- [69] D. Ding, Z. Jiang, en C. Liu, “Object tracking algorithm based on particle filter with color and texture feature”, in *Chinese Control Conference, CCC*, vol. 2016-Augus, no. 1. TCCT, 2016, pp. 4031–4036.

- [70] D. Held, S. Thrun, en S. Savarese, “Learning to Track at 100 FPS with Deep Regression Networks”, Tech. Rep. [Online]. Beschikbaar: <http://davheld.github.io/GOTURN/GOTURN.html>
- [71] B. Babenko, S. Belongie, en M. H. Yang, “Visual tracking with online multiple instance learning”, in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, vol. 2009 IEEE. IEEE, jun 2009, pp. 983–990. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/5206737/>
- [72] M. Danelljan, G. Hager, F. S. Khan, en M. Felsberg, “Discriminative Scale Space Tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017. [Online]. Beschikbaar: <https://arxiv.org/pdf/1609.06141.pdf>
- [73] A. Kaehler en G. Bradski, *Learning OpenCV - computer vision in c++ with the opencv Library*. O’Reilly, 2013, vol. 53, no. 9.
- [74] D. E. King, “Dlibml: A Machine Learning Toolkit”, Tech. Rep., 2009. [Online]. Beschikbaar: <http://www.jmlr.org/papers/volume10/king09a/king09a.pdf>
- [75] A. Lukežič, T. Vojř, L. Čehovin Zajc, J. Matas, en M. Kristan, “Discriminative Correlation Filter Tracker with Channel and Spatial Reliability”, *International Journal of Computer Vision*, vol. 126, no. 7, pp. 671–688, 2018.
- [76] João F. Henriques, Caseiro Rui, Martins Pedro, en Batista Horge, “High-Speed Tracking with Kernelized Correlation Filters”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. [Online]. Beschikbaar: http://www.robots.ox.ac.uk/~joao/publications/henriques_tpami2015.pdf
- [77] R. E. Schapire, “Explaining AdaBoost”. [Online]. Beschikbaar: <http://rob.schapire.net/papers/explaining-adaboost.pdf>
- [78] Z. Kalal, K. Mikolajczyk, en J. Matas, “Tracking-learning-detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, jul 2012. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/6104061/>
- [79] —, “Forward-backward error: Automatic detection of tracking failures”, in *Proceedings - International Conference on Pattern Recognition*. IEEE, aug 2010, pp. 2756–2759. [Online]. Beschikbaar: <http://ieeexplore.ieee.org/document/5596017/>

- [80] Y. Lu, Z. Zhou, en J. Zhao, “Visual Object Tracking Using PCA Correlation Filters”, 2018. [Online]. Beschikbaar: http://www.cs.colostate.edu/~vision/publications/bolme_cvpr10.pdf
- [81] Y. Wu, J. Lim, en M. H. Yang, “Online object tracking: A benchmark”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418. [Online]. Beschikbaar: https://www.cv-foundation.org/openaccess/content_cvpr_2013/papers/Wu_Online_Object_Tracking_2013_CVPR_paper.pdf
- [82] J. Su, D. V. Vargas, en S. Kouichi, “One pixel attack for fooling deep neural networks”, 2017. [Online]. Beschikbaar: <https://github.com/Carina02/One-Pixel-Attack><http://arxiv.org/abs/1710.08864>
- [83] J. Wang, L. Yang, A. R. Rezaie, J. Li, en S. Louis, “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, no. April, 2011. [Online]. Beschikbaar: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [84] R&A Cycles, “2018 Argon 18 E-117 Tri Ultegra Bike”. [Online]. Beschikbaar: <https://www.racycles.com/product/detail/14750>
- [85] R. ergonomiesite, “Fietsmaat - Ergonomie site”, 2017. [Online]. Beschikbaar: <http://www.ergonomiesite.be/fietsmaat/>
- [86] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, en J. Míguez, “Particle filtering”, *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003. [Online]. Beschikbaar: <https://arxiv.org/pdf/1309.7807.pdf>
- [87] X. Xu, B. Huang, en S. Dubljevic, “Optimal state estimation for linear systems with state constraints”, in *IFAC-PapersOnLine*, vol. 28, no. 8, 2015, pp. 153–157. [Online]. Beschikbaar: <https://linkinghub.elsevier.com/retrieve/pii/S240589631501040X>
- [88] D. V. Hamme, W. Goeman, P. Velaert, en W. Philips, *Robust monocular visual odometry for road vehicles using uncertain perspective projection*, 2015, vol. 2015, no. 1.
- [89] Tridavid, “Triatlon 2017 - Fietsroute”. [Online]. Beschikbaar: <https://www.routeyou.com/nl-be/route/view/4624306/fietsroute/triatlon-2017-versie-23-02>
- [90] D. Velasco-Montero, J. Fernández-Berni, R. Carmona-Galán, en Á. Rodríguez-Vázquez, “Optimum Network/Framework Selection from High-Level Specifications in Embedded Deep Learning Vision Applications”, *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11182 LNCS, pp. 369–379, 2018.

- [91] Redactie CamDo, “GoPro Battery Consumption Comparison”. [Online]. Beschikbaar: <https://cam-do.com/pages/power-consumption-by-gopro-camera-model>
- [92] Ximea, “Compact cameras with PCIe interface for embedded applications and deep integrations”, Tech. Rep. [Online]. Beschikbaar: <https://www.ximea.com/files/brochures/xiX-OEM-cameras-for-integration-2017-brochure-HQ.pdf>
- [93] Intel® Software, “Deploy real-time object detectors on Intel® Movidius™ Neural Compute Stick”. [Online]. Beschikbaar: <https://www.slideshare.net/IntelSoftware/deploy-realtime-object-detectors-on-intel-movidius-neural-compute-stick>
- [94] Nvidia, “Jetson Nano Brings AI Computing to Everyone — NVIDIA Developer Blog”. [Online]. Beschikbaar: <https://devblogs.nvidia.com/jetson-nano-ai-computing/>

Bijlagen

Bijlage A: Enquête

Affiche Enquête

FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Computervisiegebaseerde detectie van drafting tijdens het fietsen

```
graph TD; Q1["I ♥ triathlon"] -- Yes --> R1; Q1 -- No --> Q2["I ♥ duathlon"]; Q2 -- Yes --> R1; Q2 -- No --> Q3["I ♥ non-drafting competitions"]; Q3 -- Yes --> R1; Q3 -- No --> Q4["when they draft in a non-drafting competition"]; Q4 -- Yes --> R1; Q4 -- No --> Q5["Would you like to help with an investigation?"]; Q5 -- Yes --> R1; Q5 -- No --> Q6["Keep on tripping!"]; R1["I need you!"]
```

Heb jij ook een afkeer van het draften in een *non-drafting* triathlon wedstrijd?

Vul dan de enquête in horende bij dit bericht!

<https://goo.gl/Gnyy4k>

Simon Van den Bossche

1e Master Industriële Wetenschappen
Elektronica-ICT – ICT beeldtechnologie

✉ sidnboss.vandenbossche@ugent.be

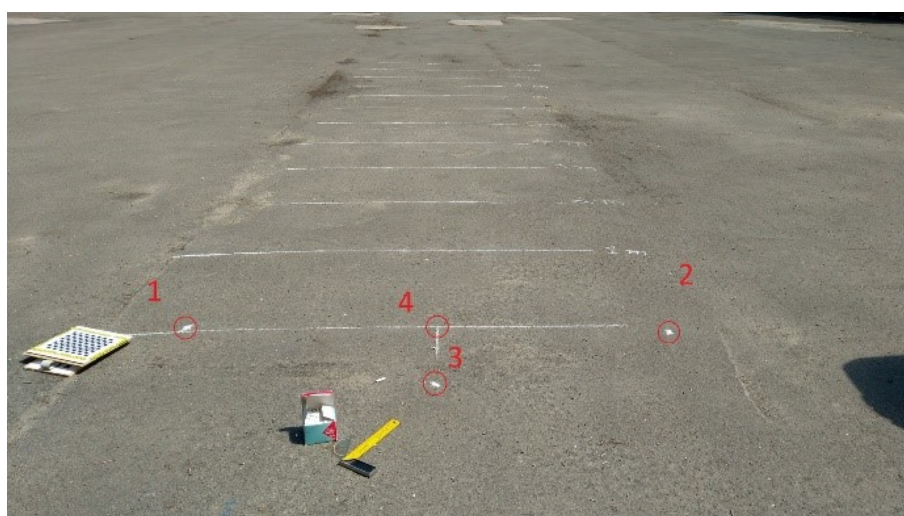
UNIVERSITEIT
GENT

Figuur 81: Affiche van de uitgevoerde enquête

Bijlage B: Fietsafmetingen bij wedstrijdopnames



Figuur 82: Afstand tot en met 10 meter

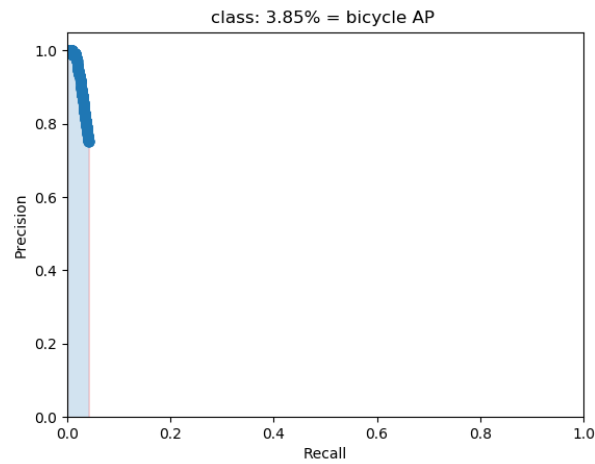


Figuur 83: Opgemeten afstanden

Fietstype	Camera-hoogte (cm)		Grondvlak (cm)				Fietsafmeting (cm)		
	Camera-as	Onderkant camera	1-4	4-2	3-4	1-3	3-2	Wielstraal	Lengte
Camerafiets (trainingsritten)	97,50	92,50	-	-	47,50	-	-	32,10	166,50
Draftende fiets (trainingsritten)	-	-	-	-	-	-	-	32,10	164,50
Camerafiets (Swimkap)	92,00	87,00	117,83	104,67	50,00	128,00	116,00	34,25	168,00
Camerafiets (Triathlon Deinze)	88,50	82,50	80,00	114,50	53,00	184,00	170,00	-	-
Camerafiets (Zwintriathlon)	95,00	88,30	153,00	92,00	49,00	171,00	121,00	-	-
Camerafiets (Den Halven van Damme)	99,50	94,00	123,5	73,00	50,00	145,00	104,00	34,00	171,00

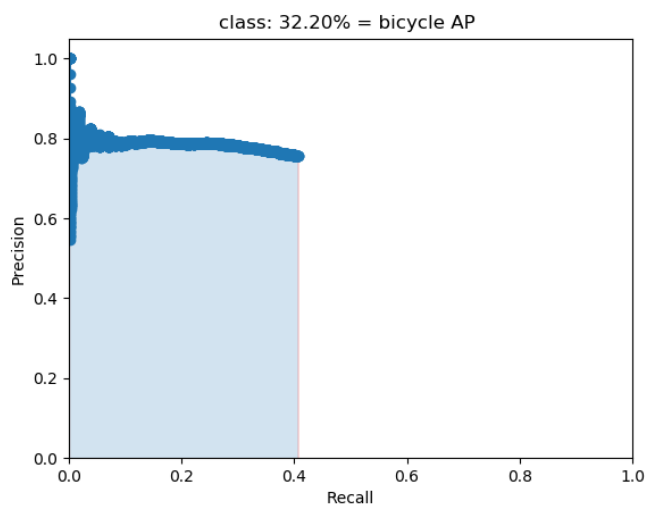
Bijlage C: Precision recall curves basis fietsdetector-netwerken

SSD



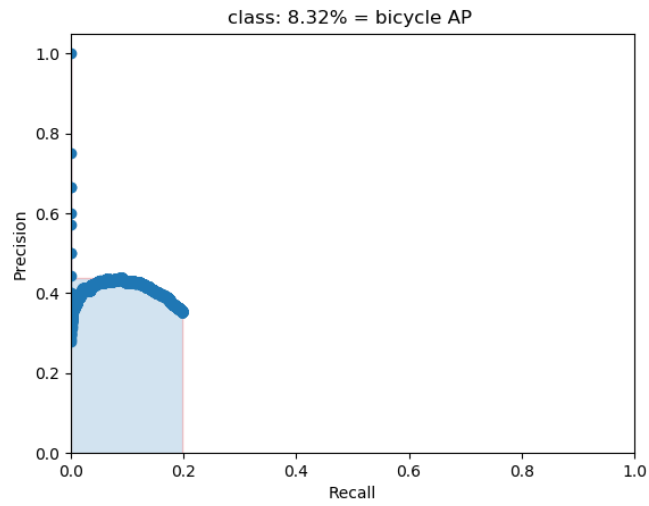
Figuur 84: Precision recall SSD

Faster RCNN



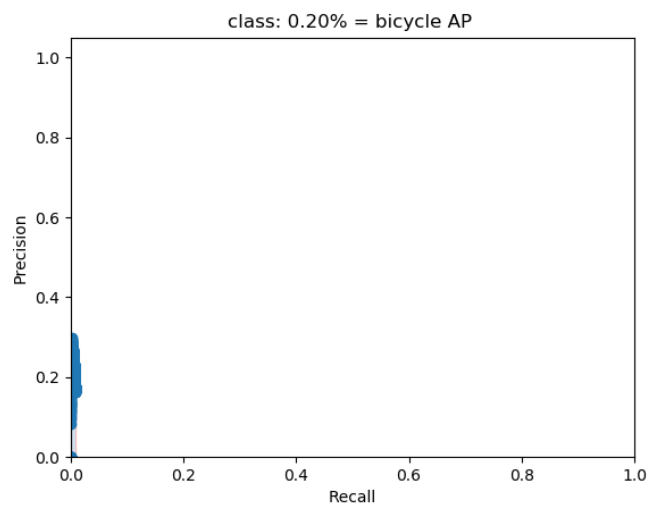
Figuur 85: Precision recall Faster RCNN

Mask RCNN



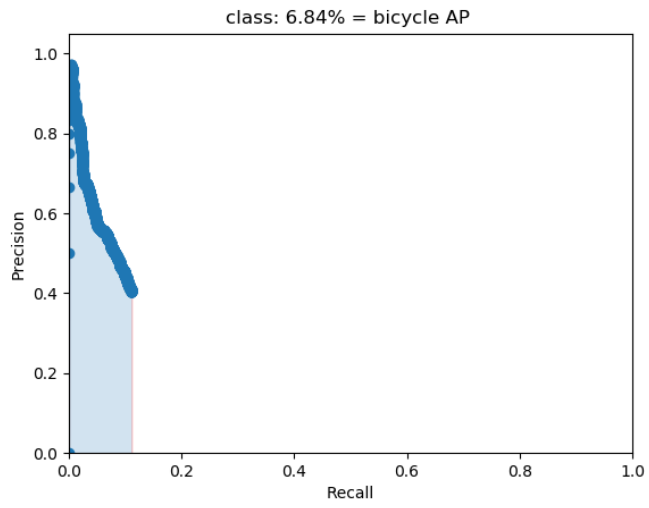
Figuur 86: Precision recall Mask RCNN

YOLOv3-tiny basic



Figuur 87: Precision recall YOLOv3-tiny basic

YOLOv3-416 basic



Figuur 88: Precision recall YOLOv3-416 basic

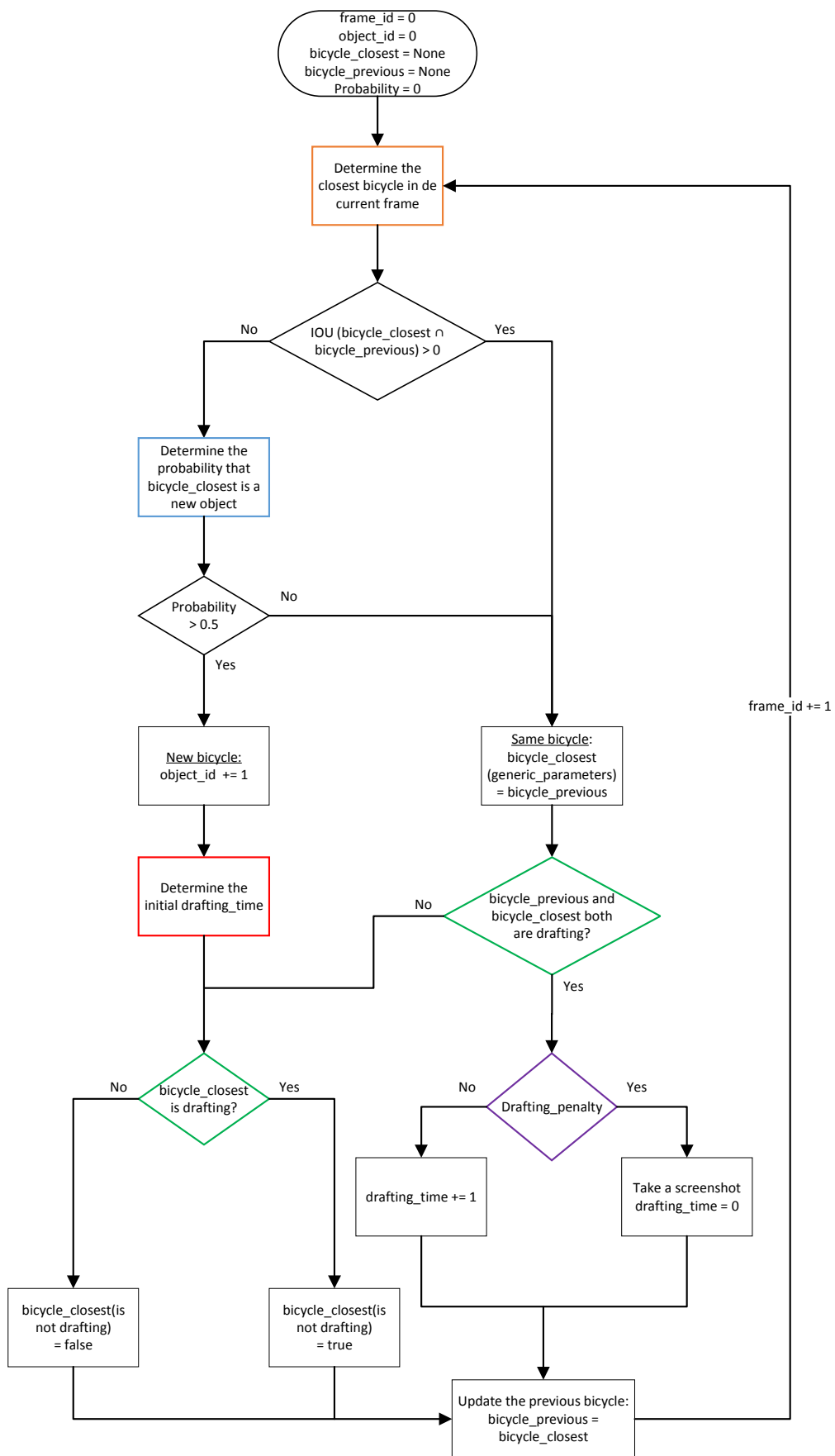
Bijlage D: Flowcharts voor het algoritme van drafting-detectie

De flowchart van het algoritme van het draftingdetectiesysteem is hieronder weergegeven waarbij de kleuren die daarop volgen.

Hoofdapplicatie

De flowchart voor de hoofdapplicatie is weergegeven in Figuur 89, waarin er twee soorten fietsen worden beschouwd `bicycle_closest` en `bicycle_previous`, beiden geïnitieerd op `None`. Als de objectdetector (meer specifiek de objectdetector en de objecttracker) een fiets heeft gevonden in het huidige frame (`frame_id`), wordt er gekeken dat het gaat om een fiets die al in een eerder frame is gedetecteerd of het een nieuwe fiets is (bijvoorbeeld een voorbijsteker die op dat moment de dichtste fietser is). Als de bounding box van de fiets uit het vorige frame overlapt met deze uit het huidige frame, wordt er vanuit gegaan dat dit dezelfde fiets is ('Same bicycle'). Nadien wordt er gecontroleerd de fietser in beide gevallen aan het draften was en wordt er een draftingpenalty toegekend indien `Drafting_penalty` 'Yes' geeft. Dientengevolge wordt er een screenshot genomen en wordt de draftingtijd (`drafting_time`) voor deze fiets op 0 geplaatst. Als het beslissingsblok `Drafting_penalty` echter 'No' geeft, wordt de draftingtijd voor deze fietser geïncrmenteerd. Nadien wordt voor beide gevallen `bicycle_previous` geüpdatet en klaar gezet voor een volgend frame (`frame_id += 1`). Indien de bounding boxes echter niet overlappen niet overlappen, wordt de kans bepaald of het effectief om een nieuwe fiets gaat. De fiets `bicycle_closest` kan namelijk een valse detectie zijn zoals een omstaander, niet de meest dichtste fiets, ... Als deze kans kleiner is dan 50 % wordt verondersteld dat het om een valse detectie gaat en wordt de fiets in het huidige frame (`bicycle_closest`) geïnitieerd met de parameters van de fiets uit het vorige frame (`bicycle_previous`). Indien deze kans echter groter is dan 50 % gaat het om een nieuwe fiets ('New bicycle'). In dat geval wordt een corresponderend `object_id` toegewezen aan deze fiets met name `bicycle_closest`. Alsook wordt het begin (`drafting_time`) geïnitieerd. Dit is nodig omdat iemand die voorbijgestoken wordt, zelf 5 seconden krijgt om zich uit de draftingzone te bewegen.

Er wordt nadien gekeken of deze fiets in het huidige frame aan het draften is. Deze controle wordt ook uitgevoerd indien `bicycle_closest` en `bicycle_previous` beide niet in de draftingzone zitten indien de IOU groter is dan 0. Op die manier kan dus per frame bepaald worden of een fietser aan het draften is. Nadien wordt ook in dit geval de `bicycle_previous` geüpdatet en klaar gezet voor een volgend frame (`frame_id += 1`).



Figuur 89: Flowchart hoofdapplicatie

Dichtste fiets bepaling

De flowchart voor het bepalen van de dichtste fiets is weergegeven in Figuur 90. Om deze fiets te bepalen wordt er een combinatie uitgevoerd van objectdetectie door de objectdetector uit Hoofdstuk 5 en *tracking-by-detection* besproken in Hoofdstuk 6. Kortom wordt de objecttracker ingeschakeld indien er geen fietser is gedetecteerd in het huidige frame (`bicycle_closest`), maar wel in het vorige frame (`bicycle_previous`). Eerst worden er fietsers gedetecteerd in het huidige frame aan de hand van de objectdetector. Als deze één of meer fietsers heeft gedetecteerd, wordt de dichtste fietser bepaald: `bicycle.d`. Indien dit niet het geval is wordt er gecheckt of de meegegeven `bicycle_previous` bestaat (niet `None` is). Indien dit niet het geval is, wordt de `tracking_time` op 0 geplaatst en `bicycle_closest` op `None`. Als er in het vorige frame wel degelijk een fiets is gedetecteerd, wordt deze gebruikt om deze in het huidige frame te schatten aan de hand van de objecttracker: `bicycle.t`. Indien deze bestaat en de tracker niet langer dan 3 seconden in werking is gesteld, wordt deze toegewezen aan `bicycle_closest` en wordt de `tracking_time` geïncrementeerd.

Draftingcontrole

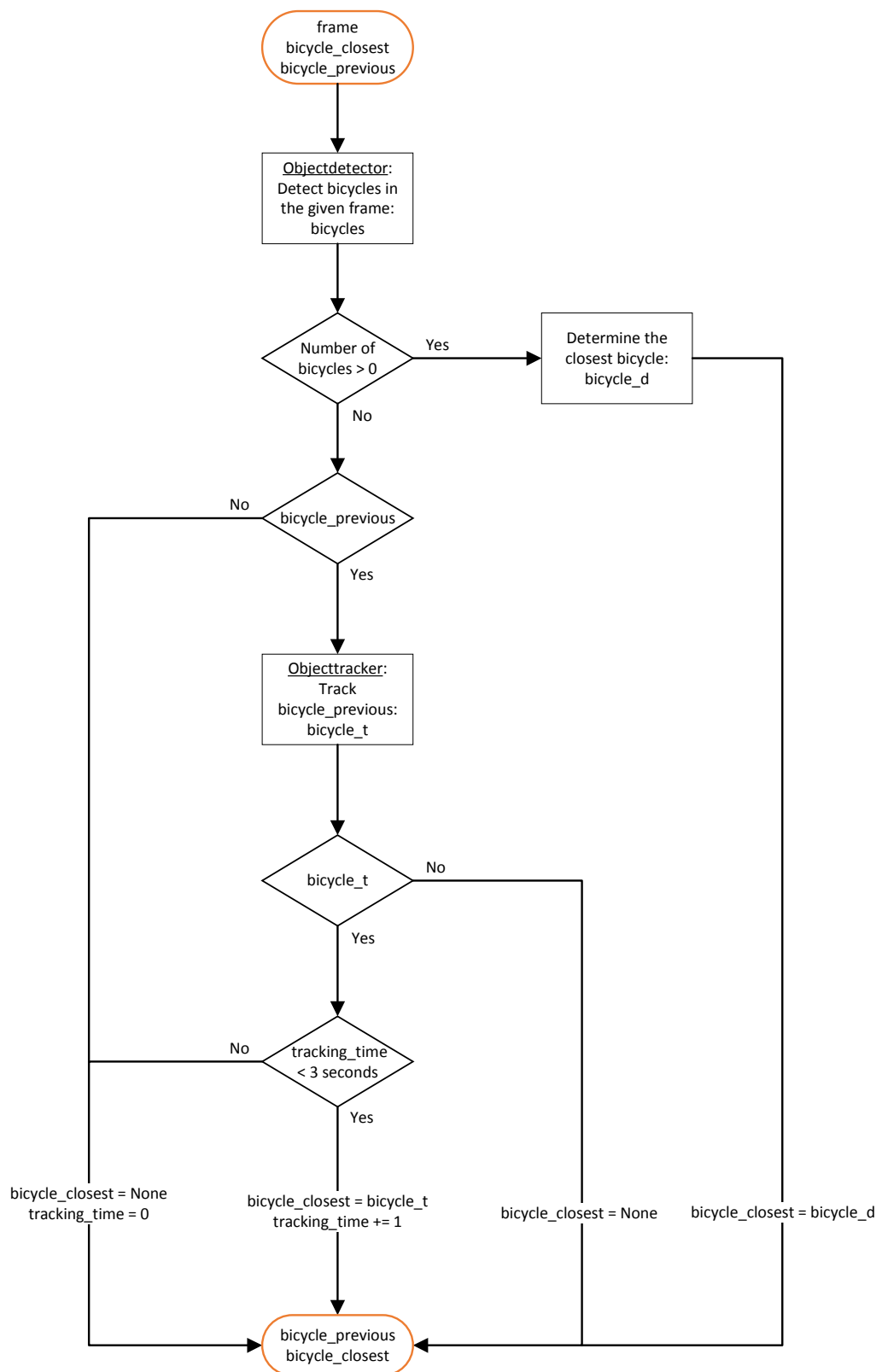
De flowchart voor de draftingcontrole wordt geïllustreerd in Figuur 91. Om te controleren of een fiets aan het draften is, wordt eerst en vooral gecontroleerd of zijn overeenkomstige *boolean* op `true` staat. Als dit het geval is, wordt er ‘Yes’ teruggegeven. In het andere geval wordt de afstand bepaald en vervolgens gefilterd. Indien deze afstand kleiner is of gelijk aan de draftingafstand marge (bijvoorbeeld 10 meter), dan wordt ‘No’ teruggegeven en in het andere geval ‘Yes’.

Nieuwe fiets bepaling

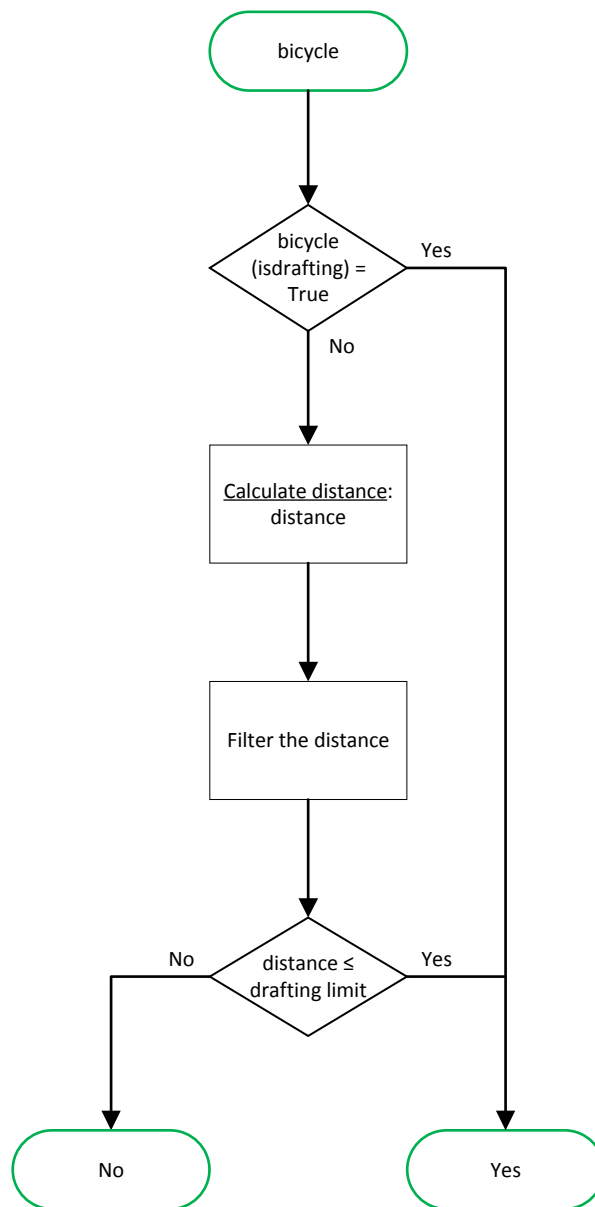
Om te bepalen of er een nieuwe fiets in het beeld optreedt wordt er vijf keer (de waarde kan anders ingesteld worden maar een *threshold* van vijf blijkt in de praktijk het meest geschikt) gekeken of dit het geval is. Hierna wordt de overeenkomstige kans toegekend en teruggegeven. De flowchart hiervoor is weergegeven in Figuur 92.

Draftingtijd beginwaarde bepaling

De initiële draftingtijd wordt op 0 geplaatst. Indien er een fietser wordt voorbijgestoken, en dus links in het beeld tevoorschijn komt, wordt deze op -5 geïnitieerd nadat er bijvoorbeeld drie keer is vanuit gegaan dat deze fietser aan het voorbijsteken is. Dit wordt visueel geïllustreerd in Figuur 93.



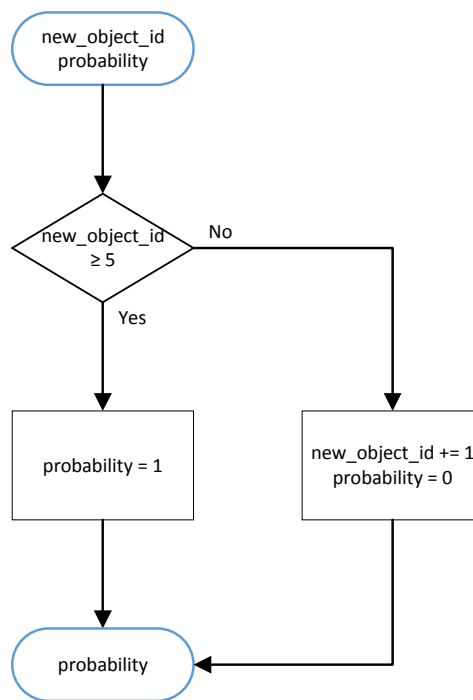
Figuur 90: Flowchart voor de bepaling van de dichtste fiets



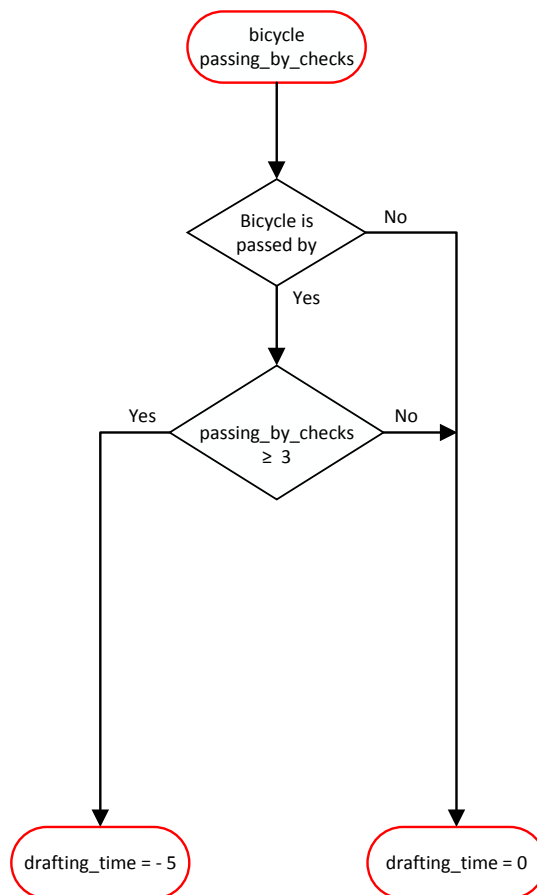
Figuur 91: Flowchart voor de draftingcontrole

Toekenning van een draftingpenalty

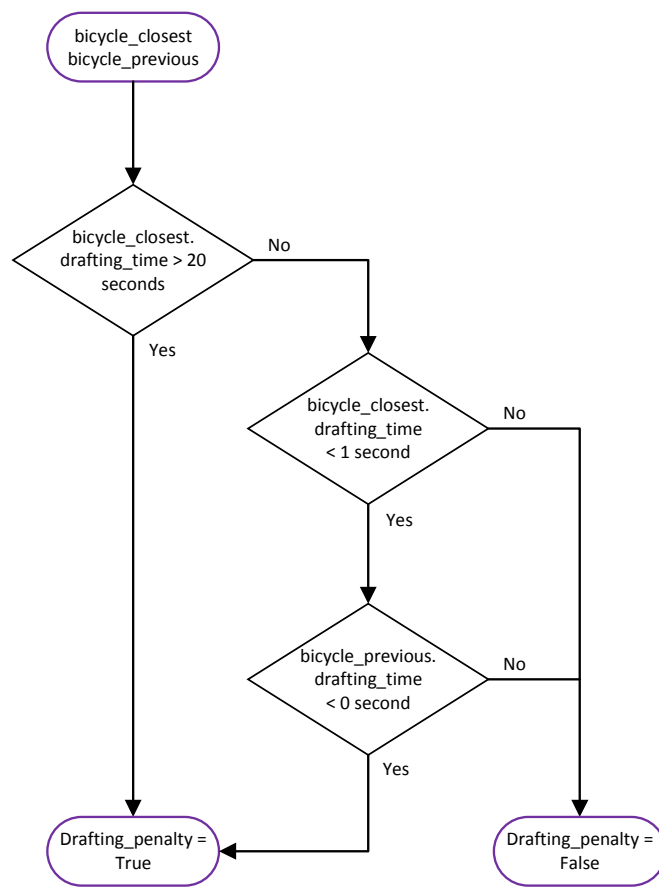
Er wordt automatisch een draftingpenalty toegekend als de triatleet zich langer dan 20 seconden in de draftingzone bevindt, alsook bij een voorbijgestoken fietser die zich langer dan 5 seconden in de desbetreffende zone begeeft. Dit wordt visueel weergegeven in Figuur 94.



Figuur 92: Flowchart voor de bepaling van een nieuwe fiets



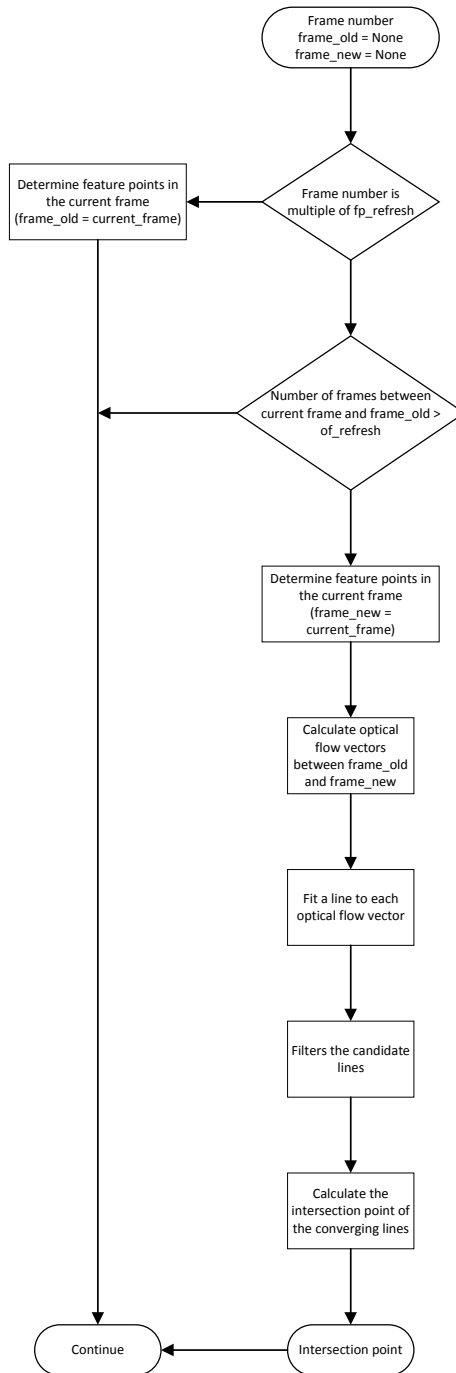
Figuur 93: Flowchart voor de bepaling van de initiële draftingtijd



Figuur 94: Flowchart voor de toekenning van een draftingpenalty

Bijlage E: Flowchart voor het algoritme van de kalibratiechecker

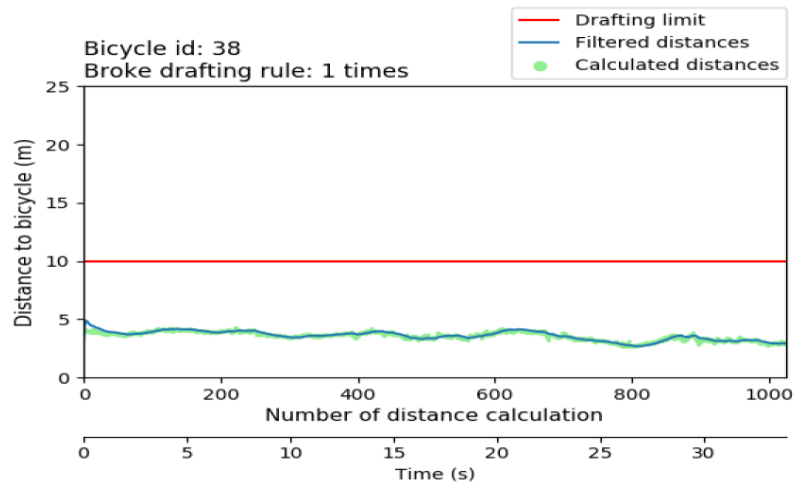
De flowchart van het algoritme voor de kalibratiechecker is weergegeven in Figuur 95.



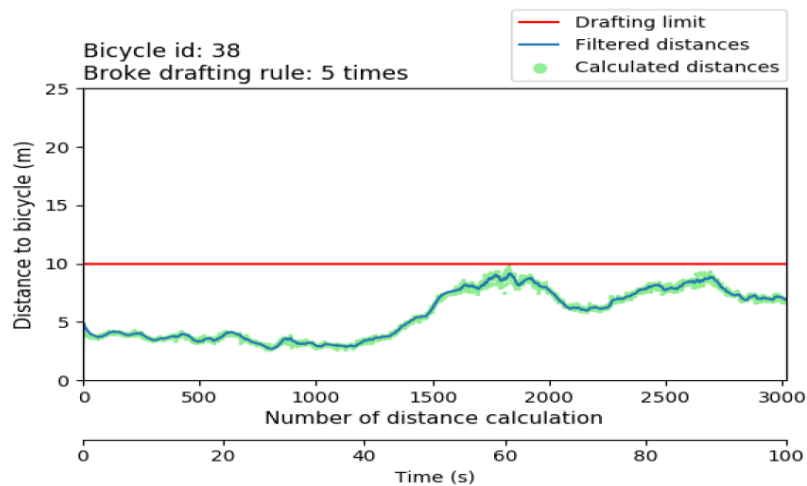
Figuur 95: Flowchart voor de kalibratiechecker

Bijlage F: Afstandsgrafieken voor enkele situaties tijdens een triatlonwedstrijd

Draften

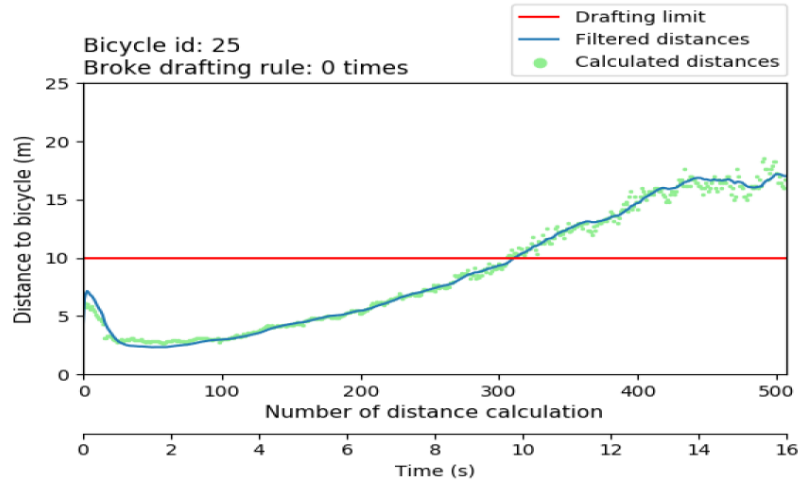


Figuur 96: Een atleet die meer dan 30 seconden aan het draften is



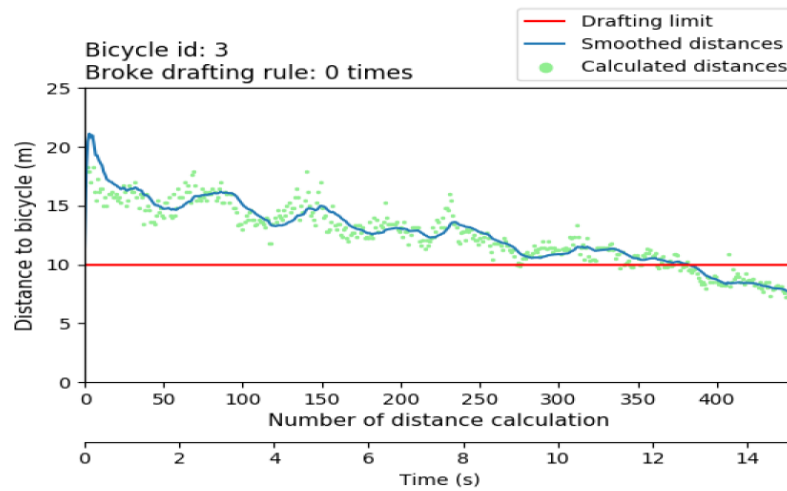
Figuur 97: Een atleet die 100 seconden aan het draften is

Vorbijsteken



Figuur 98: Situatie waarbij er iemand wordt voorbijgestoken

Vorbijgestoken worden



Figuur 99: Situatie waarbij de achterligger voorbijsteekt

Bijlage G: afstandstesten beeldregistraties

Beeldregistratie - Statische situatietest



Figuur 100: Voorbeeld van een beeldregistratie van de statische situatietest

Beeldregistratie - Dynamische situatietest



Figuur 101: Voorbeeld van een beeldregistratie van de dynamische situatietest

Bijlage H: Kostprijsraming en specificaties

Tabel 7 geeft een overzicht weer van een grove schatting van de fps throughput, het energieverbruik en de prijs. Op die manier wordt een idee gegeven over de mogelijke implementatie van deze proof of concept.

Tabel 7: Benodigdheden, specificaties en kostprijs voor het draftingdetectiesysteem

Product	Tiny YOLOv3 fps	Verbruik (Watt)	Prijs (€)
Camera:			
GoPro Hero 6 [91]	30/60	4	430
Ximea CMOS CMV20000 (MX120MG-CM) [92]	133	8	-
Ingebed systeem:			
Raspberry Pi 3 Model B+ (CPU) + Intel NCS 2 [93]	8,64	3.75 - 7.5	140
Nvidia Jetson Nano bord 4GB (GPU + CPU) [94]	25	5	100
Totaal	/	9 W	€ 530

Als de onderdelen in het grijs gemarkeerd beschouwd worden, dan wordt een grof totaal bekomen van 9 Watt en 530 euro voor respectievelijk totaal verbruik en prijs. Voor een rit van 3 uur en een power van 5,0 V komt dit neer op een benodigde batterij van 5400 mAh.

