Universiteit Leiden
The Netherlands

Opleiding Informatica

Ice Skater Detection

Using Computer Vision

Renz Roos

Supervisors:
Arno Knobbe & Daan Pelt

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

23/08/2024

Abstract

The primary purpose of this thesis is to explore the feasibility of using computer vision to accurately track and estimate the paths of ice skaters, particularly in the bends of an ice rink. This research aims to develop and validate a method that can distinguish and follow multiple skaters simultaneously, providing a reliable system for analysing performance and movement in sports.

The methodology involves collecting video data from an ice rink, followed by a series of preprocessing steps including white balancing, background subtraction, and erosion and dilation. Blob detection is then performed to identify the skaters in the video frames. Subsequently, an assignment algorithm matches data points to the correct skaters, and the RANSAC regressor is applied to filter out outliers and accurately estimate the skaters' paths. This process ensures the precise tracking of skaters' movements even in scenarios involving multiple individuals.

The results demonstrate that the combined use of blob detection, assignment algorithms, and the RANSAC regressor allows for accurate and efficient tracking of skaters' paths. The preprocessing steps significantly enhance the detection accuracy, and the RANSAC regressor effectively estimates the paths by filtering out outliers. The methodology proves successful in tracking a single skater and multiple skaters at the same time, providing clear and smooth plots that reflect the actual paths taken by the skaters.

The research concludes that computer vision can be effectively used for ice skater detection and path estimation. The developed methodology offers a robust foundation for further advancements in sports analytics, with potential applications in real-time tracking systems for competitive events. Future research could focus on enhancing blob detection under various lighting conditions, integrating deep learning techniques, and developing real-time processing capabilities to provide immediate feedback and insights during events.

Contents

1 Introduction

Ice skating is a popular sport that comes in many forms. In regards to this bachelor thesis, we will only be looking into speed skating. In speed skating there are still a few disciplines with distances varying between 500 and 10.000 meters. The ice rinks that are used are exactly 400 meter long and have two bends with the same radius between 25 and 26 meters. In these bends is where the skaters lose the most speed and time. This is why there is a big potential gain to be had if this lost speed is minimised.

The Dutch people have always been big on ice skating, so it should come as no surprise that The Netherlands is one of the leading countries when it comes to ice skating. This, in turn, leads to high expectations for athletes and the Dutch Royal Speed Skating Federation, or KNSB for short. To deal with these expectations and stay on top, the KNSB hasn't just invested in its athletes, but in its equipment as well. Innovation in speed skating began at a basic level, such as sharper blades, smoother ice surface and as of more recent, the use of data has become more and more important. The use of computers is no exception, equipment like transponders and sensors in the ice have already made its way into the sport. Advances like these make it easier to collect and analyse data sets. This, in turn, makes it easier to analyse skaters' movement, optimise training schedules and to detect patterns to prevent injuries and improve their overall performance.

With this context, the KNSB would like to be able to have more precise speed measurements, especially within the corners. Their current system is able to give the speed of a skater at twelve locations on the ice rink [thi17], but this isn't sufficient anymore. With better measurements, the skaters will be able to locate where to improve more easily and obtain higher sporting performance. So, there is a need to measure every fraction of a second. This bachelor thesis won't focus on the speed, but the location of the skaters and to see the path that the skater took through the bend. The master thesis of Michael de Rooij [dR24] will take a deeper dive into this topic and will measure the speeds of the skaters and their individual skates using a neural network.

Computer vision [Sze22] technology presents a promising possible solution. There are three main reasons for choosing computer vision. The first one is that computer vision is able to do calculations on every frame of a video, which checks off the requirement to measure at every fraction of a second. Secondly, the reason for computer vision is that the measurements can be done on the fly. So, there is no need for manual input. The last reason is that Thialf still has the cameras up and running, so there aren't any extra costs. Given the aforementioned information, the central research question guiding this bachelor thesis is:*"Can computer vision be used to accurately track and estimate the path of ice skaters in the bends of the ice rink?"*

2 Related work

There has already been a variety of research on the subject of speed skating. These works shall be looked at in section 2.1, after which section 2.2 will discuss how computer vision has already had a role in sports.

2.1 Speed skating research

In speed skating there are few areas of research. One of these areas is techniques used to maximise the speed of the skaters. As early as the eighties, research shows that there is an advantage to be had when the ice skater keeps their knees at a small angle relative to the fully extended leg [vIS83]. This, along with an optimal lean [EVdKV16] and push-off angle [DANK13], gives the skater the maximum speed through the corners.

However, there is more to it than just the stance of the speed skaters. A study shows that a thin layer of water on the ice's surface causes low friction during speed skating [JJdK92]. Moreover, there are also studies into the training of the athletes [JOK20] [AKC17] [JOF14]. There is even a study into the influence of having a home advantage [Kon05].

2.2 Computer vision in sports

Over the years, there has been plenty of research into using computer vision for sports [SL14] [Jef18]. Often, the first step is having to determine what to build the detection upon. Often the answer will be to use the openCV library. OpenCV is an open source library for computer vision and machine learning and is available in multiple programming languages and offers a lot when it comes to blob detection [HJL+14].

Hereafter the next step is often to use background subtraction. This can be challenging because the background probably won't be the same at every interval [BHB22], think of shadows, weather or viewing angle. So, it is important to adjust these and not keep what is considered the background a constant. Another consideration to keep in mind is subtracting the full image at once or looking at every pixel individually. This gets important when you work with high brightness [ACPL10].

Sousa et al. [ACPL10] showed that when working with such high brightness, it works better to determine for every pixel individually if they are part of the background or not. When one would subtract the image at once the result would be that the players would become almost undetectable, since they would be eliminated from the image.

3 Methodology

This chapter describes the approach taken to find an answer to the research question. Firstly, an overview of the video data is given accompanied with how the data was acquired. After this, the data will be put through white balancing, background subtraction, erosion and dilation as part of the blob detection. After the blob detection, the blobs will be filtered and cut out based on the desired aspect ratio. Lastly, the coordinates of the cutouts will be plotted and a RANSAC regressor will be applied to determine inliers and outliers per person on the videos.

3.1 Data

The data for this research was collected by filming multiple skaters going around one of the bends in the Thialf ice skating rink. The cameras are installed in the ceiling above the second bend of the rink. The videos were recorded in two separate recording sessions with different settings, causing us to have two types of videos with varying light settings, colour balance, frame rates and line markings on the ice.

3.2 Blob detection

Blob detection is the first part of the pipeline. The blob detection consists of three components, namely white balancing, background subtraction and erosion and dilation, which will be explained later in detail. The main goal of the blob detection is to see if it is possible to detect skaters and return their x and y coordinates as well as their timestamp.

3.2.1 White balancing

White balancing is the first preprocessing step in video analysis that ensures the colour balance of images is corrected, resulting in natural and consistent colours across all frames, even under varying lighting conditions. This step is particularly important when dealing with videos captured in different lighting setups, which our dataset is, and not accounting for it can lead to a significant effect on the accuracy of the video analysis.

The aim here is to make any type of video uniform and thus usable without having to reconfigure settings within the written code. An example is given in figure 1, here the image on the left looks too warm, meaning the RGB values have too much red in them. The image on the right has this balanced due to white balancing.

As mentioned in 3.1, our dataset consists of two different types of videos that vary light in light settings and colour. Applying white balancing would fix these differences and ensure a more uniform dataset.

3.2.2 Background subtraction

Background subtraction aims, as the name implies, to remove the background and only leave the parts of the frames that we are interested in. In our case, these are the ice skaters. In figure 2, the idea is laid out. There is an input stream of frames that go to the background subtraction. Here a

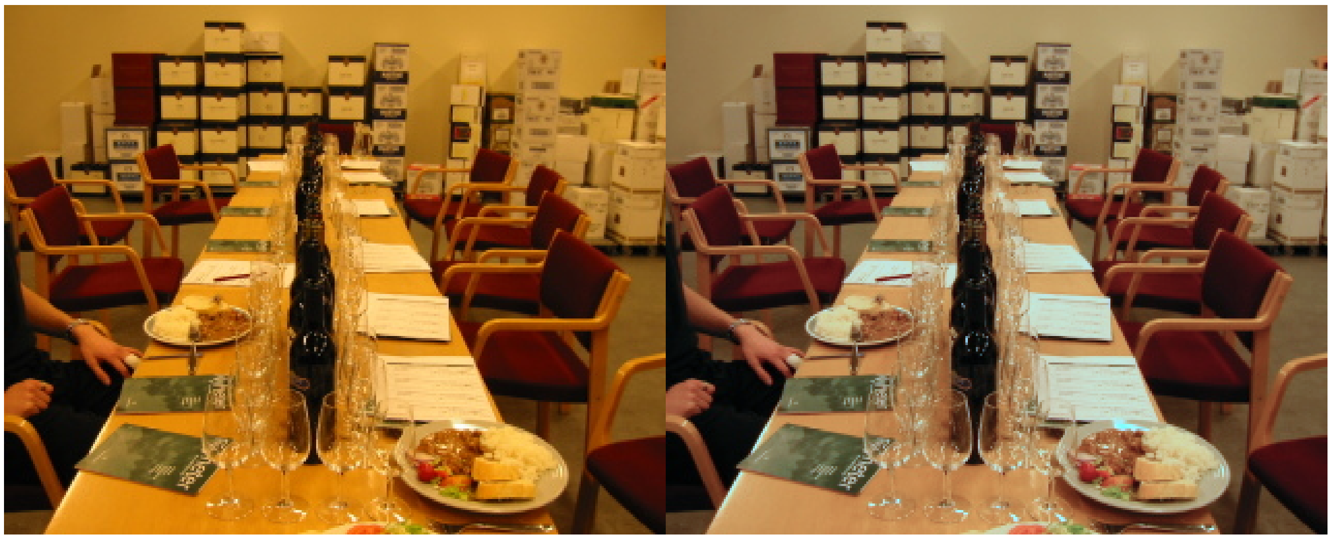

Figure 1: An example of white balancing [htt17].

predetermined background image will be subtracted from every frame. Both the frames and the background are in black and white, causing the resulting image to be black where the frame and the background image have the same brightness. When a pixel isn't completely black there is a differential. The greater this differential, the brighter the pixel. However, we want to binarize every pixel to say it is or it isn't part of the background. So, a threshold value will be set to turn the pixel into black or white.

Considering that our camera is stationary and that background is the stationary ice rink, the background will always be the same and thus relatively easy to remove. So, first we need to find a way to get an image with the background. After which we can subtract this from every frame in the video.

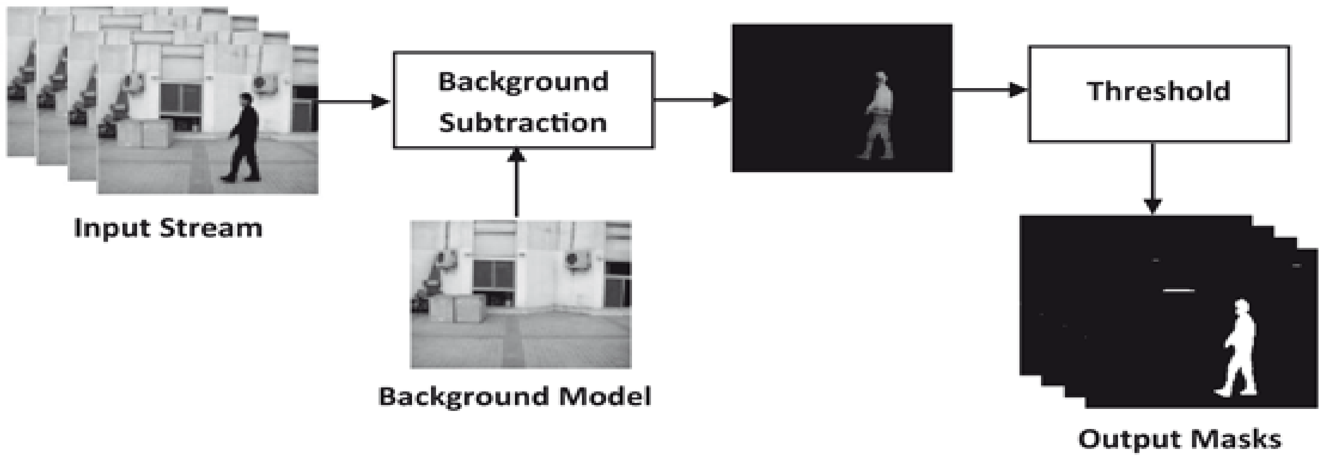

Figure 2: An example of background subtraction [Res20].

3.2.3 Erosion and dilation

After the subtraction is done, the resulting image will probably not only include the ice skaters, but some noise as well. This noise, which can interfere with the detection and produce inaccurate results, needs to be removed.

To address this, erosion will be applied to the image. Erosion works by reducing the boundaries of objects in the image, thus removing small noise elements. The downside is that it also erodes objects that aren't noise, in our case that would be the ice skaters. This causes the ice skaters to become smaller and thus harder to detect. This effect can be seen in figure 3. Here, the two grids on the top left show us the reduction of noise after erosion. However, the object of interest has become smaller and the hole has become bigger, which is unwanted.

To counter this effect, dilation will be applied. Dilation works by doing the exact opposite of erosion, so expanding the objects. This then should restore the original shape of the ice skaters which would be easier to detect again. This doesn't apply to the noise, because since the noise doesn't exist anymore, there is nothing to dilate.

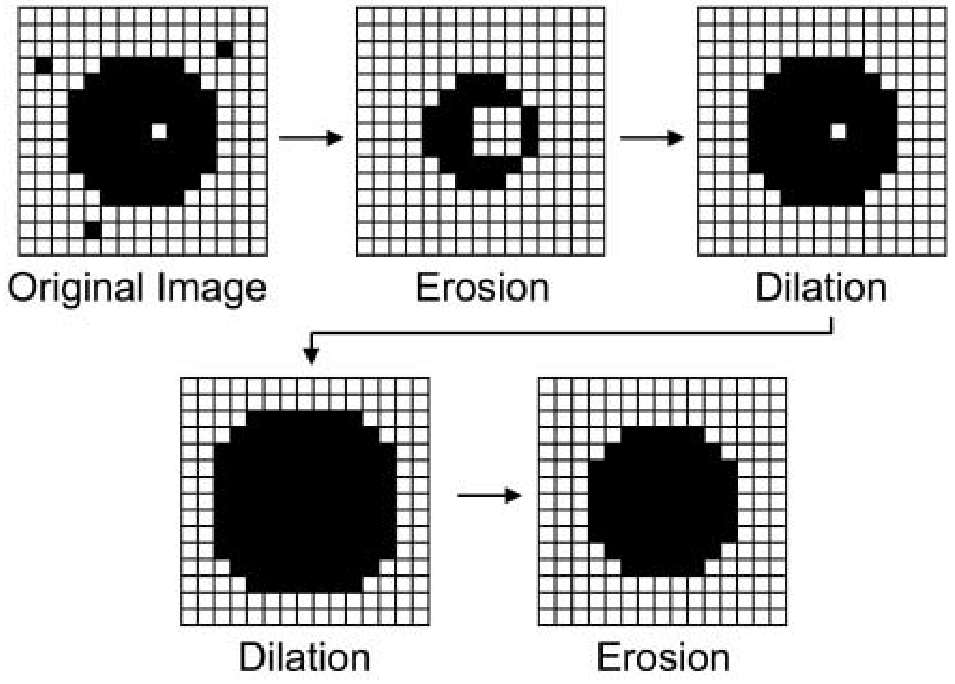

Figure 3: An example of erosion and dilation [Res05].

3.2.4 Cutouts

Now the skaters are ready to be detected. The detection works by finding blobs on the empty background. Here we will be returned the coordinates of every blob, or skater, as well as its size. This creates a bounding box around the blob.

With this information, cutouts around the skater are made. These cutouts are essentially cropped sections of the image, containing one skater, and are all the same size. The cutout size and bounding box size aren't equal. The aspect ratio of the cutout is predetermined and thus will need to either trim or extend the bounding box. Lastly, to ensure the skater remains the focal point of the cutout, the skater's centre should align with the centre of the cutout. This may require shifting the cutout's centre away from the bounding box's centre.

All of this is necessary, because this will create a dataset that will be used in a different research thesis [dR24] where these cutouts will be used in a neural network to detect the ice skaters' skates to measure the speeds of each skate individually. In order to put the dataset into the neural network, the data needs to be put into a specific aspect ratio.

3.3 Assigning data points

When the cutouts are made and all of the data is ready to be assigned, there is no information about how many skaters there are in the video. As there can be more than one skater per video, there needs to be a way to figure this out. Not only that, but the data points then need to be assigned to the correct person. This will be done with an algorithm, which will have to be specially written. More information will be found in section 4.3.

3.4 RANSAC

To get an answer to the research question, the path estimation still needs to happen. This can now be done since all data is divided and will be done with a RANSAC regressor [RAN].

RANSAC, meaning random sample consensus, is an iterative algorithm which is designed to estimate an underlying model from the data that it is given. RANSAC is also able to filter the outliers out of the data to ensure as few errors as possible.

The estimation starts by randomly selecting small subsets of data points and fitting a model to each subset. Each model will be checked on how many data points from the entire dataset fit this model. This will be done for many iterations and the model with the most inliers is selected as the best representation of the data.

In our case, this will be done for every skater individually, since they have their own set of data points. This means that the result will be as many graphs of path estimations as there are skaters in the video.

4 Implementation Details

In order to definitively answer the research question, it has been divided into sub-questions. To answer these questions, we will be performing experiments. After doing the experiments and subsequently answering the sub-questions, we can combine these answers to give an answer to the research question.

4.1 Data

As mentioned in 3.1, the dataset consists of two types of videos of ice skaters going around in one of the bends in Thialf. The two types of data are shown in figure 4. The differences become immediately clear and that leads us to our first sub-question, namely *"Is it possible to create one uniform dataset of these two types of data?"*.

The experiment is to white balance and resize the videos. The resizing is fairly straightforward, the openCV library has a built-in function to do this. White balancing is done by taking each frame and taking the RGB values of the frame and taking the power of 2.2 of this value. This is done because an RGB can at max be 255, meaning that a colour with a higher value will have a limited increase and colours with a lower value will gain more. After which the values are divided by its mean RGB value and multiplied by 0.3. This to move the mean down, which is to not blow out the picture. Lastly the frame is rooted by 2.2 again to remove the power of 2.2. This should balance out the colours and thus result in frames that look uniform [hr22].

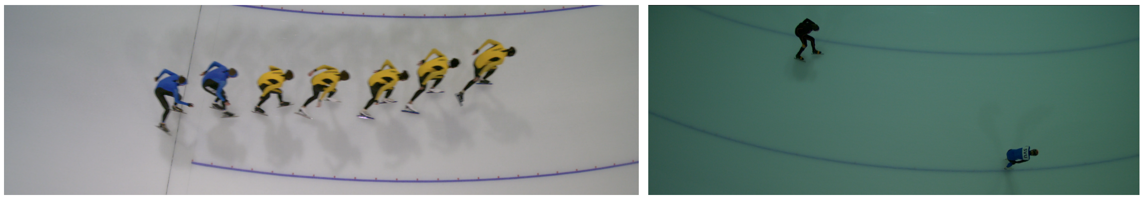

Figure 4: Comparison between the two types of data with examples of the older videos on the left and the newer ones on the right.

4.2 Blob detection

After the white balancing, we need to subtract the background from the frame. In order to subtract the background, we first need to establish the background. This creates two new sub-question *"Is it possible to establish what the background of the video is?"* and *"Is it possible to subtract the background from the current frame?"*.

To establish the background, we take the set of frames from the video and take the mean RGB value for every pixel over these frames. The mean RGB values should be close to the RGB value that was most present at a pixel location for the duration of the video. These values should then give us the mean image and thus the background.

Once we have the background, we can set up removing it from every frame. This is done by grey scaling both the current frame and the background image and taking the absolute difference of every pixel value. This, then leaves us with the skaters and probably some noise. To mediate the noise, first turn every pixel into either white or black, based on a threshold. After which, we erode the remaining image with a 5 by 5 kernel. What this does is it takes a pixel and if the 5 by 5 grid of pixels has more white pixels, the pixel becomes white, otherwise it will be black. This removes the noise, since noise often doesn't come in clusters. What now also happens is that the skater gets eroded, but since it does come as a cluster of pixels it won't be eroded completely. However, we still want to get these pixels back, so we dilate the image again. This time we do it with a 9 by 9 grid and we have 5 iterations, so the skaters become more clearly visible.

Once all of this is done, we can finally get to the detection part of the blob detection. Here we need to be able to detect the ice skaters and return their coordinates of the centre point of the bounding box. Thus, the sub-question for this part is *"Are we able to return the coordinates of the skaters for every frame?"*.

The image we got after the dilation is put into an openCV function called *findContours* [?]. This will give us the coordinates where openCV detected a blob. We then filter the blobs on size, since a human will always be relatively the same shape. With the coordinates, a cutout from the original frame will be made with the desired aspect ratio. If we need to expand the bounding box to fit the aspect ratio and the bounding box is on the edge, the cutout will be filled with a black box to fill the needed space. This is to keep the skater in the centre of the cutout, thus the centre point of the cutout should be the coordinates of the skater.

4.3 Assigning data points

After getting all the data points, along with their timestamps as well, we want to determine the path that every person took. Since, we don't have prior information about how many people were on the ice and the data doesn't tell us that directly either, we have to come up with a way to determine that from the data. Not only this, but we then also have to assign the data points to the correct person. So here we have two sub-questions as well, namely *"Can we determine the correct amount of ice skaters in the video?"* and *"Can we assign all the data points to the correct ice skater?"*.

The experiment is set up in a way that lets us answer both sub-questions with the same algorithm, the pseudocode is seen in algorithm 1. Here we go over all of the data points per frame. We create a dictionary of lists of data points, of which every skater gets one list. We start by checking if the index is smaller than the dictionary length, since it is empty we will add our first skater to the dictionary and assign the first data point. For the next data point, we then try to measure the mean distance between data points in the current dictionary entry, along with measuring whether the distance of the new data point is within a set range of the current mean. For this to actually happen, a skater needs at least two assigned data points. So, the try fails and we go to the exception where we are just going to assume it would be in range. It may still be from the same skater, thus we can check if the x value of the new data point is higher than the old one and if it is an older timestamp. If this fails it will probably belong to another person. When the lists in the dictionary have multiple entries we'll actually go through the try section of the code. First the mean distance of the existing list is calculated, then we check if the distance between the new data point and the last data point in the list is within a range of the mean distance. If this isn't the case and if the new distance isn't extremely far, we have a second chance based on the mean slope of the list. If the mean distance fails it could be that the blob detection failed to detect a skater for one or more frames, using the slope we can determine that the data point probably still belongs to that skater, assuming the skater is going at close to a constant speed. Once this algorithm is finished, we filter out dictionary entries that have a list of less than five entries, since these are probably part of someone else and thus not relevant to use on its own.

4.4 RANSAC

Lastly, we need to set up the RANSAC to get a graph of the path of the ice skaters. So, the sub-question here is *"Can we estimate the path taken for every ice skater?"*.
For the RANSAC, we use scikit-learn [sci] since it comes with a built-in RANSAC regressor[RAN]. The experiment here is set up by first graphing the x values against the timestamps and graphing the y values against the timestamp. We now apply RANSAC to filter out outliers and if a data point is considered to be an inlier in both the graphs it will be used for the graph where x and y will be plotted. Here we use RANSAC again to get the final path taken by the ice skater.

Algorithm 1 Data point assignment algorithm

```
for frames in numberOfFrames do
    dictionary ← Ø
    coordinatesOfFrame ← allCoordinates(frame)
    for coordinates in coordinatesOfFrame do
        i ← 0
        while i < length of dictionary do
            try:
                meanDistance ← Distance between all x and t of dictionary(i)
                newDistance ← Distance between x and t of last enrty dictionary(i)and coordinates
                inRange ← newDistance in range meanDistance*(2/3) and meanDistance*(4/3)
                if not inRange and newDistance/10 ≤ meanDistance then
                    inRange ← mean slope of dictionary(i)
                    in range of the slope of coordinates and last data point in dictionary(i)
                end

            catch:
                    inRange ← True
            end
            if coordinates.x > max(dictionary(i).x) * 0.99 and inRange and coordinates.t >
            max(dictionary(i).t) then
                dictionary(i) ← dictionary(i) + coordinates
            end


            i+ = 1
        end

        if i ≥ len(dictionary) then
            dictionary(i) ← Ø
            dictionary(i) ← dictionary(i) + coordinates
        end

    end

end
```

5 Results

In this chapter, we will answer the aforementioned sub-questions.

5.1 Data

To create the uniform dataset of the two types of data we have, we white balanced and resized the videos. In figure 5 the results are shown to be quite similar. The main differences are the blue lines and the angle of the video, but these aren't important to us and won't impact the blob detection. So, it would be acceptable to consider the videos uniform.

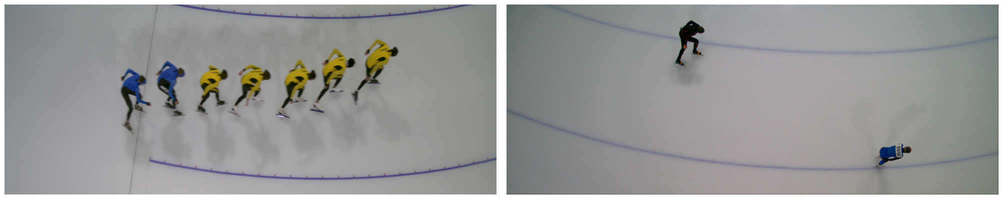

Figure 5: Comparison between the two types of data with examples of the older videos on the left and the newer ones on the right after white balancing and resizing.

5.2 Blob detection

The first question to answer for blob detection is about how to subtract the background. A comparison can be seen in 6, the image on the left is a handpicked frame from one of the newer videos where there are no ice skaters in frame, meaning that this is the actual background. However, we don't want to handpick frames and there is no way for the program to figure out in what frame only the background is visible. It can guess of course, but that won't always be correct. On the right is the resulting image we get after the background subtraction experiment described in 4.2. The images are near identical and thus we can safely say that it is possible to get the background.

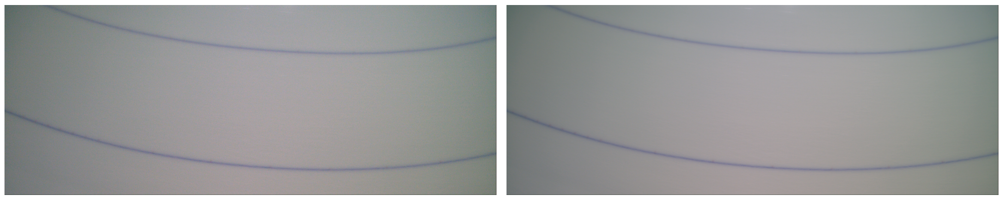

Figure 6: Comparison between a frame from one of the videos on the left and the resulting background image from that video on the right.

With this result, we can test to see if we are able to remove the background from the frames. The result is shown in figure 7, the images are in black and white since this is what the program uses as well. The image in the top left is our reference image with white balancing. In the image in the top right, we have the background removed. We see that the skaters are visible and nothing else. On the bottom image dilation is added to make sure in the next step the program has an easier time to detect the blobs.

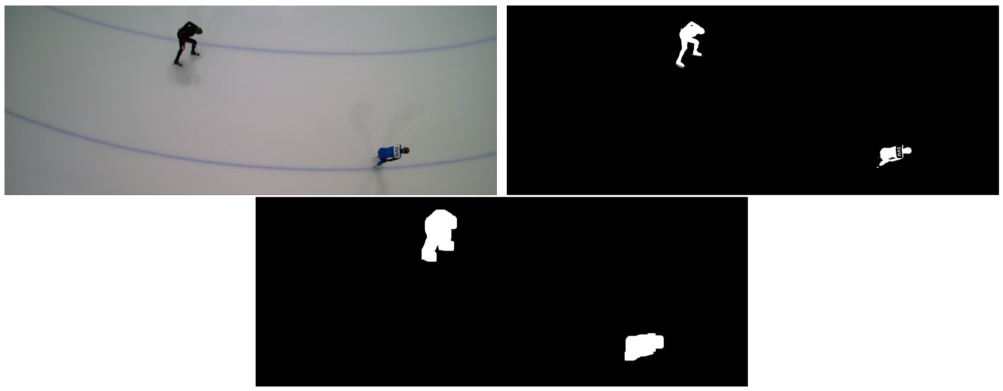

Figure 7: Comparison between a frame from one of the videos on the top left, the same frame with the background and noise removed on the top right and the same frame dilated on the bottom.

The last step for the blob detection is to see the results for the detection itself. In figure 8 we can see a frame of a video and the resulting cutouts of that frame. The red bounding boxes are where a skater is detected and the green dots are the centre points of the cutouts. The cutouts show all of the ice skaters in the centre, which is what we want. A few cutout examples are shown in figure 9. Here the first two are cutouts of the people seen in figure 8. The skaters are fully in frame and the skaters are precisely in the centre of the cutout. The last cutout is an example of how a cutout looks when the skater is on the edge of the screen. A black padding will be added to make sure all cutouts are the same size and to keep the skater in the centre. A small flaw is that sometimes people aren't picked up or are merged together and are too large to be considered an ice skater. So, sometimes data is missing, but these can be extrapolated based on the previous data by the RANSAC in the final tests.

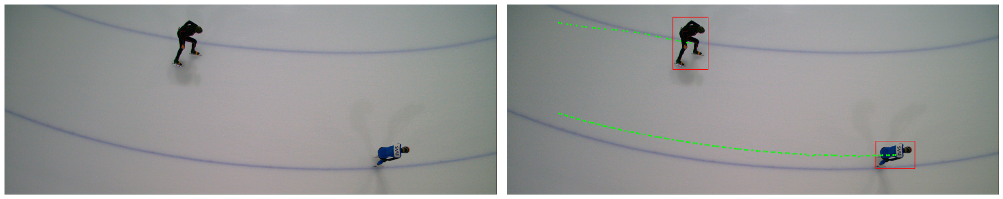

Figure 8: Comparison between a frame from one of the videos on the left and the same frame with the detected skaters on the right. On the right the red bounding boxes show the detected skater and the green dots show all centre points of all previous cutouts.

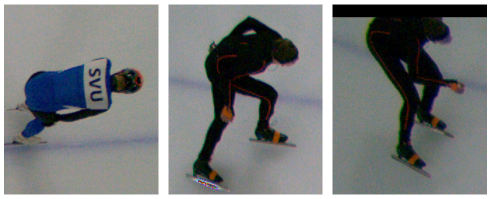

Figure 9: Comparison between a frame from one of the videos on the left and the same frame with the detected skaters on the right. On the right the red bounding boxes show the detected skater and the green dots show all centre points of all previous cutouts.

5.3 Assigning data points

The two sub-questions are answered using the same algorithm as mentioned in 4.3. In figure 10, the results of the algorithm are shown when only one skater is in the video. The graph of the on the left tells us the position of the skaters on the x and t graph. This shows our skaters going from left to right over the course of the video. A constant speed of the skater gives us a straight line from bottom left to top right. On the right, we have the x and y graph. This is the detected path this skater took. It is clear that there are some outliers and these will later be accounted for by the RANSAC.

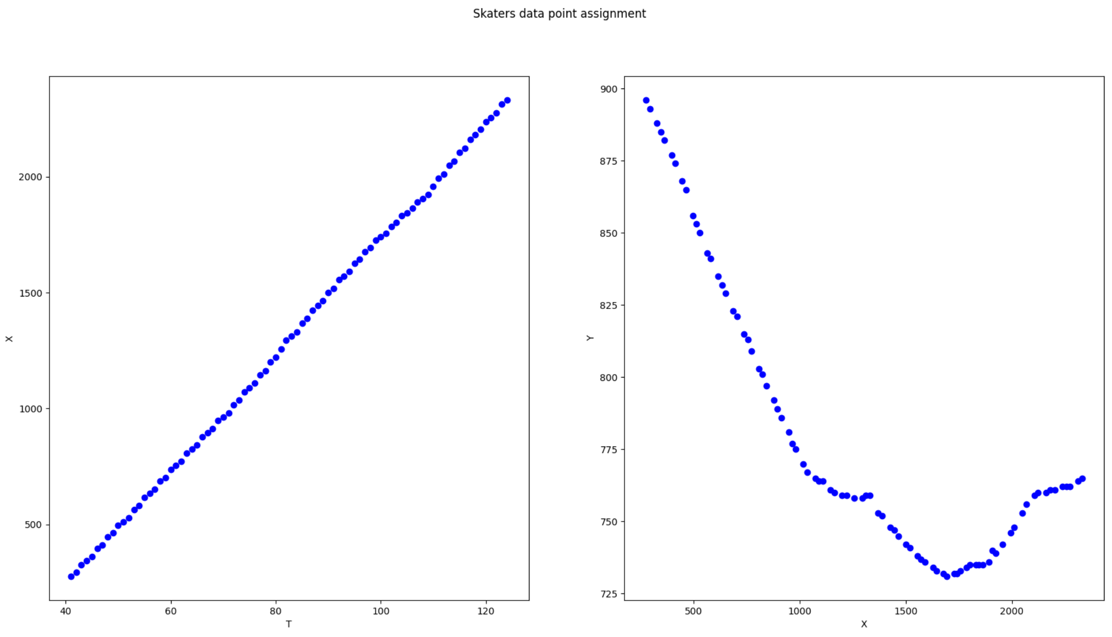

Figure 10: Results of the assignment algorithm for one skater.

With multiple ice skaters, the results get more complicated, but follow the same principle. Here we use the data collected in figure 8. The results are shown in figure 11. Every skater has its own colour, all colours in the x/t graph follow a close to linear line and thus we can safely say that the data points that are assigned are assigned to the correct skater. We can also count the number of skaters and state that the total is correct.

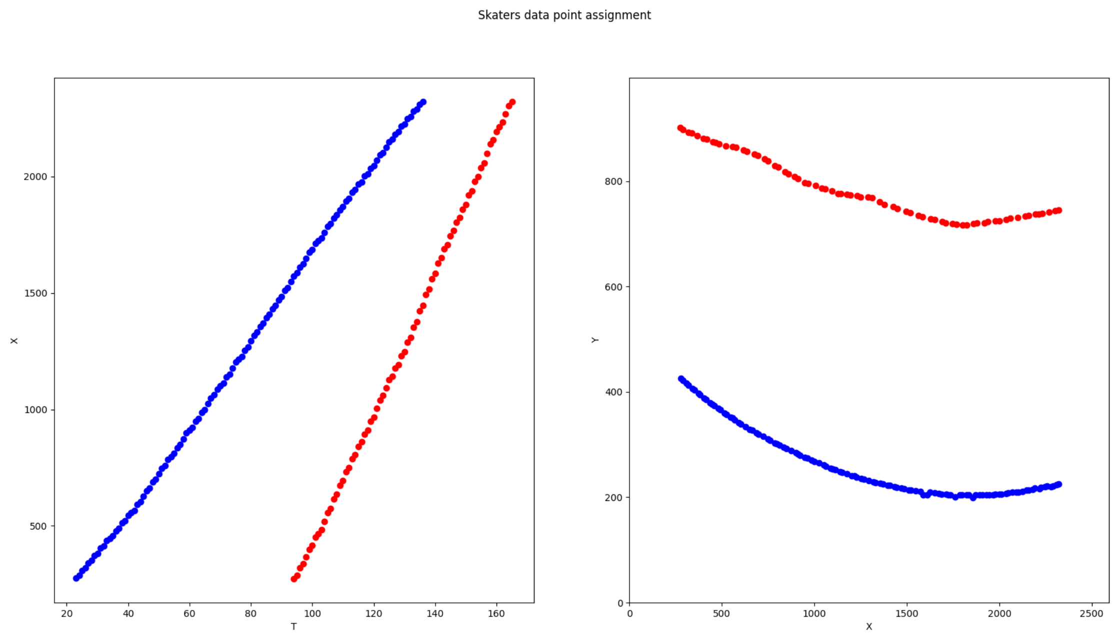

Figure 11: Results of the assignment algorithm for multiple skaters.

5.4 RANSAC

The last sub-question, we have to answer whether it is possible to determine the path taken for every skater. The results of the RANSAC are seen in figure 12. The data is for one skater only and the two rightmost graphs are the same data as in figure 11, but now RANSAC is applied. The blue dots are considered inliers and the green dots outliers. Only the data points that are inliers in both the graphs are considered a data point for the final x/y graph. This is shown on the left most graph, where the actual estimate of the path is drawn in red.

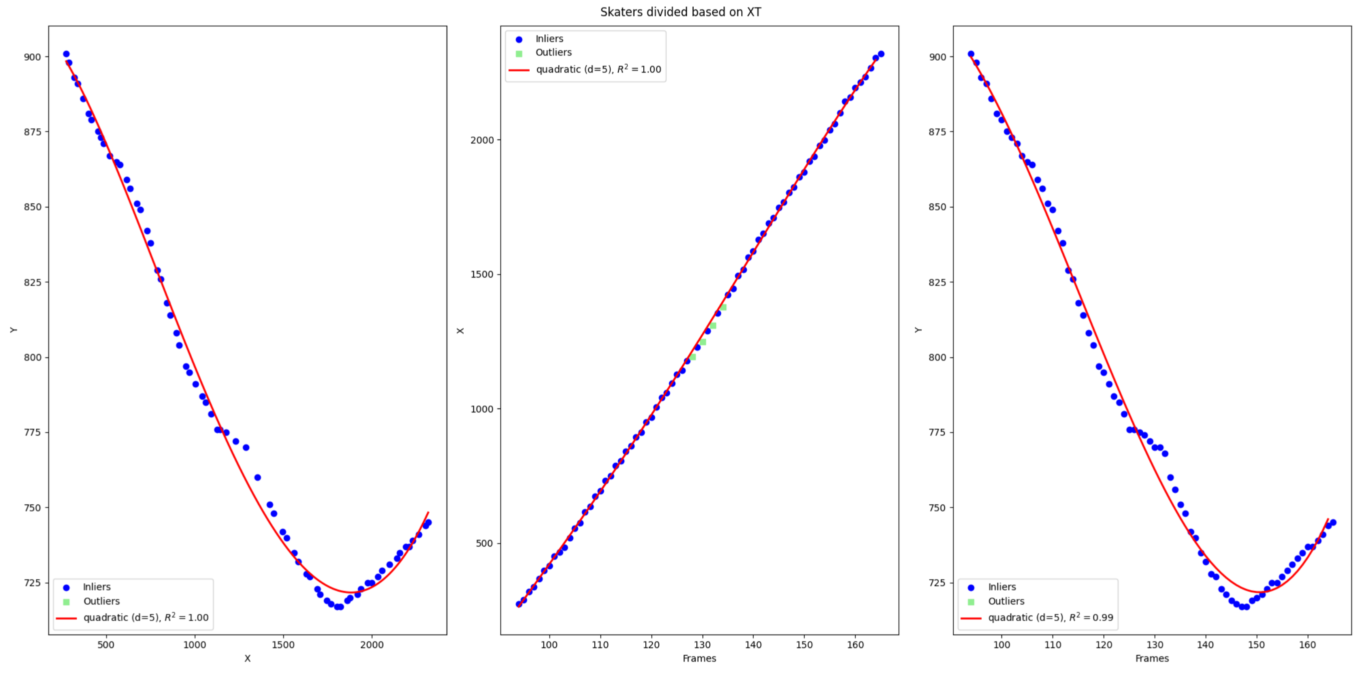

Figure 12: Results of the RANSAC regressor.

6 Conclusions and future work

In this research, we developed and evaluated an algorithm designed to track the paths of multiple ice skaters from video data using computer vision techniques. The approach combined blob detection, data point assignment, and the RANSAC regressor to accurately estimate skater paths.

The blob detection method effectively identified and tracked all individual skaters in most frames, even when multiple skaters were present. The process involved several crucial steps: white balancing, background subtraction, erosion, and dilation. White balancing normalised the lighting and colour balance across different video frames, ensuring uniformity in the dataset, which was necessary given the different types of videos. Background subtraction isolated the skaters from the static background of the ice rink, allowing the blob detection to focus on the skaters and not anything irrelevant in the background. Erosion and dilation, refined the blobs by removing noise and then dilating the image again to improve the accuracy of detection.

The assignment algorithm demonstrated robustness in distinguishing between skaters, ensuring that data points were correctly assigned to the correct person. As well as coming up with the correct amount of skaters in the video. These steps were crucial otherwise the RANSAC would be working with data that has no correlation.

The RANSAC regressor successfully filtered out outliers in the x/t and y/t graph, this in turn created a better estimate of the actual path taken by the skaters in the x/y graph. Here a smooth line is shown instead of the jaggedness of the data points.

The combination of these techniques allowed for accurate and efficient tracking of skaters' paths, even in scenarios involving multiple individuals. The successful application of this methodology suggests its potential for broader use in sports analytics and other domains requiring precise movement tracking. Coaches and athletes can use this system to analyse performance, optimise training schedules, and detect areas for improvement. Moreover, the insights gained from this research could inform the development of real-time tracking systems for competitive events, enhancing the ability to monitor and evaluate athlete performance instantly.

Future research could explore enhancements in blob detection under varying lighting conditions and environments to further improve accuracy and reliability. Advanced techniques such as deep learning could be integrated to handle more complex scenarios and diverse datasets. Additionally, the integration of machine learning techniques could improve the adaptability of the tracking system. For example, neural networks could be trained to recognize and track skaters more effectively, even in challenging conditions. Another potential area of improvement is the development of real-time processing capabilities. This would enable live tracking and analysis during events, providing immediate feedback and insights.

In conclusion, this research has demonstrated the feasibility and effectiveness of using computer vision for ice skater detection and path estimation. The methodology developed here sets a foundation for further advancements in sports analytics and other applications requiring precise motion tracking.

References

[ACPL10] Sousa A., Santiago C., Reis L. P., and Estriga M. L. Automatic detection and tracking of handball players, 2010.

[AKC17] Nico Hofman Benjamin Van der Burgh Arno Knobbe, Jac Orie and Ricardo Cachucho. Sports analytics for professional speed skating. *Data Mining and Knowledge Discovery, 31(6):1872–1902*, 2017.

[BHB22] Thulasya Banoth, Mohammad Farukh Hashmi, and Neeraj Bokde. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. *Applied Sciences*, 12:4429, 04 2022.

[DANK13] Marco JM Hoozemans Dionne A Noordhof, Carl Foster and Jos J De Koning. Changes in speed skating velocity in relation to push-off effectiveness. *International journal of sports physiology and performance, 8(2):188–194*, 2013.

[dR24] Michael de Rooij. *Video-based tracking of a speed skater in the corner*. PhD thesis, Leiden University, 2024.

[EVdKV16] FCT Van Der Helm E Van der Kruk, AL Schwab and HEJ Veeger. Getting the angles straight in speed skating: a validation study on an imu filter design to measure the lean angle of the skate on the straights. *Procedia engineering, 147:590–595*, 2016.

[HJL$^+$14] Markus Hovorka, Clemens Jung, Thomas Langenau, Philipp Lütge, Patrick Podest, Veronika Schrenk, and Bruno Tiefengraber. Opening opencv. *Junior Journal*, 2014. Received D M 2014; Accepted D M 2014.

[hr22] https://stackoverflow.com/users/2602877/christoph rackwitz, 2022. [Online; accessed August 23, 2024].

[htt17] https://stackoverflow.com/users/1337313/saptarshi. Example of white balancing., 2017. [Online; accessed August 23, 2024].

[Jef18] Colby T. Jeffries. Sports analytics with computer vision. *Senior Independent Study Theses. Paper 8103.*, 2018.

[JJdK92] Gerrit Jan van Ingen Schenau Jos J. de Koning, Gert de Groot. Ice friction during speed skating. *Journal of Biomechanics, Volume 25, Issue 6, 1992, Pages 565-571,*, 1992.

[JOF14] Jos J de Koning Jac Orie, Nico Hofman and Carl Foster. Thirty-eight years of training distribution in olympic speed skaters. *International journal of sports physiology and performance, 9(1):93–99*, 2014.

[JOK20] Laurentius A Meerhoff Jac Orie, Nico Hofman and Arno Knobbe. Training distribution in 1500m speed skating: a case study of an olympic gold medalist. *International journal of sports physiology and performance, 16(1):149–153*, 2020.

[Kon05] R. H. Koning. Home advantage in speed skating: Evidence from individual data. *Journal of Sports Sciences, 23(4), 417–427.*, 2005.

[RAN] scikit-learn, RANSACRegressor. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html.

[Res05] ResearchGate. A digital vision chip for early feature extraction with rotated template-matching ca, 2005. [Online; accessed August 23, 2024].

[Res20] ResearchGate. Example of background subtraction., 2020. [Online; accessed August 23, 2024].

[sci] scikit-learn, Machine Learning in Python. https://scikit-learn.org/stable/.

[SL14] Hou Lihong Shen Li. Research of background segmentation method in sports video. *TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.12, No.6, June 2014, pp. 4274-4282*, 2014.

[Sze22] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Cham, 2022.

[thi17] Thialf eerste ijsbaan met realtime feedback snelheid!, 2017. Last accessed 07-03-2024.

[vIS83] de Groot G. Hollander A.P. van Ingen Schenau, G.J. Some technical, physiological and anthropometrical aspects of speed skating. *Europ. J. Appl. Physiol. 50, 343–354*, 1983.