IT UNIVERSITY OF COPENHAGEN

QUANTIFYING CYCLIST BEHAVIOR AT INTERSECTIONS USING AUTOMATED VIDEO ANALYSIS

Bachelor project

May 17, 2021

edbe | Edi Begovic

hoja | Høgni Jacobsen

Supervised by Michael Szell

Abstract

In many European cities, such as Copenhagen and Amsterdam, cycling is an essential part of urban transportation. Transport infrastructure often prioritizes cars, resulting in unfriendly or even dangerous environments for cycling. Having quantifiable data on the movement and behavior of cyclists is a crucial element in modern city planning and would result in friendlier urban infrastructure for cyclists. The current data collection methods are reliant on the human annotation of video recordings, which is expensive, time-consuming, and could face challenges in reproducibility. Here we set up and evaluate a new methodological approach to replace manual annotation. In this approach, we set up a data collection and analysis pipeline consisting of camera selection and setup, video syncing, camera calibration, object detection, projection, merging of different camera sources, and object tracking. We also develop a web app for efficiently exploring collected footage. To test our setup in a case study, we record and analyze videos of the Dybbelsbro intersection in Copenhagen with multiple cameras. Our method succeeds in detecting cyclist trajectories and transforming them into a birds-eye view, allowing us to explore both fine-grained details and an aggregated view of cyclist behavior. Using our method, we produce favorable results compared to manual annotation.

Contents

1 Introduction **1**

2 Background **1**
2.1 Inclusive infrastructure 1
2.2 Previous work 1
2.3 OpenDataCam 2

3 Methodological proposal **3**
3.1 Initial tests with OpenDataCam 3
3.2 Proposal 3

4 Implementation **4**
4.1 Data collection 4
4.2 Data Processing 5
4.3 Analysis 8

5 Case studies **9**
5.1 Interpretation of desire paths 9
5.2 Desire line counts 9
5.3 Behavioral analysis 9

6 Discussion **10**
6.1 Privacy 10
6.2 Limitations 10
6.3 Future work 11

7 Conclusion **11**

Appendices **13**

A Behavior analysis **13**

B Undistorted pictures **14**

1 Introduction

With cities supporting more extensive diversity in modes of transportation, having quantifiable data on cyclist movement is a crucial element in modern city planning. Movement patterns of cyclists and other traffic participants and the intersection in-between is to a large degree determined by the design of the shared infrastructure. Traditionally, traffic behavior analysis has been dependent on manual human review, either by observing locations on-site or through recorded footage (Schneider, Patten, and Toole, 2005). This approach comes at a high cost, and unless rigid methodology is defined beforehand, reports are often ad-hoc and qualitative.

While bicycle paths at intersections do increase cyclists' perception of safety, studies conducted in Denmark show a significant increase in accidents after the installation of bicycle paths (Agerholm, Caspersen, and Lahrmann, 2008). We thus aim to build a system that can automatically capture the behavioral patterns of cyclists at intersections. Using recorded video footage from stationary cameras and applying modern object detection algorithms, we can extract the trajectories of road users. This in turn can help identify the relationships between behavior and infrastructure and their effect on safety.

This project therefore takes an experimental approach by exploring what parts of modern mobility analysis can be automated through video analysis, especially centered around cyclists at intersections. A considerable benefit of implementing such a system comes from the ability to scale and standardize measurements. As both cameras and hardware for graphical computation have become readily available, such an approach has become more financially viable.

2 Background

2.1 Inclusive infrastructure

As bicycle infrastructure is primarily publicly funded, it quickly becomes a politicized topic. While many cities push for a *'build it, and they will come'*-approach, the need for a certain mass of cyclists to attract the casual rider is often neglected. If new cyclists feel uncomfortable (stress, embarrassment, etc.), they are more likely to give up on the bike as a viable means of everyday transportation. This leaves the infrastructure underutilized, with the public image of money and space being wasted, thus resulting in a pushback against further cycling initiatives (Lind, Honey-Rosés, and Corbera, 2020). Therefore, it is crucial to carefully plan new cycling infrastructure and rapidly adapt to the observed behavior of new users.

2.2 Previous work

2.2.1 Bicycle choreography

In the study (Copenhagenize Design Co., 2014) an intersection in Copenhagen is reviewed, focusing on high-level aspects such as routes taken by cyclists, as well as the behavior exhibited by the individual cyclists. This is done using 'desire lines Analysis', a tool pioneered by the same authors as the study (Colville-Andersen, 2018a). Desire lines encompass the trajectories of cyclists using both 'intended pathways', usually directed through infrastructure such as bike lanes, but also their own paths, which may deviate from the engineered structure or legal framework. To assess behavior, the study splits cyclists into three groups:

- **Conformists**: fully rule compliant
- **Momentumists**: lose interpretation of current rules - encompasses actions legal in other countries
- **Recklists**: dangerous behavior - both for self and others

The proportions of cyclists belonging to each of the above groups were then compared for a day. Notably, the share of *momentumists* increased during the afternoon.

A recurring observation made by the study was the tendency of cyclists to get influenced by their fellow riders. This behavior is described as 'follow the leader', with cyclists being more likely to imitate non-conformist actions when another cyclist goes first.

2.2.2 Staging mobilities

The book 'Designing Mobilities' (Jensen, 2014) explores *'what physical, social, technical, and cultural conditions contribute to the staging of contemporary urban mobilities'*. It covers how to capture and represent *mobilities* and flows by presenting a theoretical

framework for 'staging mobilities'. Most importantly it introduces a set of concepts for articulating *situational perspectives*, including the two metaphors *the river* and *the ballet*.

The *river* references the notion of capturing mobility from a 'birds-eye view', including the layout, flows, and obstacles of a location, and hence, can be interpreted as a 'flowing riverbed'. From this perspective, we can observe road users' behavior in aggregate and as part of more significant 'streams'. This can be used to encapsulate abstract and high-level features of cyclists, such as the desire lines when infrastructure is changed, or temporary obstacles come along (e.g. road work).

In contrast, the *ballet* is the micro-perspective, encompassing the gestures and small interactions at the individual level. This perspective is an important supplement to the *river* as subtle interactional patterns, for instance, signaling agreement or evasion cannot be seen from above.

2.2.3 Mobilities in 'situ'

Study of mobilities in 'situ' (Olsen and Vries Bækgaard, 2015) uses the above framework as part of a case study involving cycling infrastructure in Copenhagen and Amsterdam. Quantitative data (as part of the *river* perspective) was obtained by filming cyclists and conducting 'desire lines analysis' (Colville-Andersen, 2018b) to gain a structured overview of the behavior of cyclists and intersections.

The Dybbølsbro/Ingerslevgade intersection in Copenhagen, also central to our experimentation, is part of a longer route covered in the study. Most notably, in the desire line analysis, it was found that the north-south connection made up 40% of the cyclists flow, with east-west making up 27%. The remaining 33% of cyclists traffic thus consisted of left or right turns. Critical points were also mapped out, mostly involving interactions between cyclists and pedestrians when congestion was high.

For the *ballet* perspective, the study performed 'ride-along' interviews. The main components of this interview framework consist of greeting participents and asking any generel questions, followed by the 'ride-along' stage where the interviewer closely follows the participent on a stretch of their journey. This particular study followed the participants on bike from behind, recording them with a camera mounted on the handle bars. At last, the 'follow-up' stage consists of a longer interview (40-60 minutes) revolving around the ride.

2.3 OpenDataCam

OpenDataCam (ODC) is an existing open-source tool for processing video footage from urban environments to quantify traffic. The novelty of OpenDataCam comes from the simple on-site, real-time processing of video.

The backbone of OpenDataCam is YOLO[1], *You Only Look Once*, a real-time object detection algorithm. YOLO is trained on the COCO dataset[2], which comprises over 330,000 thousand images with over 1,5 million objects in over 80 object classes, including cars, people, and cyclists, well known for its speed and accuracy (Redmon et al., 2016). Because of the computational needs of YOLO, for OpenDataCam to run locally, it requires dedicated CUDA-enabled hardware such as an Nvidia Jetson-series device.

The user interacts with the application through a web interface (figure 1) that displays a live preview of the footage with indicators around all detected objects.

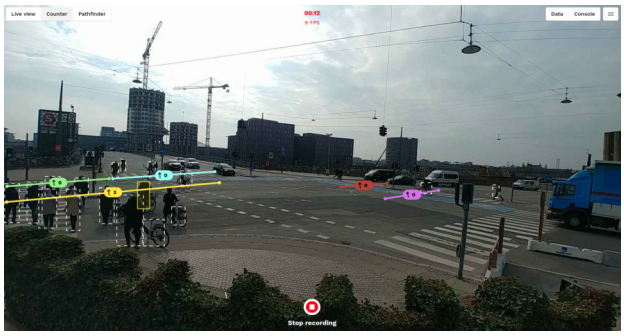

Figure 1: *OpenDataCam Web Interface*

Using the interface, a user can draw bi-directional counting lines, which, when passed by a detected object at a certain angle, increments a counter and keeps track of the object's identifier. Apart from counting functionality, the app also offers the trajectories of uniquely identified objects.

[1] YOLO object detection
[2] COCO dataset

3 Methodological proposal

While on-site assertion of an intersection is plausible, the scope of a single observer is limited, as one can only keep track of so many cyclists at once. Likewise with analyzing recorded video footage, while not bound by time, is still a time-consuming process, especially as the views of cyclists can be obstructed by other passing vehicles or permanent fixtures. As an example, previous desire line studies conducted by Copenhagenize were done by keeping counts of hand-written references to trajectories (figure 2).

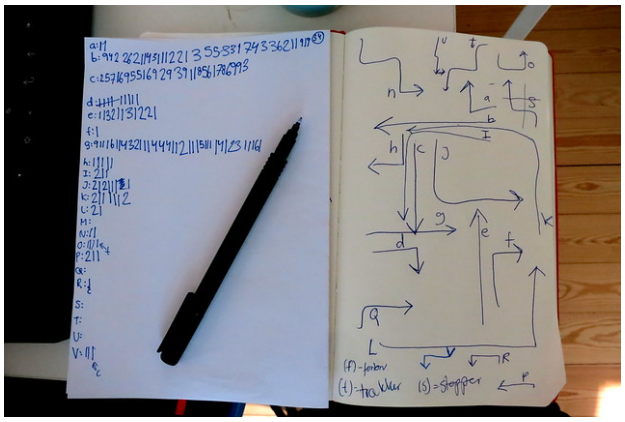

Figure 2: *Manual desire line counts (Copenhagenize, 2014)*

3.1 Initial tests with OpenDataCam

As a baseline, we deployed an OpenDataCam setup using pre-recorded footage from the Dybbølsbro intersection in Copenhagen. Our setup ran on an Nvidia Jetson Xavier NX combined with two Raspberry Pis for filming/streaming video from the intersection.
Our goal was to extract pre-defined desire lines using the counting line functionality. We did this by drawing an initial 'base line' followed by two (or more) additional counting lines, which would allow us to branch cyclists into multiple segments (figure 3). Inspecting the overlap of unique cyclists who passed both the base line and one (or more) of the branching lines would yield which path they took through the intersection.

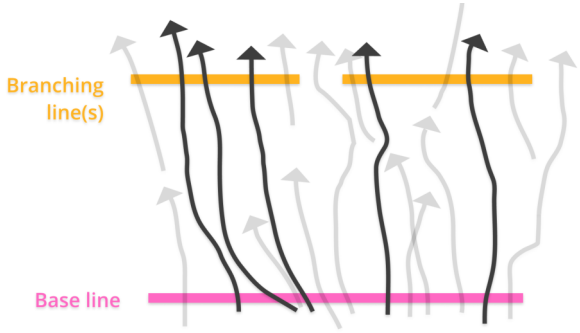

Figure 3: *'base line' (pink) and branching lines (orange)*

However, this approach was limited by the performance of the detection of cyclists implemented in the current version of OpenDataCam (YOLOv4). Due to the intermittent identification of cyclists, the tracking algorithm would constantly re-identify the same cyclists as new ones. Thus, even when the same cyclist passed both the 'base line' and one of the branching lines, they would not be included in the desire line count.

We discuss the findings derived from OpenDataCam in our case study in later sections of the report.

3.2 Proposal

We wish to improve and extend the functionality of OpenDataCam, centering around cyclists and intersections. Using a newer version of YOLO for object detection will result in better tracking accuracy for cyclists, and thus more complete trajectories. While OpenDataCam emphasizes real-time metrics on traffic, we specifically work with 'post-processed' footage from intersections, and therefore we can apply further assumptions about the tracking behavior to improve accuracy.

The aforementioned issue of keeping track of unique cyclists as their view is temporally obstructed is also faced by object tracking algorithms in video. In theory, a top-down overhead view overlooking the center of the intersection would be optimal for minimizing obstruction. However, the height required for such a recording setup, as well as the lack of permanent fixtures, makes this approach very impractical. Current state-of-the-art object detection algorithms (e.g. YOLO) also perform worse at categorizing objects, especially bicycles, using an overhead view compared to a

semi-frontal view (Velastin et al., 2020). We, therefore, propose deploying a multi-camera setup overlooking an intersection from multiple sides.

A top-level aerial view can help identify rule compliance through the intersection by comparing the shares of cyclists using chosen desire lines to reach the same destination. To bring nuance to the analysis, one can dig into the footage to assess the behavior of each identified cyclist using a specific desire line. This can provide insights about the context to non-intended behavior such as state of traffic signals, vehicles and the momentum of other cyclists.

4 Implementation

The approach takes place in three discrete phases: *data collection*, *data processing* and *analysis*, as summarized by figure 4.

Figure 4: *Overview of implementation*

4.1 Data collection

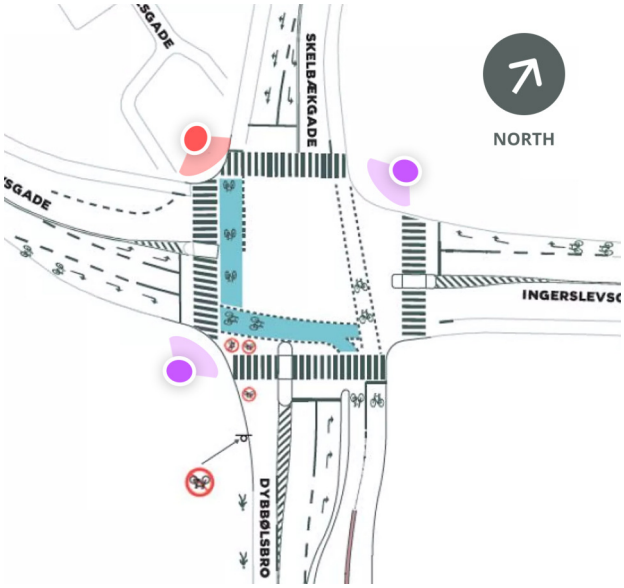

Figure 5: *Dybbølsbro/Ingerslevgade intersection*
Image source: Københavns Kommune

Our work is centered around the Dybbølsbro / Ingerslevgade intersection south-west of Copenhagen city center. The intersection faces several challenges, producing *'conflicts, unsafe situations, illegal road user behavior, and great dissatisfaction among road users at the intersection'* (*The Copenhagen Post* 2021). From the south, the intersection connects to the Dybbølsbro bridge, which features a bi-directional cycle path, two lanes for vehicle traffic, and pedestrians from the nearby train station. This is an unusual setup in Copenhagen, as most streets feature unidirectional cycle paths following vehicle traffic. Therefore, southbound cyclists coming from the north (Skelbækgade) have to perform a two-phase turn to continue over the bridge. Firstly by heading straight over the intersection but then stopping and performing a left turn to continue. This leads to high congestion at the south-west corner of the intersection with many cyclists looking for a shortcut across. Due to its issues and size as well as previous analysis conducted on the intersection, we found it to be a good baseline for experimentation.

4.1.1 Recording location

There are two considerations to take into account when recording an intersection for video analysis.

Intersection size

The size of the intersection will determine the camera setup needed. Larger intersections require a two-camera setup, however, this might not be optimal for intersections much larger than the Dybbølsbro intersection. Initially, we used a single-camera setup but found that while we had a good view of the entire intersection, we lost tracking data when the view of the cyclists' on the opposite side of the intersection was obstructed by vehicles.

Camera mounting points

We tried three different mounting locations shown by the red and purple circles in figure 5. We faced several difficulties with camera mounting at the intersection, including camera stability, obstructions of the camera view, and the height of the mounting positions. With some of our camera mounts, stability was impacted by wind and cause the camera to sway, resulting in skewed detections later on. To cover the entire intersection with adequate coverage, we selected a two-camera setup on opposite sides of the intersection (purple circles on figure 5).

4.1.2 Camera setup

Any camera device with an adequate field-of-view (FOV) to capture the intersection that records in HD (1280×720 resolution) at a constant frame rate (at least 10FPS), and that allows remote viewing of its viewfinder will suffice. However, having tested a Raspberry Pi recording setup (including a camera module, external battery, and an LCD touch screen), we would not recommend a self-built recording setup. Although the Raspberry Pi setup meets the above-mentioned technical requirements, the challenges lie in the usability of such a setup. It introduces unnecessary complexities by having to manually assemble the hardware and install the necessary software, as well as having surety of its operation in the field. Alternative devices optimized for video recording, such as mobile phones or action cameras, are more suitable. These devices typically offer methods of remotely controlling and viewing their software interface from another device such as a PC or smartphone. This is useful when mounting the camera and adjusting its position to get a good view of the intersection. While also possible with a Raspberry Pi, it is less stable compared to using smartphones or action cameras. Hence, we chose to use smartphones for recording. We used two different smartphones, an *LG G6* and a *Samsung Galaxy S7*; however, we recommend using the same recording devices. Using the same device brings consistency when applying this method.

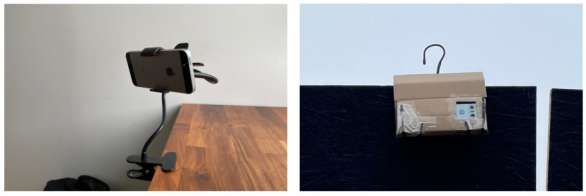

Figure 6: *Camera mounts*

We used two different camera mounts at the intersection (figure 6). One includes a flexible arm that can be clammed onto infrastructure such as light poles. The second mounting solution was a self-built holder made out of cardboard and a hook, allowing us to easily mount it at high spots using a long extension rod. Selecting an appropriate mounting height is a trade-off between maximizing the captured area of the intersection and having a frontal view of traffic. We aimed to have cameras mounted at least 3-4 meters above the ground.

4.2 Data Processing

An overview of our proposed data pipelines are shown in figure 7. The implementation can be found in the following git repository.

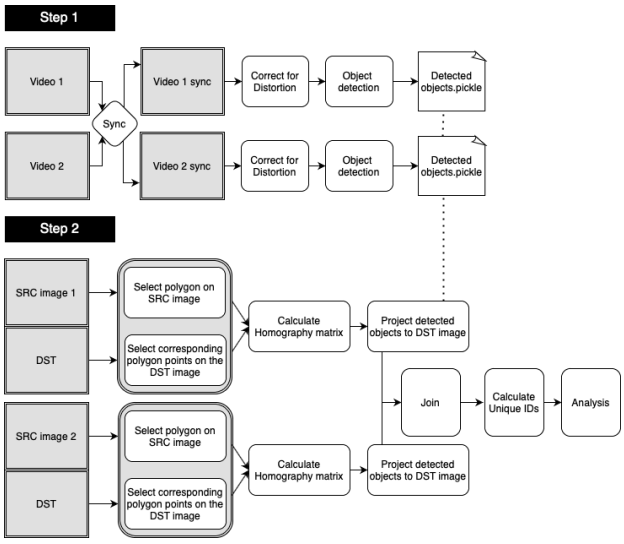

Figure 7: *Generalized Data Pipelines*

4.2.1 Video synchronization

An important preprocessing step is to cut the different video sources to the same timestamp. Should the video have different frame rates or resolutions, then these should be processed to match.

4.2.2 Camera calibration

Cameras often suffer from optical aberration where straight lines appear bent, especially noticeable in the edges of an image. This can be observed on wide-angle camera lenses, such as on the LG G6 we used. The specific type is *positive radial distortion* as shown in figure 8, with lines curving outwards in a barrel shape. These aberrations are a result of the curved shape of the camera lens.

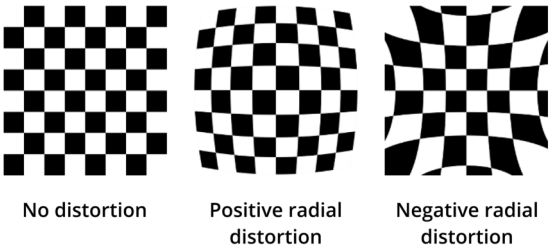

Figure 8: *Types of lens distortion*

There is also a possibility of further distortion if the camera sensor and the lens are not parallel, also known as *tangential distortion*. Ideally, there should be no radial nor tangential distortion.
It became apparent when merging data from multiple camera sources that lens distortion affected the alignment of data points. This is shown in figure 9, where trajectories detected from one camera source do not match the other.

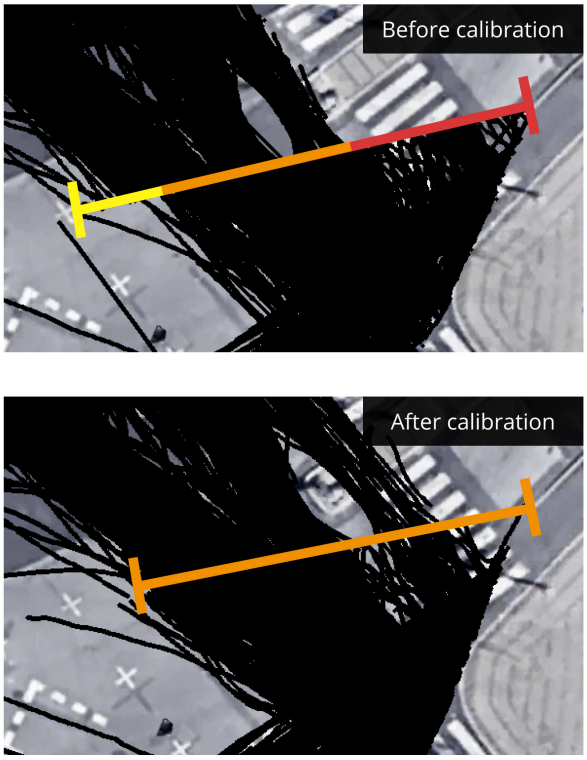

Figure 9: *Misaligned sources - before and after calibration*

To correct for the distortion we made use of OpenCVs camera calibration toolbox[3].
In order to correct for distortion, the camera matrix and distortion coefficients are needed. To get the camera matrix and distortion coefficients, we used the calibrate script provided with OpenCV, passing several snapshots of a calibration object taken with the device to be calibrated. The calibration object is a black-white chessboard pattern. We passed 20+ sample images of the calibration object in different orientations, angles, and positions in the frame. OpenCV precesses the images and returns a camera matrix and the distortion coefficients.

See appendix B for examples of before and after distortion correction.

4.2.3 Object detection

We found that our offline OpenDataCam setup using an Nvidia Jetson NX device worked well for detecting cars, but did not perform as well on cyclists. We, therefore, implemented our object detection setup using YOLOv5. YOLOv5, trained on the same COCO

[3] OpenCV

dataset[4] as previous YOLO models, showed a marked improvement on the detection of cyclists.

We ran YOLOv5 using the largest model available, yolov5x6. The model was run on an NVIDIA Tesla P100 GPU (CUDA enabled) with 16GB of memory. The model was set only to detect cyclists and to process video at a resolution of 1280 pixels.

With this setup, videos were processed at 8 FPS. This results in a total processing time of circa 5.5 hours for a 1.5-hour video. Predicted objects are represented as bounding boxes with the output being represented as $[[frameid][xmin][ymin][xmax][ymax][confidence]]$.

4.2.4 Projection

To achieve the 'birds-eye view' of the intersection, we needed to explore methods of transforming the data from the camera views.

A homography matrix is a transformation matrix between two planes (Hartley and Zisserman, 2004). It will allow us to transform and plot the cyclist movements onto an aerial photograph of the intersection.

[4] COCO dataset

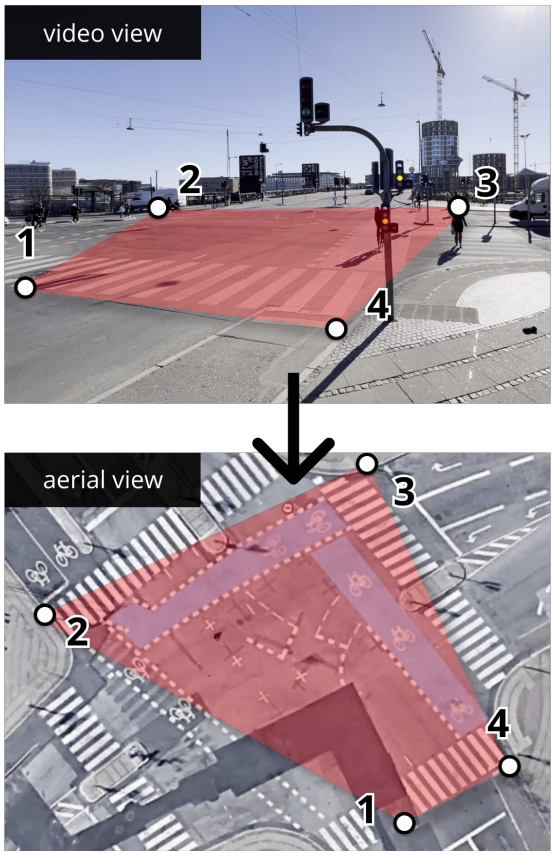

Figure 10: *Projection - ground control points*

To calculate the homography matrix, we need to solve the system of linear equations, $P = HQ$, P being points in a polygon on the source image (Figure 10, video view), Q the corresponding polygon on the destination image (Figure 10, aerial view). H being the 3×3 homography matrix.

Before we can project the cyclist onto an aerial view, we first need to calculate the 2D coordinates of their contact points with the road surface. Given equation 1, we can calculate the contact points using the objects bounding box coordinates.

$$\begin{aligned} x &= x_{\min} + \frac{(x_{\max} - x_{\min})}{2} \\ y &= y_{\min} \end{aligned} \tag{1}$$

After solving the homography matrix, we can perform the projection. This is achieved by applying the cyclist's contact coordinates, P, and the homography matrix to the equation $P = HQ$. This returns the corresponding pixel coordinates of the cyclist on the destination image.

4.2.5 Joining sources

As we are using multiple video sources, there will be an overlap of the detected cyclists. That is, the trajectory of the same cyclist will be captured by multiple cameras for specific areas. To merge the trajectories from the video sources, we take a naive approach. As the cameras are set up on opposite sides of the intersection, we cut the video sources in half (See figure 11) along the mid-point between the two cameras along the intersection to join them.

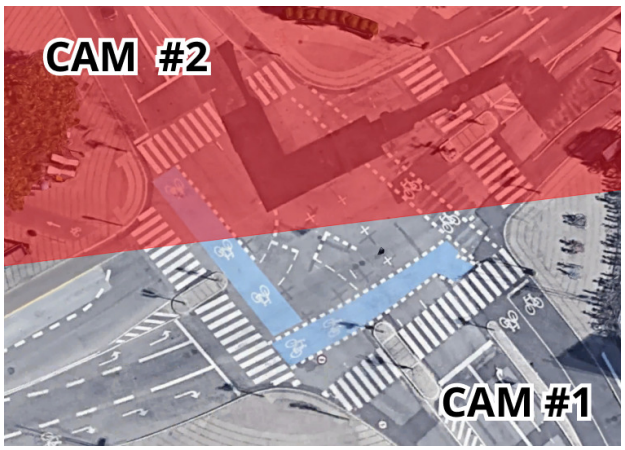

Figure 11: *Cutting of video sources*

4.2.6 Tracking

In order to connect detections into trajectories of individual cyclist, we apply SORT[5], *simple online and real-time tracking*, as initially described in (Bewley et al., 2016). SORT aims to address multiple object tracking (MOT), where objects across frames need to be connected.

SORT uses Intersection over Union (IOU) as a matching criterion for detections between frames. The IOU is calculated by dividing the overlap between the bounding boxes by the union of the bounding boxes. To identify objects between frames the IOU of a detection B in frame $t+1$ is compared with predicted position, using Kalman Filters, in frame $t+1$ of detection A from frame t, this is done for all detections between frames. The Hungarian algorithm is then applied to perform the optimal matching of detection.

In our implementation, we define arbitrary bounding boxes around the center point of the projected detections; we used a 15×15 pixel bounding box. Using arbitrary bounding boxes results from the plane of the bounding boxes from the object detection being on a different plane to that of the road's surface.

[5] SORT algorithm

4.3 Analysis

4.3.1 Web application

To interact with the tools implemented, we built a simple web application, figure 12, using Dash[6] for Python. Its main component consists of a satellite/aerial view of the chosen intersection with the trajectories of cyclists plotted as short-lived tracks. One can then explore the timeline by either explicitly referencing a timestamp or frame, using a slider, or jumping in set increments. Below the aerial view is a set of image viewers showing the current frame from all the cameras with markers around selected cyclists.

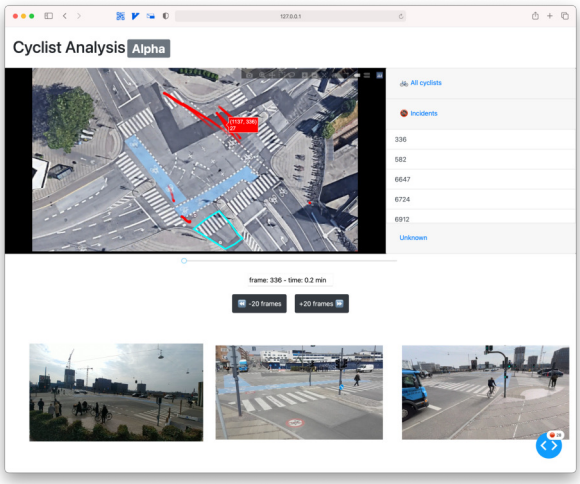

Figure 12: *Web application*

To gain insight into specific areas of the intersection, observations can be filtered by drawing a mask on the aerial view. When an area is selected, all cyclists passing it will be listed in a side panel, from which one can jump directly to the matching timestamp. We also implemented counting lines (as found in OpenDataCam), giving the user the ability to immediately see the number of cylists passing certain parts of the intersection in either direction.

[6] Dash library

Future work
Currently, the data preprocessing is not part of the application, thereby functioning more as a *proof of concept*. It would be relatively easy to connect the data processing pipeline to the web interface, thus simply making the user select video sources, map the *ground control points* (for aerial projection), and potentially tweaking other hyper-parameters.

4.3.2 Rainbow tracks

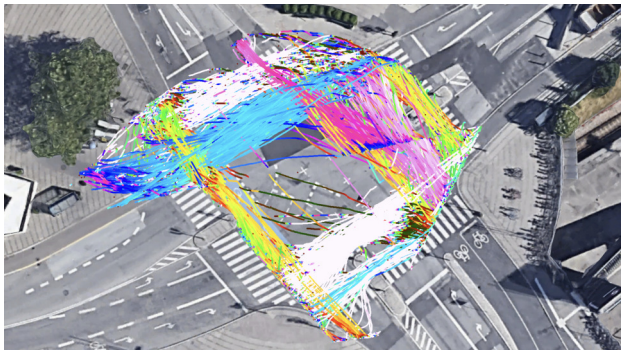

Figure 13: *Rainbow tracks*

To find aggregated desire lines from the data we took an approach which we call 'Rainbow tracks', as seen in figure 13. This involves coloring tracks by the bearing between consecutive points in each trajectory. After calculating the bearing, we then get a color from a gradient color wheel. This approach has the added benefit of encoding direction into each track.

5 Case studies

To quantify the effectiveness of our method, we compared various metrics from manually annotated data. Manually annotated data acting as the ground truth. Further, we also incorporate the same metrics from OpenDataCam.

5.1 Interpretation of desire paths

From the trajectory analysis in figure 13 we derived eight commonly used desire paths. We mainly focus on desire lines for south-bound traffic from Skelbækgade (figure 14) as referenced in section 4.1.

Figure 14: *Common desire lines of cyclists coming from NW*

5.2 Desire line counts

We initiated two uni-directional counting lines in the middle of the intersection as a way to access the traffic volume for desire lines 1 and 3 (fig. 14).
In table 1 we show a comparison of manually annotated counts, OpenDataCam counting lines, and our implementation. The counts are based on 1.5 hours of footage and only counting objects detected as cyclists.

	Manual annotation	Our method	OpenDataCam
1 (A+B)	192	184 (-4%)	63 (-67%)
3 (A+B)	99	102 (+3%)	50 (-49%)

Table 1: *Comparison of counts*

While our counting method got close to the true counts, OpenDataCam captured less than half the cyclists passing its counting lines. Further, while the absolute counts of OpenDataCam did not match the manual annotation, nor did the proportions between the two counts. Although the counting line implementation varies between our tool and OpenDataCam, the difference can primarily be attributed to the better detection accuracy of bicycles in YOLO5.

5.3 Behavioral analysis

As mentioned in section 4.3.1, we couple the top-level view with references to timestamps in the recorded footage, such that the behavior of cyclists can be observed at street level. Whenever a unique trajectory is selected, or a part of the intersection is marked off, one can inspect each cyclist and the context leading up to a detection.

In figure 15 we see a detection of a cyclist passing straight over Dybbølsbro instead of turning left and continuing over the bi-directional cycle path (see desire line **2A**). Unknowingly thinking he is right, he closely passes, at full speed, another rider crossing the street. Shortly after, he notices the traffic markings and turns around.

Figure 15: *Detection zone triggered*

Other cyclists (unknowingly) performing the same maneuver all tended to look to their left as they passed the pedestrian crossing (examples in appendix A). As they look, they also steer slightly in the same direction, which can lead to close contact with vehicles in the narrow car-only lane.

Another commonly observed behavior was cyclists trying to shorten the trajectories of their turns (fig. 16) at the 'waiting corner' at Ingerslevgade (see desire line **1B**).

Figure 16: *Shortend left-turn*

When volumes of other traffic were low or over the short window of all traffic having red lights, cyclists would use the opportunity to swiftly cross the intersection while still 'following the correct path' (that is, not crossing in a fully diagonal path).

6 Discussion

6.1 Privacy

Consideration should be taken towards privacy and the legality of collecting and processing data.

In the European Union (EU), where major cycling hubs of the world lie, we are governed by the General Data Protection Regulation (GDPR). GDPR regulates the processing of personal data relating to individuals by individuals, companies, and organizations.

Article 6 of the EU GDPR deals with the 'Lawfulness of processing'. Specifically, Article 6(1)(f) makes provision for legitimate interests, which could cover the scope of this research. However, this is not the only consideration. Local authorities shall be consulted before commencing any research.

6.2 Limitations

6.2.1 Hardware

The Galaxy S7 used does not live up to our criteria regarding its ability to record the entire intersection from its mounting location. While it captured an adequate FOV, we could produce better results with a camera that is up to standard. The choice to use this device was due to only receiving a different device too late in the process.

6.2.2 Multiple object tracking with joined data

We currently have no method, other than visual inspection, of quantifying how well the multiple object detection works along the joining line of the two videos.

6.2.3 Low congestion

Filming took place at 12 PM on a Wednesday, this is a period of relatively low congestion at the Dybbølsbro intersection. Higher congestion periods were not considered.

While we have seen promising object identification results with small groups of cyclists, it is unclear as to how object detection would perform in intersections

with higher congested.

6.3 Future work

6.3.1 Trajectory clustering

Trajectory clustering such as in Gariel, Srivastava, and Feron, 2011 would be a future addition to the tool. Clustering would enable a more fine-grained examination of desire lines, along with greater metadata such as traffic volume per desire path.

6.3.2 Improved multiple object detection

The SORT method of multiple object detection could be replaced with its extended version Deep SORT. Deep SORT extends on SORT by, in addition to IOU as a linking criterion, incorporating deep features of the objects for linking. Deep SORT could greatly improve the multiple object tracking and paving the way for an implementation that uses more than two cameras.

6.3.3 More detectors

Rule compliance and unintended behavior are tightly coupled with the state of the traffic signal at intersections. Having functionality to mark traffic lights and can filter results based on their state would be a useful addition. *Road User Behaviour Analysis* (RUBA)[7] is a video analysis tool for traffic developed at Aalborg University. While its detection mechanisms do not support mode classification as in our method or OpenDatacam, it features various other specialized detectors, including traffic light state. Another detector in RUBA is the 'stationary detector', which activation when road users are detected but non-moving in a marked area. For cyclists, this kind of detection would help identify spots with high congestion.

7 Conclusion

This study proposes a tool for automatic analysis of cyclists at intersections using video analysis. The setup is centered around the 'staging mobilities' framework, giving a planner both the ability to inspect cyclist traffic at intersections from multiple perspectives. All footage for experimentation was filmed with multiple cameras at the Dybbølsbro intersection in Copenhagen due to its complexities. Using computer vision to detect cyclists from the videos, we extract the trajectories of cyclists and map them to a single aerial view from multiple sources. A light web application was built to interact with the processed data. It shows the trajectories of individual cyclists over time and an aggregated view of the most common paths. Further, contextual analysis can be done for cyclists who have passed an area-of-interest marked on the areal view through references to the footage. We showcase the application of the tool at the Dybbølsbro intersection by inspecting cyclist counts and segments of footage in areas-of-interest. The results closely match the annotated data and are achieved in a short timespan.

[7] Road User Behaviour Analysis - RUBA

Bibliography

Agerholm, Niels, Sofie Caspersen, and Harry Lahrmann (2008). "Traffic safety on bicycle paths: results from a new large scale Danish study". English. In: null ; Conference date: 13-04-2008 Through 14-04-2008.

Bewley, Alex et al. (2016). "Simple online and realtime tracking". In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

Colville-Andersen, Mikael (2018a). *Copenhagenize: The Definitive Guide to Global Bicycle Urbanism*. English.

– (2018b). *Desire Lines and Understanding Behavior*. URL: https://link.springer.com/chapter/10.5822%2F978-1-61091-939-5_12.

Copenhagenize Design Co. (2014). "The Choreography of an Urban Intersection". In: p. 36. URL: https://pedromadruga.com/bicycle-choreography-copenhagenize.pdf.

Gariel, Maxime, Ashok N. Srivastava, and Eric Feron (Dec. 2011). "Trajectory Clustering and an Application to Airspace Monitoring". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4, pp. 1511–1524. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2160628. URL: http://ieeexplore.ieee.org/document/5959983/ (visited on 05/10/2021).

Hartley, Richard and Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, pp. 88–92. DOI: 10.1017/CBO9780511811685.

Jensen, Ole B. (2014). *Designing Mobilities*. English. Art and Urbanism Series 4. Aalborg Universitetsforlag. ISBN: 978-87-7112-098-1.

Lind, Adam, Jordi Honey-Rosés, and Esteve Corbera (2020). "Rule compliance and desire lines in Barcelona's cycling network". In: *Transportation Letters* 0.0, pp. 1–10. DOI: 10.1080/19427867.2020.1803542. eprint: https://doi.org/10.1080/19427867.2020.1803542. URL: https://doi.org/10.1080/19427867.2020.1803542.

Olsen, Andreas Hansborg and Mikkel de Vries Bækgaard (2015). "A study of mobilities 'in situ': Understanding cycling as an embodied practice". In: *Aalborg University*, p. 82. URL: https://projekter.aau.dk/projekter/en/studentthesis/en-undersoegelse-af-mobile-situationer(33f27421-ce1d-4f6c-b24f-ea1dff6d62c4).html.

Redmon, Joseph et al. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. arXiv: 1506.02640 [cs.CV].

Schneider, Robert J., Robert S. Patten, and Jennifer L. Toole (2005). "Case Study Analysis of Pedestrian and Bicycle Data Collection in U.S. Communities". In: *Transportation Research Record* 1939.1, pp. 77–90. DOI: 10.1177/0361198105193900110. eprint: https://doi.org/10.1177/0361198105193900110. URL: https://doi.org/10.1177/0361198105193900110.

The Copenhagen Post (2021). URL: https://cphpost.dk/?p=123482.

Velastin, Sergio A. et al. (2020). "Detecting, Tracking and Counting People Getting On/Off a Metropolitan Train Using a Standard Video Camera". In: *Sensors* 20.21. ISSN: 1424-8220. DOI: 10.3390/s20216251. URL: https://www.mdpi.com/1424-8220/20/21/6251.

Appendices

A Behavior analysis

1.a

1.b

2.a

3.a

4.a

5.a

6.a

7.a

8a

9a

B Undistorted pictures

LG G6

Samsung Galaxy S7