Video-Based Risk Assessment for Cyclists

Miguel Nobre da Costa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Doctor Manuel Ricardo de Almeida Rodrigues Marques

Professor João Paulo Salgado Arriscado Costeira

Examination Committee

Chairperson: Professor João Fernando Cardoso Silva Sequeira

Supervisor: Doctor Manuel Ricardo de Almeida Rodrigues Marques

Member of the Committee: Professor João Miguel Raposo Sanches

May 2017

“The capacity to learn is a gift; The ability to learn is a skill; The willingness to learn is a choice.”

-Brian Herbert, House Harkonnen

Acknowledgments

First, I would like to thank both my supervisors: Professor João Paulo Costeira and Doctor Manuel Marques. Both made me feel welcomed at ISR while still provided a tremendous willingness to help and share their insights throughout the whole thesis. Moreover, I would also like to acknowledge Beatriz Quintino Ferreira for her insights and her valuable help.

Secondly, I would like to thank my family: my mother Isabel, who keeps our family together; both my grandparents who always gave their all so that their children and grandchildren would have a better life than what they had; and finally, my father Carlos, who I know would be extremely proud of my academic achievements if he was still with us.

Abstract

Due to their zero pollution emissions, health improvements benefits and ease of access bicycles are gaining an increasing popularity as a mean of transportation in today's world. However, traffic accidents involving bikes are not decreasing, as well as fatalities. Thus, it is important to assess cyclists' safety in urban scenarios to allow city planners to develop better infrastructures that foster better protection for cyclers.

Therefore, from smartphone captured data and video, we propose a video-based framework to assess dangerous situations for bicyclists. We take advantage of motion estimation (optical flow) to estimate the Focus of Expansion on a set of images and then use this to define risk areas on the image. We then use the defined areas on the image to create a risk descriptor on the given situation and given the detected objects on the image. Our framework enables the assessment of risk on different criteria (Path Occupation and Proximity) based on our risk descriptor.

Finally, we test our framework on real data gathered from the improved developed smartphone application and achieve promising results.

Keywords

Cyclist, Risk Assessment, Focus of Expansion, Optical Flow, Smartphone Data, Computer Vision

Resumo

Devido a ser um meio de transporte com zero emissões poluentes, ter benefícios para a saúde e a sua facilidade de acesso, a popularidade das bicicleta está a aumentar nos dias que correm. Contudo, o número de acidentes onde estão envolvidas bicicletas e consequentes fatalidades parece não estar a diminuir. Por isso, é importante que se devsenvolva um método que consiga analisar quão seguro este meio de transporte é, para que planeadores urbanos consigam desenvolver melhores estruturas e redes para bicicletas, como ciclovias, de modo a promover uma melhor segurança para os ciclistas.

Com isto, é a partir de dados e vídeos capturados por um *smartphone* que propomos um método que consiga avaliar situações perigosas para os ciclistas. Explorando o movimento numa sequência de imagens (fluxo óptico) conseguimos estimar o Foco de Expansão e usamos este Foco para definir áreas de perigo na imagem. Depois, usando estas regiões é criado um descritor de perigo com base nos diferentes objectos detectados na imagem para uma dada situação. Deste modo, o noso método permite definir diferentes critérios de perigo (Ocupação de via, Proximidade) com base no descritor de perigo criado.

Finalmente, testamos o nosso método de avaliação de perigo em imagens capturadas pelo *smartphone* e concluímos com resultados muitos promissores.

Palavras-chave

Ciclista, Análise de Risco, Foco de Expansão, Fluxo Óptico, Dados de *Smartphone*,Visão por Computador

Contents

List of Figures

List of Tables

Glossary

App	Application
FOE	Focus of Expansion
FPS	Frames per second
GPS	Global Positioning System
OF	Optical Flow
R-CNN	Region Convolutional Neural Network
NN	Neural Network

Nomenclature

I_x, I_y, I_t	Partial derivatives for the optical flow equations in respect to x, y and t
q_h	Pixel h in the image
V_x, V_y	Velocity vector in respect to x and y
$\tilde{x}$	Estimation of the FOE as a result of a optimization process
v_i	Optical flow vector i
L_i	Line that is the extension of optical flow v_i
$f(x, L_i)$	Distance between point x and line L_i
m_i	Magnitude weight for optical flow vector i
d_i	Magnitude of optical flow i
$\bar{d}$	Average of magnitudes of optical flows
d_{max}	Maximum magnitude of optical flows
o_i	Overall object associated weight for optical flow i
$\hat{o}_{i,k}$	Object associated weight for optical flow i for object $object_k$
$object_k$	Object k
s_k	Confidence score given by the Neural Network to object k
w_i	Overall weight (magnitude and object associated) for optical flow i
x_{FOE}	Coordinates on the image of the Focus of Expansion
$L_\delta(a)$	Huber Loss function
$\widetilde{x_j}$	Estimation of the FOE for previous frame j
x_t	Weighted average of the previous t FOEs
d_t	Risk descriptor for time t
d_t^l	Risk descriptor for time t in sub-region l
$r_t^{l,k}$	Risk of each object k in time t and sub-region l
α_k	Factor of risk for the type of object k
γ_l	Factor of risk for region and sub-region l
$a_t^{l,k}$	Area of object k on sub-region l at time t
b_t^l	Area of sub-region l on time t
φ_l	Factor of risk for region l
ψ_l	Factor of risk for sub-region l

Chapter 1

Introduction

1.1 Cycling in Urban Areas

Most environmental and mobility problems in cities across developed countries comes from prevalent car usage. It is paramount that we find ways to combat this growing trend which causes high pollution levels, health problems and accidents. By fighting against this tendency, we can decrease the economic dependence on fuel which may lead to decreasing the society's susceptibility to an energy crisis, as it was done in the Netherlands to fight the oil crisis of 1970 by investing in cycling infrastructure [1].

Furthermore, active commuting is one potential way to combat this growing dependence on fossil fuels. Walking and cycling have been shown to have cardiovascular benefits when practiced for around 30 minutes per day, for 5 days a week [2]. A study done by Oja *et al.* [3] showed that this type of commuting does enhance cardiorespiratory and metabolic fitness of sedentary adults. It also concluded that cycling is more effective than walking due to its slightly higher intensity.

On the other hand, a more recent trend in Europe is to promote tourism through what is called cycle tourism, as tourism represent the third largest economic activity in the EU. Since 2010, several projects have been undertaken to promote cycling as a way to discover regions and from 2007 to 2013 600 million euros have been spent for creating cycle infrastructures that contribute to this type of tourism [4]. The European Cyclists' Federation currently is responsible for managing a European project called EuroVelo, which incorporates more than $45.000\ km$ in cycle routes spread across Europe in order to promote tourism [5]. Weston and Mota [6] summarise what are the principal implications that cycling would take in the sustainability of tourism.

Notwithstanding having such a big importance on the environment, health and economy, cycling has not yet seen such a growth in use as one would expect. The study conducted in Belgium by Vandenbulcke *et al.* [1], discretise the major determinants that contribute to a higher or lower bicycle use. These can be divided into Demographic and Socio-economic, Cultural and Societal and Environmental and Political determinants. They conclude that if properly implemented, policies that address these determinants can result in an increasing bike use. Issues like the lack of cycling routes, too high or too low traffic volumes and even low quality of traffic signalling can result in major deal breakers for newcomers in cycling. Thus, to promote cycling commuting these infrastructures need to be heavily thought out by city planners, otherwise the fear caused by roads and its vehicles can be too daunting for new cyclists, as enumerated by Pucher [7].

Factors like poor road or bike lane design, bad road signalling and behavioural characteristics are expected to influence the number of accidents that happen to cyclists. Accident statistics in the USA show that the highest fatality count was in 2015 since 1995, despite the injury count being down 10% from 2015 to 2014, [8]. In 2013, the City of Boston release a study [9], in which it describes ways to make roadways safer for vulnerable users by using the "six E's of bicycle planning: Engineering, Education, Enforcement, Encouragement, Evaluation, and Equity", and evaluating its running program for improving bike use. It also

states that through its investment in bike lanes, bike sharing programs and bike facilities improvements it has transformed the city into “becoming a world-class cycling city”. The City of New York have also released data [10], [11], where it shows the decrease in Cycling Risk (which they evaluate by the number of severe injuries and fatalities per cycling trips) by the implementation of protected bicycle corridors on its main avenues.

However, these studies that lead to policy changes and better urban cycling design are mostly done by taking into account the number of injuries, fatalities and accidents that happen in a certain location that involves bicyclists. Our approach targets the risk assessment of cyclists in urban areas using a data capture system that ultimately can identity risky or harmful areas before accidents happen and, therefore, prevent these accidents from ever happening at all.

1.2 Sensing: Getting data from smartphones

Today smartphones are cheap and widely used by most population in cities. In fact, more smartphones have been purchase than PCs since 2011 [12] and its market share on Connected Devices is expected to keep on increasing, achieving 87% of the worldwide smart connected device market jointly with tablets according to IDC [13]. Smartphone applications and their development are also very common nowadays. Statistically, Apple’s number of application in its store have grown by almost a factor of 10 since 2010 [14], and on Android’s Play Store an average of 50,000 apps are released every month [15].

Smartphones are also equipped with a variety of sensors which allow the capture of a multiplicity of acceleration and velocity signals, such as linear accelerations, gravitational accelerations and angular velocities. Other sensors allow the recording of audio (using the microphone) and video (using one of the available cameras). Another set of important information that can be gathered using this device is GPS data, which enables the identification of events in a geographical fashion. This variety of sensor driven data, aligned with the simplicity on developing applications, has led to a variety of available developed apps.

Given the wide usage of smartphones today, it is only natural that to study the risk associated to cycling we could use a smartphone to capture data and translate the collected data into risk factors. Several works have been conducted which start from the development of a smartphone application and use the captured data to perform studies in the activity of the users. Studies vary from human activity recognition to human driving behaviour.

In [16] a Context Pyramid with raw sensor data is used to estimate ubiquitous position, recognize motion and human behaviour. Mitchel *et al.* [17] use the smartphone accelerometers to automatically identify a sporting activity. And although Avci *et al.* [18] do a survey on activity recognition using inertial sensors in Wireless Sensor Networks and not inbuilt smartphone sensors, they enumerate a series of health and medical applications where these sensory data could provide assistance to patients with cognitive disorders,

child and elderly care and in rehabilitation. Also, work done in [19] tracks users' physical activities and provide feedback in order to foment healthier habits and lifestyle. Su et al. [20] does an overall view on different activity recognitions processes, focusing on data that support the main activity recognition algorithms.

In [21], smartphone's sensors (accelerometer, gyroscope, magnetometer, GPS, video) are used to characteristically divide driving as non-aggressive or aggressive. Eren *et al.* [22] use sensory data to obtain position, speed, acceleration and deceleration to estimate commuting safety by analysing driver behaviour. More work on analysing driver's behaviour have been done in in [23], where this analysis lead to a road anomaly detection system. Moreover, the sensory data gathered from smartphones can also be applied to fuel saving. As studied by Seraj *et al.* [24], the breakdown of the sensory data captured by their developed Android application and from sensors in the vehicle is able to help drivers reduce their fuel consumption.

Another set of sensory data that may lead to interesting results is GPS. The resulting data enables for a more on-site study, where we can check for data variability according to geographical and real world scenarios. Strauss *et al*., [25], use GPS-driven data to model the volume of cyclists in an area and assess injury risk throughout the road segments and intersections in the island of Montreal. Moreover, in [26] accelerometers are used to estimate the vehicle's speed and then sense the traffic volume at a location.

Previous work in [27], also analysed stress from ECG data by analysing the relation between stress level and the Inter-beat Intervals variation. However, it concluded that there was a misinterpretation between stressful events and high intensity intervals (e.g. when terrain was unlevelled), which needed to be classified separately using the corresponding videos captured by an action camera and GPS data. This led to improvements being made to the previously developed app in [27].

To sum up, we can see that most related work is done over the inertial sensors to estimate and identify certain activities or events. However, our work focus more on image processing and computer vision to detect and contextualize a set of events that may cause stress to the user. And although our improved developed application from [27] also records inertial, sound and GPS data, we take advantage of the information only available through the captured images.

1.3 Computer Vision on Risk Analysis

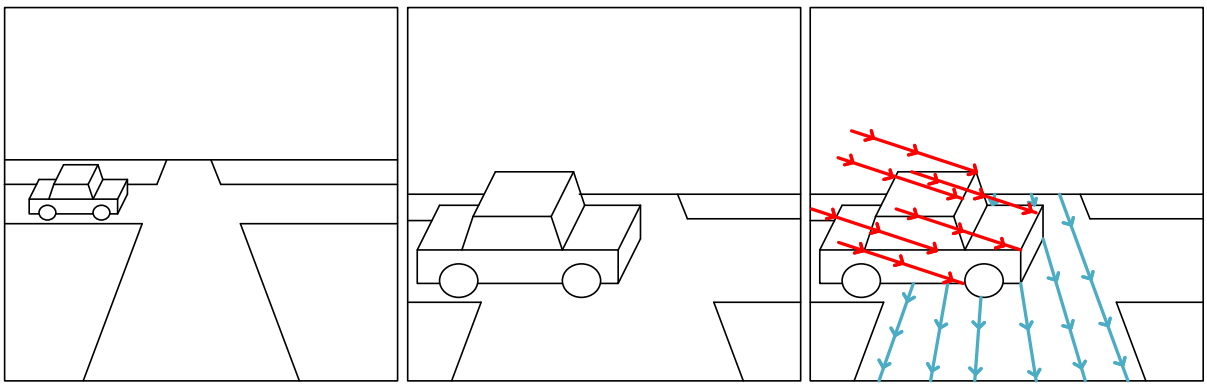
In order to assess risk from a video recorded on a bike, we come up with the following question: *What causes risk to cyclists and how can we judge it?*

In an urban scenario, there are several events that may lead to a cyclist being harmed like collision with other vehicles or pedestrians, speeding vehicles, running red lights/stop signals, misleading traffic signals, among many others. Thus, it is important to be able to distinguish what these objects or situations are.

As briefly mentioned above, our aim is to take advantage of images taken from a smartphone to assess dangerous situations to cyclers. Consequently, we need to divide the captured images into sections, depending on what information of risk they contain. So, one way to important feature must be the detection of different objects on a scene, like cars, buses or even people.

Several works have been conducted in the last years to improve object detection and segmentation. The current state of the art consists on convolutional neural networks which divide an image into regions or sub-images and try to detect objects on each one of the regions. To do this, each region is passed through a series of layers and then, if the score for the detected object is above a certain threshold, the region is said to contain that object. However, although this is a straightforward procedure, it is still computational heavy. In fact, the computational bottleneck in these object detection algorithms is still tied with the region proposals for the classification. Faster R-CNN [28], explores this by introducing a novel way to divide the image using a Region Proposal Network. Newer procedures, continue to explore this bottleneck and try to fasten even more the complete procedure. YOLO [29] presents the problem solution as a unique neural network pipeline, and thus increasing speed. SSD [30] goes even further by discretizing the output space into different aspect ratios and scales the making procedure adjustments to better match the object's bounding box.

However, it is not only important to find objects and their locations on an image, it also important to figure out it the discovered object presents a threat to the cyclist. In order to do this, we also need to find the direction the user is moving towards. This motion produced by the cyclist produces a particular point in the image called the Focus of Expansion (FOE), which represents the convergence of all motion in the image in a single point [31]. Although each object movement in the image can produce its own FOE, we are only interested in the one made by the user's movement, as it is the risk to the cycler we are studying. The FOE can be discovered by calculating an image's optical flow (OF). The optical flow is used to detect motion



(a) (b) (c)

Figure 1.3.1 – Left and middle images represent the motion between two captured images. On the right, the optical flow (from the car and the main road) resulting from the previous images is shown.

between a series of similar images. The optical flow is represented as vectors that depict the amount and position of the motion that happened between two images.

Figure 1.3.1 shows how the OF can be seen in a road scenario. Here we can see two distinct types of optical flow: one produced by our movement in relation to the road (in blue); and the second which translates the motion of the car itself but with a contribution of our motion as well (in red).

Although optical flow has been studied for many years now, it still presents an open-problem due to its calculation complexity caused by large motions, texture-less regions, shadows, reflections and many other factors. Plus, it is computational demanding, as one needs to find an appropriate motion match across the whole image. The most prominent and widely algorithm to calculate the sparse optical flow is the Lucas and Kanade pyramidal algorithm [32]. More recent works present the problem of solving dense optical flow problems to better get a denser and accurate motion estimation, [33], [34].

To sum up, we do an analysis the risk of a given situation by examining to where the user is moving and whether there is any object that may present as a treat between the user and its destination.

1.4 Outline of the Thesis

In this thesis, we focus on detecting and contextualizing stressful events to a cycler. This study is made based on image processing from image gathered from a smartphone attached to a bicycle's handlebar. These events are based on objects in the proximity of the biker, or objects that are, in some way, an obstacle in the route the cyclist is taking.

To detect events that may cause risk to the cycler, we divide our work in four main steps, as shown in Figure 1.4.1. We calculate the Focus of Expansion first with the Optical Flow between two image and second with detected objects in the image. Finally, we use the FOE and detected objects to calculate a risk associated to the presented situation.

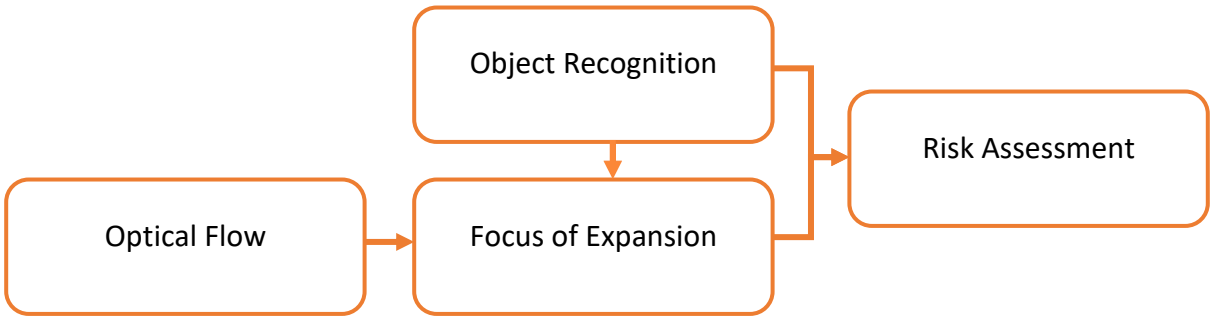


Figure 1.4.1 – Workflow of the work developed.

This thesis is separated in 5 chapters, including this introductory chapter.

On Chapter 2, the data acquisition process is presented, as well as the organization of the recorded data on the server.

Following, is Chapter 3, which describes how the risk assessment of stressful situation is made, starting from the discovery of the optical flow from the images and then discovering the focus of expansion associated with them.

Chapter 4 presents some results from the processing made to some of the acquired images from the database.

Finally, Chapter 5 wraps up the work developed and suggests some future work.

1.5 Contributions of the Thesis

With the work developed on this thesis we improved the previously available Android app. This new app is now able to capture new sets of data like video and sound acquisition, new GPS information (speed and measurement error) and a new set of acceleration signals, as well as the smartphone rotation matrix and orientation. New work was also developed on the backend server, developing an automatically data storage location for the data that is uploaded from the smartphone's app.

Another contribution is the detection of stressful or risky events and estimation of the Focus of Expansion purely based on image processing.

The database of all the recorded data (sensory data, videos and sound recording) is made available, as well as all the processing code used in this thesis.

Chapter 2

Data Acquisition

In this chapter, we will describe how and what data is acquired by the developed system, focusing on the types of data that are acquired and how the data is gathered by the system and then uploaded to a central location where we can access it and then use it in processing.

2.1 Mobility in Cities: The Application

To gather data from the smartphone an Android app was used. This app was built over the previously made app in [27]. This previously developed smartphone app enabled the recording of simple acceleration, velocity and GPS signals. The improvements made it possible to record another sets of acceleration and velocity signals, more GPS information than the one available before, and most important, it makes it possible to also record audio and video directly from the app. Moreover, it is now possible to upload the recorded data directly to our database if an internet connection is available to the smartphone.

Specifically, the app is able to acquire the following data from the smartphone:

- Date: the date and time of the measurement, [$year, month, day, hour, minute, seconds$];
- Acceleration: get the smartphone's applied force of acceleration along the 3 axes (x, y and z), [m/s^2];
- Linear Acceleration: get the smartphone's acceleration along the 3 axes (x, y and z), excluding any acceleration caused by gravity, [m/s^2];
- Gravity Acceleration: get the smartphone's acceleration along the 3 axes (x, y and z), caused strictly by gravity, [m/s^2];
- Angular Velocity: get the angular velocity around the 3 axes (x, y and z), [$rads/s$];
- Orientation: get the degrees of rotation that the smartphone makes around the three axis (x, y and z), [°];
- Rotation Vector: used to measure the vector of rotation of the device, [$unitless$];
- Step Counter: some smartphones got an inbuilt step counter, [$number\ of\ steps$]. However because not all tested smartphones got this sensor, the data resulting from this sensor was disregarded for now;
- Significant Motion: some smartphones got an inbuilt triggered significant motion sensor, which is responsible for automatically detect what type is motion is present (walking, bicycling or driving), [$0\ or\ 1$]. Again, because not all tested smartphones have this sensor, the data resulting from this sensor was disregarded for now;
- GPS coordinates, velocity and GPS accuraccy: get the current latitude, longitude, speed at which the phone is moving and the error associated with the coordinates, [$°, °, m/s, m$];

- Video: if enabled, get image recording from the smartphone's camera, and;
- Audio: if enabled, get audio recording from the smartphone's microphone.

The acceleration and angular velocities axis x, y and z are depicted in Figure 2.1.1 for each smartphone.

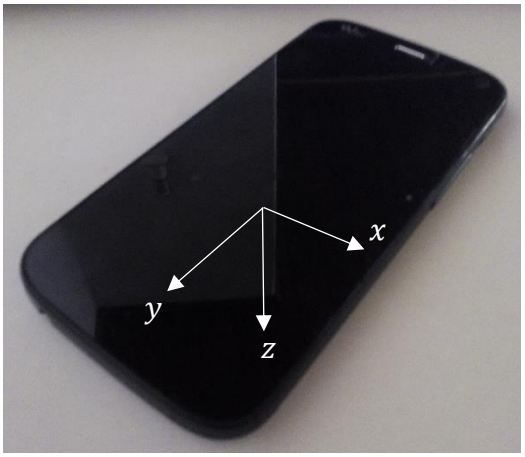


Figure 2.1.1 - Representation of the smartphone axes.

Given that different smartphones were used to acquire data, the video outputs are different because each smartphone has a different camera hardware and therefore, the resolution and quality of the resulting video for each phone is different.

Additionally, some personal information is also asked when the user first uses the application, such as:

- Username: name under which the user is registered;
- Password: password associated with the username;
- Gender: gender of the user;
- Age: the user's age in an interval manner (e.g. 15-24);
- Biking experience: the user's biking skill: Little, Some experience or Professional;
- Bike suspension: whether the bike used has a suspension or not.

This information is used to create an account on the server that stores the captured data, but this is further described below.

2.2 Data on the server: Backend

There are two ways for the app to communicate with the server: either registering or uploading a data file to it. In the former, the user's personal information is sent to the server to create a unique account. In the latter, the data files are uploaded to the server.

When registering, the user's personal information is used to create a user folder on the server associated to the user that is recording the data, so that all recorded data from the same user is saved under the same directory. This allows for a more direct link between events that may be associated with reactions (biking experience) or smoother accelerations (the suspension on the bike may attenuate steeper acceleration transitions). Note that usernames cannot be repeated, so in the event that a user tries to register with an already used username, it is asked the user to register under a different username. In case the username and password match with an already registered pair, then it is considered that the user reinstalled the app, and it is only logging in, and thus allow the user to continue using the app.

The second communication type is uploading a file. By doing this, it allows for a faster, simpler and more organised way to get access to all the data from every user, instead of having to copy all recorded files via a USB connection between the smartphone and the computer.

Whenever a file is selected and then uploaded to the server in the app, it sends the file, the user that is sending it, its password and a simple authentication password. In the server, first we check if the authentication password is valid. If not, we simply ignore the request received. Otherwise we then check if it corresponds to an already registered user using the sent user-password pair. If this match to the ones registered on the server, then the uploaded file is saved under the user's folder on the server, where we keep its name, so that we have a time registry of when the file was recorded.

If the uploaded file was of the sensor type (".txt"), we automatically generate a geographical map of the user's trip, so that the user can have some feedback of how we are exploring the data he sent. In the future, we also would like to generate other type of information and give it to the user, for example how many high-risk event happened during the trip, where and what caused them, or a map of the speed at which the user travelled.

Chapter 3

Risk Assessment using the Focus of Expansion

In this chapter, we describe how we use optical flow to discover the focus of expansion in a set of images and further use the discovered focus of expansion to signal possible stress or risky events in a set of images.

3.1 Overview

Given our problem of risk assessment to cyclists, our work consists in evaluating the trajectory of the cyclist and estimate the amount of risk of a certain scenario.

Associated with risk is, inevitably, the direction the bicyclist is taking. It is evident that a person located in front of the cycler presents a higher threat than a parked car in the side of the road, in the case that the cyclist is moving parallel to the road. So, it is paramount to first find the movement of the cycler. In an image, this direction corresponds to the Focus of Expansion. This point, in the image frame, represents the convergence of the depicted motion in the image, where such motion is given by the optical flow. Second, it is also vital to detect objects in the image, as their location on the image may present different levels of risk. Figure 3.1.1 present a summary of our risk assessment framework.

Figure 3.1.1 – Risk assessment framework: The estimation of the Focus of Expansion (red point) allows a representation of different level risk areas in the image. Risk is evaluated taking into account the objects in the image (blue rectangles) and the path of the cyclist (red area in the image).

So, as mentioned, our risk assessment framework is based on the estimation of the Focus of Expansion and the relation between objects in the image frame and this point. Again, the Focus of Expansion is the

point in the image that maps the convergence of the cyclist's motion vectors in the image. To compute such an important point from the optical flow we consider different properties of the image, which we will describe next.

3.2 Static Scene

Let's first consider the static scene case where we have a certain amount of motion described only by the cyclist between two consecutive video frames (and no other motion coming from other moving objects in the image). In this case, the optical flow calculated has no interference from motions related to moving objects. As such, the motion vectors point towards only one location in the image, and the intersection of all these vectors result in the Focus of Expansion.

To compute the optical flow between two frames needed to estimate the Focus of Expansion, which is represented as a vector between two corresponding points in the two frames, one can use the Lucas-Kanade method. This method proposes to solve the optical flow equations (1) with partial derivatives I_x, I_y and I_t for x, y and t respectively, for all pixels q_h, with $h = 1, \dots, n$, around point p, for the velocity vector $V = [V_x, V_y]$.

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ &\vdots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned} \tag{1}$$

This problem can be written in matrix form as

$$Av = b\,, \tag{2}$$

where

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \text{ and } b = \begin{bmatrix} -I_t(q_1) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{3}$$

To solve this problem, the Lucas-Kanade method obtains a solution using the Least Squares principle (4).

$$v = (A^T A)^{-1} A^T b \tag{4}$$

Given the motion vectors in the image v_i, with $i = 0, \dots, N$, one can then find the Focus of Expansion as the intersection of all motion vectors. In other words, one must find the point which is closest to all lines L_i which are the extension of each optical flow vector v_i. Let $x \in \mathbb{R}^2$ be a point in the image frame and each line L_i

be mapped in parametric form as $L_i = \{p_i + v_i t\}$, with $p_i, v_i \in \mathbb{R}^2$. The distance between point x and L_i can be seen as $f(x, L_i) = \left|(x - p_i) - \frac{(x-p_i)(v_i)}{|v_i|^2}(v_i)\right|$. Assuming some error in the calculations of the optical flow vectors (due to the window size or level or pyramids used), the point $\tilde{x}$, which is the point closest to all lines L_i, can be computed using the following optimization problem:

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} f(x, L_i) \tag{5}$$

So, assuming *a priori* that the optical flow computed across the image has some error associated in its calculations, we choose to weigh the computed optical flow vectors. This way we can distinguish some better vectors over others. Because velocity vectors depend on the amount of motion that is done on a certain area of the image, one way to do this weighting is to look at vectors magnitudes. So, vectors which are closer to the previous frame's FOE are smaller than the vectors farthest from the FOE. So, we assign weights regarding the magnitude of vectors according to their distance to the FOE. Specifically, we divide the image into 4 concentric circles with different radius and with centre on the previously found FOE in the last frame. We then look at the distribution of magnitudes in each annulus (formed by a circle minus its inner circles) and on the inner most circle and compute the mean of the magnitudes for each one of the 4 areas. We divide the vectors into different level bins (similar to what is done in [35]) according to its magnitude and how close it is to the mean of the area they are in. This way, we can discretise the weights m_i we assign to each vector v_i. We assign m_i in accordance to (6). Additionally, if the magnitude is below a certain threshold we assign it the lowest m_i score possible ($m_i = 0.1$). Figure 3.2.1 shows a representation of this.

$$m_i = \begin{cases} 0.10 & if\ 0 < \log(d_i) < \frac{1}{3}\log(\bar{d}), \\ 0.75 & if\ \frac{1}{3}\log(\bar{d}) < \log(d_i) < \frac{1}{2}\log(\bar{d})\ or\ \frac{5}{3}\log(\bar{d}) < \log(d_i) < \log(d_{max}), \\ 1.00 & otherwise, \end{cases} \tag{6}$$

where, d_i is the magnitude of vector v_i, $\bar{d}$ the mean of magnitudes in the area the vector is in and d_{max} the maximum magnitude found in the area the vector is in.

This way, we can re-map our problem to use the magnitude weights m_i as:

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} m_i \cdot f(x, L_i) \tag{7}$$

Solving this new optimization problem will result in a better estimation of the FOE because we can better choose what vectors weigh more in the FOE estimation.

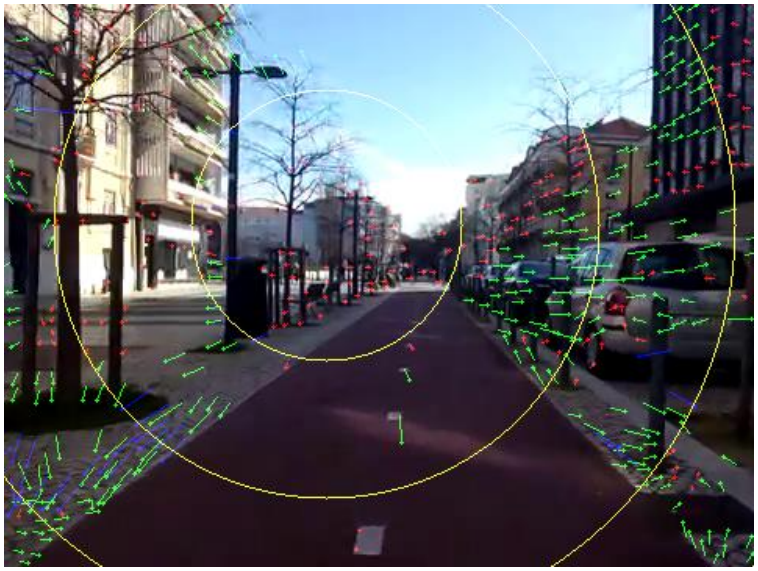


Figure 3.2.1 - In yellow are shown the limits of the concentric circles used for the calculation of the magnitude weights m_i. Green vectors represent m_i=1, blue vectors m_i=0.75 and vectors in red have a m_i=0.1.

3.3 Dynamic Scene

However, static scenes are rare in our scenarios. In fact, in most instances there are other moving objects in the scene like cars, people on the sidewalk or other bikes and thus it creates what we call the dynamic scene.

So, we perform another type of weighting, considering the objects in the scene and with the knowledge that their movement may disrupt the calculations of the optical flow. Thus, we first must detect all the interesting objects in the scene (i.e. cars, people, motorcycles, bikes and buses) and assess how they contribute to the OF miscalculations.

To detect objects in a scene, we feed the Faster R-CNN [28] with the image we are analysing in parallel. This object detector has proven to do well with objects that we are interested in detecting, namely, cars, buses, motorbikes, bicycles and people. The R-CNN (hereon now named NN for simplification), was trained using the PASCAL VOC 2007 [36] dataset, which contains the above-mentioned classes of objects, among others, which we discard in our analysis. The NN outputs the object location (under a bounding box format), its class and a score of confidence on the detection (which is an interval of $[0,1]$). Figure 3.3.2 depicts the output of the NN for a given image frame.

So, each OF vector is checked whether it is calculated on an object (the vector coordinates are checked whether they are inside any object's bounding box). Again, this is done because these objects may present a motion in a direction that is different from the one that the cyclist is taking and thus inducing error in the

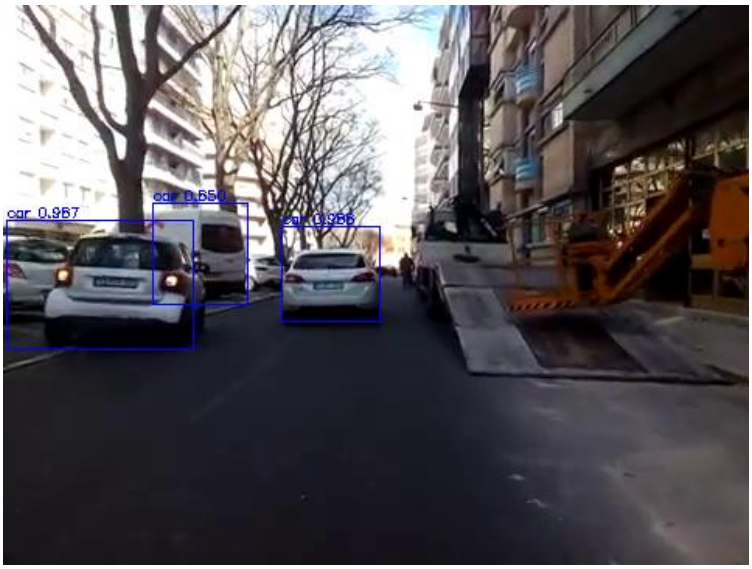


Figure 3.3.2 – Output of the NN showing the objects bounding box, class and confidence score.

FOE estimation. Therefore, we weight these vectors that correspond to objects, giving each vector v_i a weight o_i. Because the NN can output overlapped objects (refer to Figure 3.3.2), we decide that for vectors which correspond to overlapped objects in the image we calculate its weight with a contribution of each overlapped object. Thus, weights o_i can be calculated with:

$$o_i = \prod_{k=object_0}^{object_K} e^{-s_k}, \tag{8}$$

where s_k is the confidence score outputted by the NN to a given $object_k$, and k is the set of all overlapping objects.

In the case that for vector v_i there is only one object, (8) can be simplified to:

$$o_i = e^{-s_k}, \tag{9}$$

This weight reflects the confidence of the detection by considering the score given by the NN and it penalizes objects that have a high confidence of being objects. Note that the case where there is no object detected ($s_k = 0$), the weight of that vector o_i corresponds to its maximum value ($o_i = 1$). Figure 3.3.3 shows the vectors in accordance to the confidence score s_k.

By imposing these weights on the vectors, we can better estimate how a vector contributes to the estimation of the FOE. Thus, the optimization problem for the estimation of the Focus of Expansion is given by:

Figure 3.3.3 – Object associated weights given the confidence score outputted by the NN. Image (b) masks the vectors with the confidence score of image (a).

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} o_i \cdot f(x, L_i) \tag{10}$$

This new optimization problem will result in a better estimation of the FOE considering the object associated weights.

3.4 Focus of Expansion

In order to estimate the Focus of Expansion on a given frame we take the previously calculated optical flow vectors and their weights and try to find the point in the image to where they all converge. To do this is the same as trying to solve an optimization problem which tries to minimize the distance between the FOE and all the lines that are mapped as the extension of the optical flow vectors.

Furthermore, we are able to use the previously calculated weights (both magnitude weights and object presence weights) to better estimate which vectors are better used in the optimization and thus view the impact of the vector weighting. There are several methods that can be used to compute the FOE. However, we chose to solve the optimization problem using the Huber Loss distance as a distance metric.

The Huber Loss function works as a kind of intermediate between the Least Squares Distance and the Manhattan Distance, as it is quadratic for small values and linear for larger values. This way, outliers contribute linearly (similar to the l_1-norm), whereas values close to the desired value contribute more.

This problem can be formulated as in (11) disregarding all weights explained before.

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}_\delta(f(x, L_i)) \tag{11}$$

Considering the magnitude weights and object associated weight, we can formulate this optimization as:

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}_\delta(w_i \cdot f(x, L_i)) \,, \tag{12}$$

where $\mathcal{L}_\delta(a)$ is the Huber Loss Function, as defined in

$$\mathcal{L}_\delta(a) = \begin{cases} \frac{1}{2}a^2 & for\ |a| < \delta, \\ \delta(|a| - \frac{1}{2}\delta) & otherwise. \end{cases} \tag{13}$$

and weights w_i of a given OF vector v_i can be calculated using (14). Note that because both m_i and o_i vary in the interval $[0,1]$, also w_i varies in this interval.

$$w_i = m_i \cdot o_i \tag{14}$$

where m_i is the magnitude weight and o_i the object associated weight for vector v_i explained before.

A comparison between the Huber Loss Distance optimization and other FOE estimation methods can be seen in Appendix B.

3.5 Iterative Object Weights Refinement

However, given our scenario, one must take in consideration one more aspect: although most objects in a scene are non-static, they can be static, i.e., if a car is parked on the side or if a person is standing still on the sidewalk, they can be thought-out as static objects, as they do not contribute with any different motion to the one the cyclist is taking. Thus, this re-weighting of the object associated weights is performed taking this in consideration.

So, we perform an iterative process where each optical flow vector which is calculated on an object is re-weighted (only on the o_i weight) according to its direction. That is, if a vector is pointing away from the $\tilde{x}$

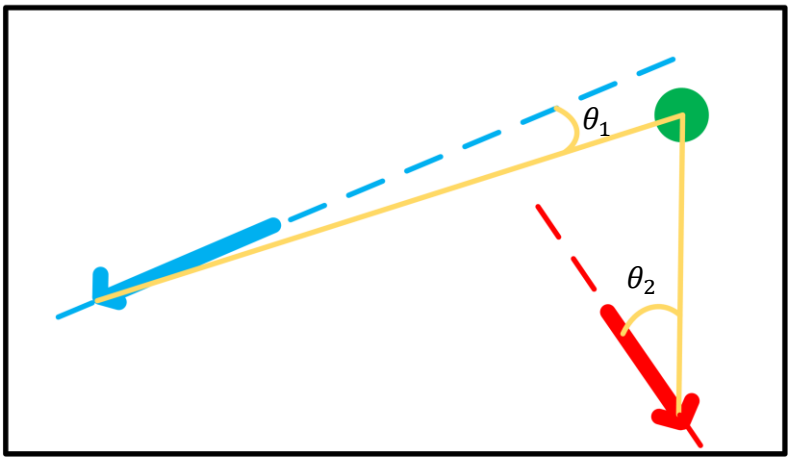


Figure 3.5.1 – Consider the FOE in green. The OF vector in blue (with the line extension from the vector in dashed blue) is considered an inlier because $\theta_1 < 15°$, whereas the vector in red is evaluated as an outlier due to $\theta_2 > 15°$.

and its direction is in line with the FOE (see Figure 3.5.1), we consider that the object is not being affected by the object, and thus we maximize $o_i = 1$. However, if the vector's direction is being affected by the object (e.g. pointing towards the FOE), we minimize its contribution and assign $o_i = 0$. Using this newly discovered weights, we rediscover the FOE using the same method as before (using the optimization problem with the newly found weights). This results in a freshly found FOE. We re-iterate this procedure until either the difference in FOEs found is below a threshold or until a maximum number of iterations have been performed.

Figure 3.5.2 shows an example where we have both static and non-static objects, and thus a re-weighting of the o_i weights is beneficial.

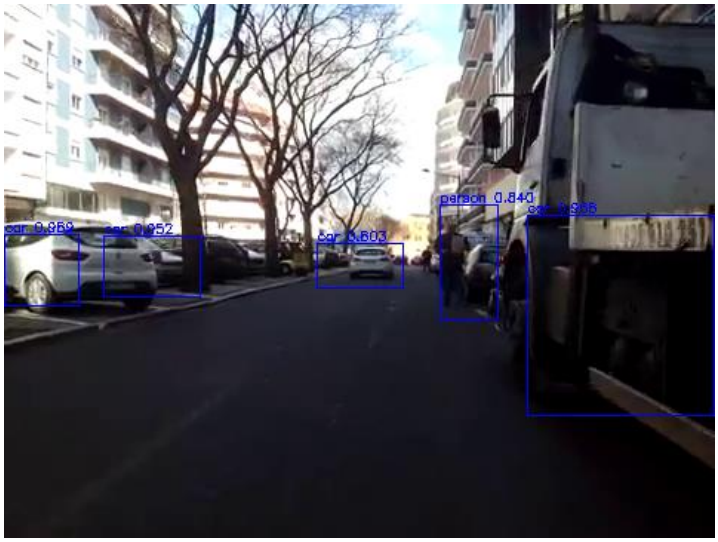


Figure 3.5.2 – Vectors calculated on the stopped van on the right side and on the cars on the left side of the image should have their object weights maximized, whereas the remaining objects on the image (car in the middle of the road and the person on the right) should have their object weight minimized.

3.6 Temporal Average of the Foci of Expansion

Following this, we apply a second step, which consists in performing a weighted average between the discovered FOE and the FOEs discovered in the previous M frames of the video. This is done because the video that was captured is not entirely stabilized, meaning that there are numerous oscillations resulting from the handling of the bicycle which affect the FOE. This weighted average is done as in (15).

$$x_t = \frac{\sum_{j=t-M}^{t} e^{-\tau(t-j)} \cdot \tilde{x}_j}{\sum_{j=t-M}^{t} e^{-\tau(t-j)}}, \tag{15}$$

where, x_t is the FOE estimate at instant time t, $\tilde{x}_j$ the FOE found using one of the methods mentioned above corresponding to instant time j and τ is the decay rate of the weights.

Figure 3.6.1 shows all estimations of the Foci of Expansion for the previous $M = 20$ frames and the final estimation of the Focus of Expansion considering this weighted average.

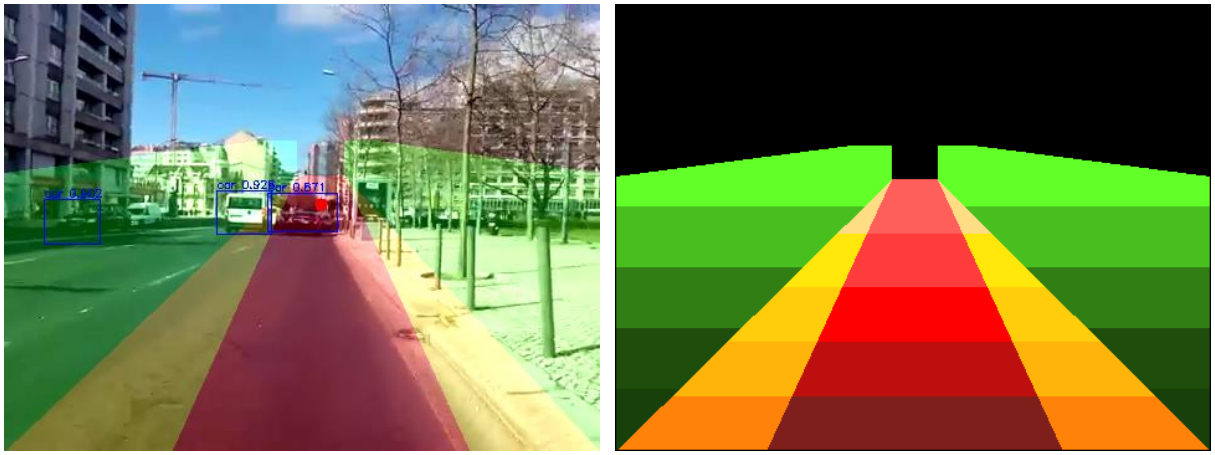
Having found the point which maps the direction of the cyclist's movement, the final step is to define areas in the image according to the FOE and assign risk levels to these areas.

Figure 3.6.1 – The weighted average of the previous and current FOE (small coloured points) and the result of this operation (large red point in the centre of the image).

3.7 Risk descriptor

The estimated FOE gives a sense of direction to where the cyclist is moving. With this, we are able to divide the image into five regions which give an idea of the path the cyclist is taking and the risk the user is subject to (see Figure 3.7.1.(a)). We further sub-divide each of the 5 areas on the image into 5 horizontal strips (see Figure 3.7.1.(b)), which are used to give a sense of distance (lower strips represent a shorter distance to the user). We use these defined areas in conjunction with the objects detected to create a risk descriptor which maps the motion and proximity risk to the cyclist. The risk descriptor at time t is a vector with 25 positions, with each value assessing a risk of a given sub-region in the image (see (16)).

$$d_t = [d_t^1 \quad \cdots \quad d_t^l \quad \cdots \quad d_t^{25}] \tag{16}$$



(a) (b)

Figure 3.7.1 – Regions and sub-regions used for the risk descriptor. In (a) the 5 regions show the colour-coded risk to the user (red is the path of the user and thus represents maximum risk, yellow the regions closest to the user's trajectory and green the furthest regions from the cyclist's path and minimum risk). Image (b) shows the horizontal strips resulting from the division of the 5 regions in (a).

To compute the value of the risk descriptor d_t^l on sub-region l we use (17), which maps the risk associated with each object k (α_k) and its confidence score given by the Neural Network (s_k), the risk associated with each region and sub-region (γ_l) and finally the ratio of the occupancy of the object in respect to the sub-region's area.

$$d_t^l = \sum_{k=object_0}^{object_K} r_t^{l,k}, \tag{17}$$

where the risk score of each object k in sub-region l ($r_t^{l,k}$) is

$$r_t^{l,k} = \alpha_k \cdot s_k \cdot \gamma_l \cdot \frac{a_t^{l,k}}{b_t^l}, \tag{18}$$

where $a_t^{l,k}$ is the object's k area in the sub-region l and b_t^l is the area of the sub-region l.

Using this descriptor, we consider several risk factors like the type of object (e.g. cars are riskier than people walking by) and its confidence score of being an object, the proximity of the object to the user and path trajectory (by using the region and sub-region level) and how big and close the object is (the ratio of areas used). Thus, we consider two major risk factors in the risk calculation: the trajectory and object proximity.

Given the aforementioned descriptor, we further propose to provide an encoding for the level of risk. This way, the risk is encoded into levels (1 to 3), providing a more informative and simpler risk assessment framework. We encompass this as a supervised classification problem, where we use the Earth Mover's Distance (EMD) [37] as a metric to compare different risk descriptors extracted from its images. While other distance metrics are used in image comparisons, like the use of the Euclidean distance for the comparison of two colour spaces, EMD is useful because it allows to compare the distance between two distributions and find the distance between the two. In our case, it is useful because it allows mapping the sub-regions neighbours and the existing symmetry in the image. This is done with the definition of the ground costs between the regions and sub-regions of the image, which may alter between regions of the distribution. This way, we are able to encapsulate in the ground costs definition the regions and sub-regions definitions mentioned above, as well as the existing symmetry in the image.

The definition of this ground distance matrix allows for different risk assessment models to be defined, depending on the distances between regions and sub-regions and the risk associated with each. Here, we propose to assess the risk in two criteria: path occupation and proximity. In Chapter 4 we further explore the two models used, as well as the definition of the ground distance matrices used for each assessment criteria.

Chapter 4

Results

In this chapter, we explore some of the results from the methods described in Chapter 3. We explore the influence of the weights and methods used in the FOE estimation, as well as the optical flow calculations. Furthermore, we analyse our two criteria for the cyclist's risk assessment. Finally, we describe some of the results got from the event classification.

4.1 Focus of Expansion

Most portable cameras in smartphones can record 30 frames-per-second (fps) videos, which, in our case, because the distance travelled between two consecutive frames is small, translates into a small motion between two successive frames. To avoid this, we perform downsampling of the video's frame rate, calculating the OF between 2 frames that are separated by 5 frames between them. Note that we could avoid this downsampling step and calculate the OF between two consecutive frames, we just had to make some adjustments to how the OF is computed and how we could use it to compute the Focus of Expansion.

To compute the OF between the selected images, we must first consider in what points to calculate it. Because we want to take advantage of the most points we can discover in the image and to have these points spread out evenly through the whole image we have a different way of discovering keypoints used in the optical flow calculations. Firstly, we start by applying a similar method as the one used in [38]. We start by dividing the image into 16 different sub-images, as shown in Figure 4.1.1. On each one, we apply a histogram equalization filter (CLAHE [39]), which improves contrast and edge characterisation. We then apply the Shi-Tomasi method [40] to detect important features in each one of the sixteen images. By doing so, we improve the feature detection on low texture areas of the whole image, in which points would be disregarded as low scoring features if the Shi-Tomasi algorithm would've been ran on the whole image. By doing this we also spread the discovered crucial points across the image, whereas before all the points could only present themselves in a section of the image and thus compromising the FOE estimation. A

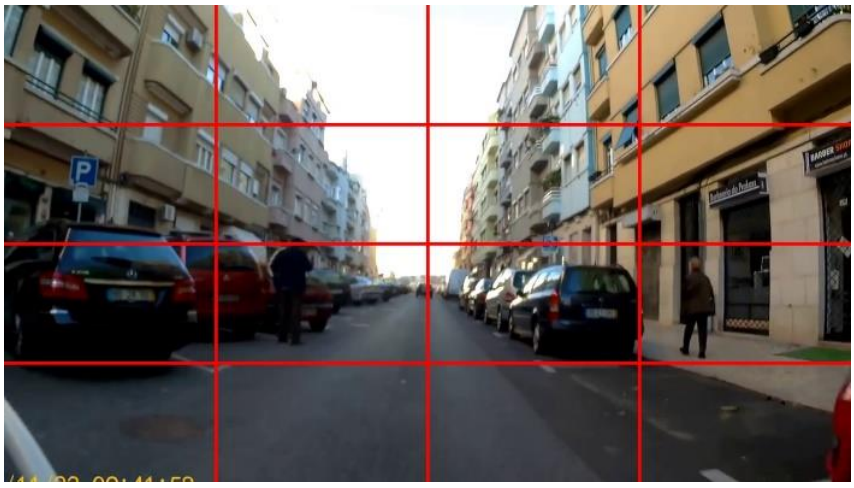


Figure 4.1.1 – Division of the original image into 16 sub-images.

comparison of the results using the Shi-Tomasi on the whole image and with the image separation can be seen in Figure 4.1.2.

(a)

(b)

Figure 4.1.2 - Features tracked using the Shi-Tomasi method [37] in red. In (a) the features are calculated on the whole image and in (b) features are calculated using the image separation into 16 sub-images.

Next, we calculate the optical flow on these discovered points using the Lukas-Kanade algorithm [32]. We adapt the window size and pyramid levels accordingly to the resolution and motion we expect. For example, given a resolution of 360p (480×360 pixels), a window of size 35×35 pixels is used, whereas, with a resolution of 1080p (1920×1080 pixels) a window size of 200×200 pixels is used. Also, given the 5 frame skip in a 30fps video mentioned above, usually a pyramid level of 1 is best used to fit the motion that is present between the two frames. Figure 4.1.3 shows the optical flow calculation between two frames with a frame skip of 5 between them and using a window size of 200×200 and a pyramid level of 1 as it is a 1080p image.

Weights used in the optimization process are calculated as stated before. For the optimization problem using the Huber Loss Distance ((11) and (12)) we used the Python Toolbox $sklearn$ and the module $HuberRegressor$ [41]. Thus, the result of the optimization step of the Huber Loss distance can be seen in Figure 4.1.4. In Figure 4.1.5 we present a heatmap with the distance between each line L_i and the solution for the optimization problem.

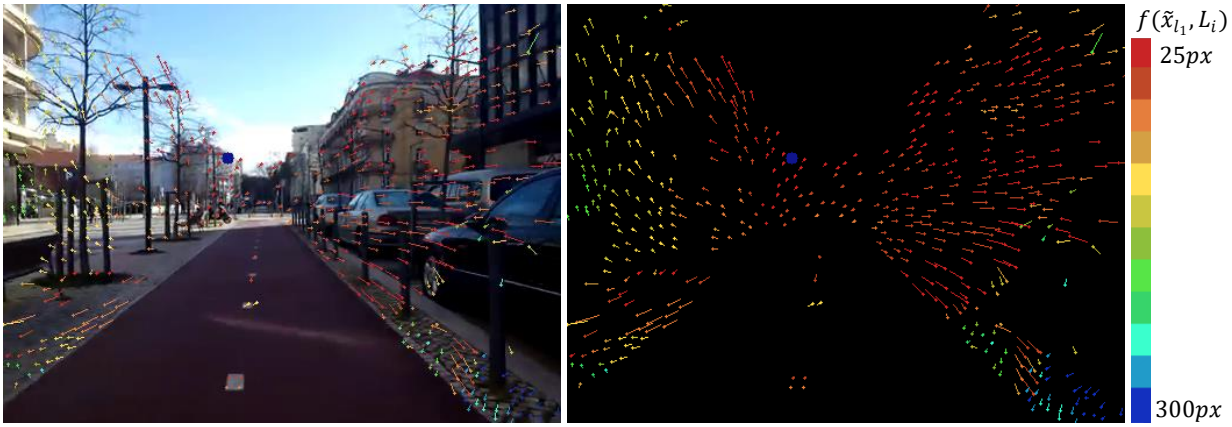
(a) (b)

(c)

Figure 4.1.3 - Optical flow calculation ((c)) between two frames ((a) and (b)).

(a) (b)

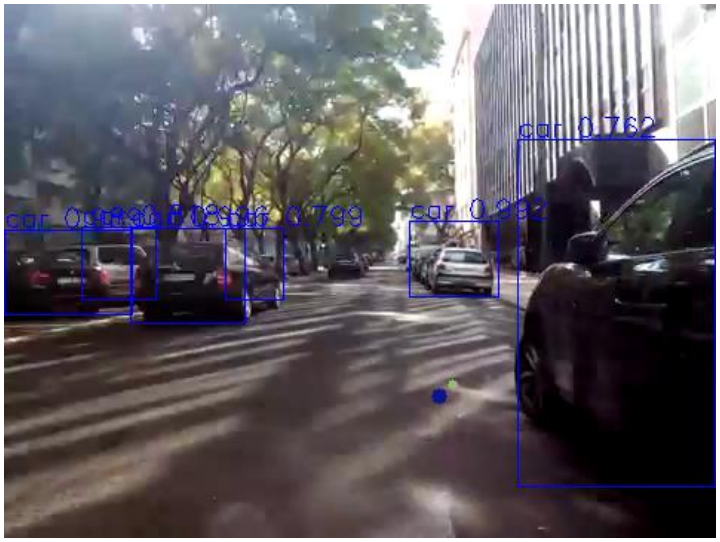
Figure 4.1.4 - Examples of the FOE estimation using the Huber Loss Distance optimization. The light blue point represents the solution without any weight consideration, whereas the dark blue point represents the optimization using weights.



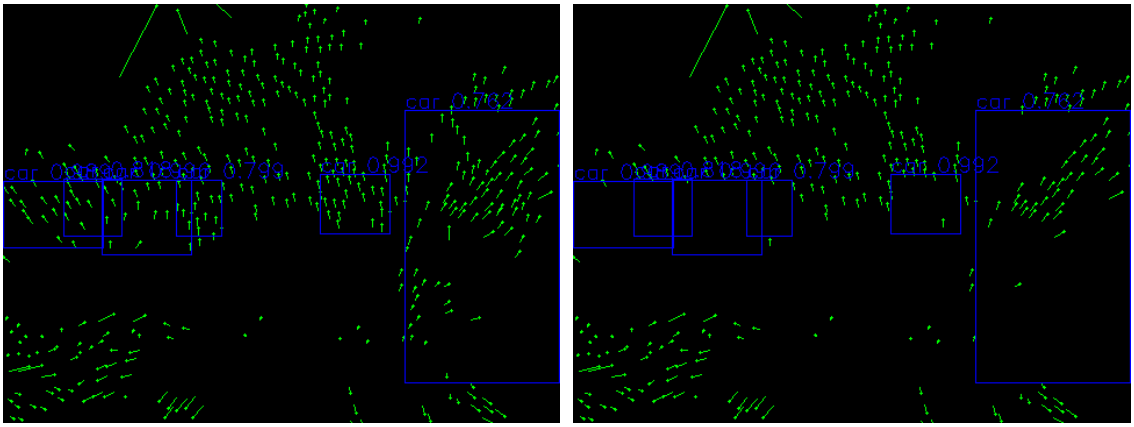
(a) (b)

Figure 4.1.5 – Heatmap of the distance between lines L_i and the estimation of the FOE given by the Huber Loss Distance optimization calculation.

After the estimation of the FOE, we proceed to refine the o_{v_i} weights. Figure 4.1.6 (a) presents the results of the iterations made to improve the Focus of Expansion and Figure 4.1.6 (b) and (c) show the resulting optical flow vector weights before and after the refinement is made. In fact, as seen in Figure 4.1.6, the moving objects (cars on the left side of the image) are much more affected than the static car on the right side of the image (which only loses a small portion of its optical flow vectors). In image frames where there are many objects that occupy a good portion of the image the final result of this process does vary a little from the initial estimate of the FOE. In images where there are little to no objects, this process offers almost no gain. A further observation is that additional iterations do not refine much more in regard to the previous iteration.



(a)



(b) (c)

Figure 4.1.6 – Results of the refinement of weights o_i. In (a), the point in blue is the estimate for the FOE given by the Huber Loss optimization. Points in pink and lime represent iterations 1 and 2, respectively, of this refinement procedure. Images (b) and (c) show the difference in vectors between the vectors used in the Huber Loss Optimization (for point in blue) and the final refinement of the weights (for point in lime).

Thus, this step does offer some improvements when there are some miscalculation of the optical flow resulting from moving objects in the image. This results in a more robust estimation of the FOE for miscalculated optical flow vectors.

However, sometimes we are faced with a problem regarding the object detector. As previously mentioned, we are using the Faster R-CNN detector with the PASCAL VOC 2007 training dataset. Because, this dataset contains detection classes which we are not interested in, the classification for the classes that we are interested in suffer some misclassification. This happens when classifying objects with wrong labels and sometimes, classifying areas of the image where there are no objects present. Figure 4.1.7 show some cases of object misclassification. This can greatly impact our developed tool, as a great part of our work is based on this object classification. Nevertheless, we still consider that although this object detection tool can be improved, we can still take advantage of the information it gives and how this information is used throughout the whole process.

Finally, the last step consists in performing the weighted average of the current estimate for the FOE and the estimates of the previous frames. Figure 4.1.8 shows the relation between the previously found FOEs (shown as colour points) and the result of the weighted average using $M = 20$. The colours of the FOEs also depict their weight for the estimate of the weighted average. This shows how useful the process of the temporal average is because as can be seen, the estimated FOE does vary a bit because of the natural swinging of the bicycle.

(a)

(b)

Figure 4.1.7 – Misclassification of objects in the image. In (a) a $person$ is classified in the middle of the cycling route and in (b) a car and $person$ are misclassified.

(a)

(b)

(c)

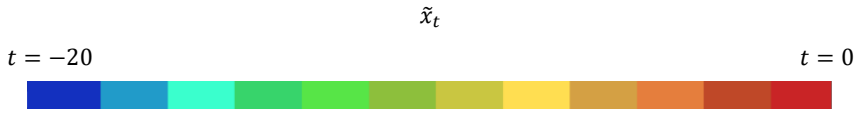


Figure 4.1.8 – Evolution of the previously found FOEs ($\tilde{x}_t$) from image (a)-(b)-(c). As one can inspect, from one image to the next, the FOE gets $t - 1$. Results are shown as a heatmap, with hotter colours representing a higher weight used in the weighted average as it represents more recent FOEs. The final result of this average is shown as a larger bright red point in the centre of the image.

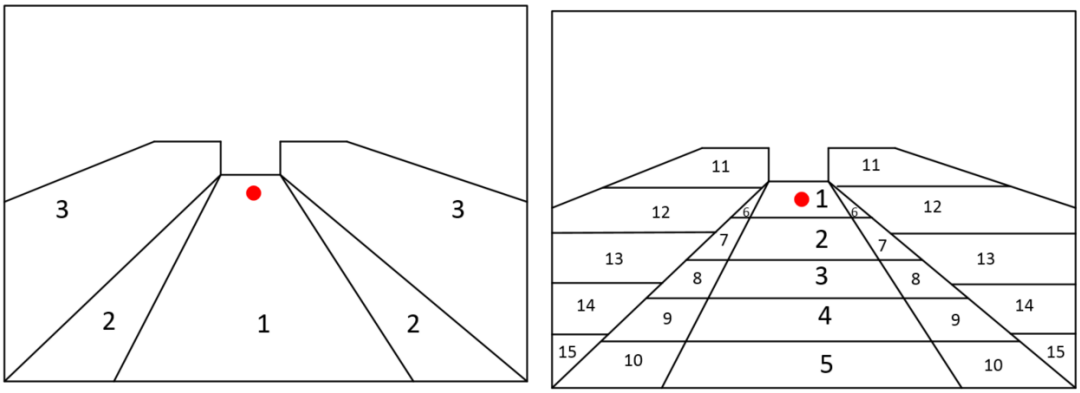
4.2 Risk Assessment

As previously mentioned, several risk assessment criteria can be defined using our framework. However, we decided to test two main risk criteria: Path Occupation and Proximity to the user.

For each of the criteria used we propose a 3-level risk. For the Path Occupation (see Figure 3.7.1) we set the risk as: 3 – the red region is occupied; 2 – the yellow region is occupied, and; 1 – the green region is the only one being occupied. For the Proximity to the user criteria we defined different regions to the ones used in the Path Occupation criteria, which better translates the real distance to the user. This way, we use the sub-regions designated before to form regions of growing semi-circles concentric with the user (see Figure 4.2.2) and again define the levels of risk as before. These regions are all constructed around the FOE. Taking Figure 4.2.1.(a) the estimated FOE is given as the red point and the regions are constructed as follows, where (x_{FOE}, y_{FOE}) are the coordinates of the FOE in the image and $width$ and $height$ are the width and height of the frame and the upper left corner of the image corresponds to $(0,0)$ and the bottom right corner to $(width, height)$.

- Region 1: Formed by the points at coordinates $(x_{FOE} + \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(x_{FOE} - \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(x_{FOE} + \frac{width}{4}, height)$, $(x_{FOE} - \frac{width}{4}, height)$;
- Region 2: Left triangle formed by: $(x_{FOE} - \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(0, height)$, $(x_{FOE} - \frac{width}{4}, height)$; and right triangle by $(x_{FOE} + \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(width, height)$, $(x_{FOE} + \frac{width}{4}, height)$;
- Region 3: Left polygon formed by: $(x_{FOE} - \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(0, height)$, $(x_{FOE} - \frac{width}{25}, \frac{7y_{FOE}}{8} - \frac{23height}{128})$, $(\frac{7x_{FOE}}{8} - \frac{27width}{200}, \frac{7y_{FOE}}{8} - \frac{23height}{128})$, $(0, y_{FOE} - \frac{height - y_{FOE}}{8})$; and right polygon formed by $(x_{FOE} + \frac{width}{25}, y_{FOE} - \frac{height}{16})$, $(width, height)$, $(x_{FOE} + \frac{width}{25}, \frac{7y_{FOE}}{8} - \frac{23height}{128})$, $(\frac{7x_{FOE}}{8} - \frac{14width}{200}, \frac{7y_{FOE}}{8} - \frac{23height}{128})$, $(width, y_{FOE} - \frac{height - y_{FOE}}{8})$;

Figure 4.2.1.(b) shows the division of the regions in (a) into 5 horizontal strips.



(a) (b)

Figure 4.2.1 – Construction of the regions and sub-regions of the frame given the estimate of the FOE (red point).

To test our risk assessment classifier, we created a training set by manually classifying around 240 image frames (with 80 images per each one of the three risk levels for each of the criteria), which were captured using the smartphone's app described in Chapter 2. Using this training set, we set to use the rule of thumb of 75% to 25% of training to test data ratio.

Additionally, because our classifying metric is based on EMD, we defined the 25×25 ground distance matrix for each one of the criteria used. Both matrices used promote the nearness of regions and sub-regions, the symmetry in the image and the borders of each region. This way, we set to add a factor (equal to 1) to distances between sub-regions and multiply by a factor (equal to 2) when transitioning from one region to the other (i.e., we add 1 to neighbouring sub-regions and we multiply the distance going from the red region to the yellow region by a factor of 2 and from the red region to the green region by a factor of 2). This way, the defined distance matrices used for the Path Occupation and for the Proximity criteria, which are presented in detail in Appendix B. In order to solve the EMD, we used the Python Toolbox $pyemd$ [42], [43].

For the object type α_l used in the calculation of the risk descriptor we consider that certain object present a bigger threat than others. This way, we define it as

$$\alpha_k = \begin{cases} 1.0 & if \text{ object } k \text{ is motorized (car, motorbike or bus)}, \\ 0.8 & if \text{ object } k \text{ is a bike}, \\ 0.6 & if \text{ object } k \text{ is a person} \end{cases} \tag{19}$$

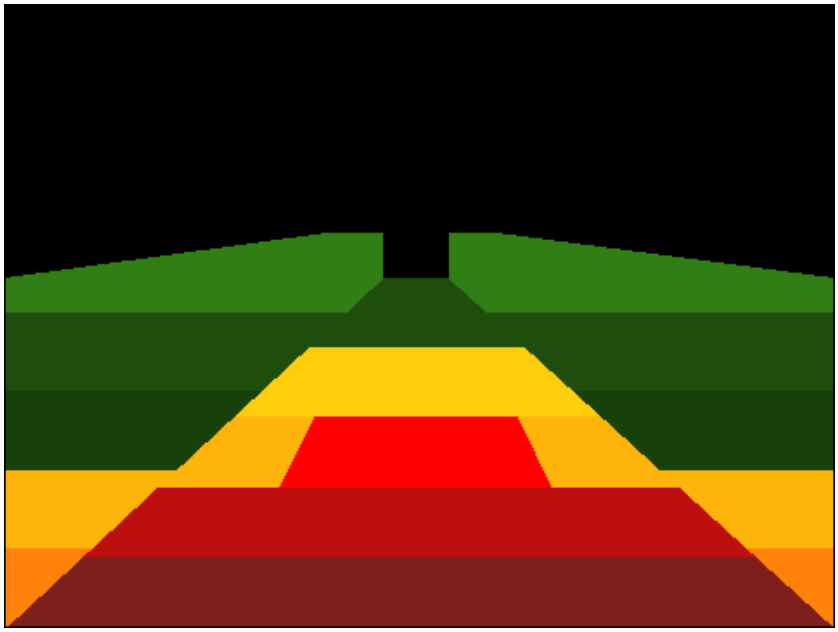


Figure 4.2.2 – Regions used for the Proximity criteria used in the risk assessment. The colours indicate the level of risk: red – highest risk level (3); yellow – intermediate risk level (2), and; green – lowest risk level (1).

Furthermore, the region and sub-region factor used in (18) is defined as

$$\gamma_l = \varphi_l \cdot \psi_l \,, \tag{20}$$

where φ_l is the region factor and ψ_l the sub-region factor and ψ_l grows linearly to the proximity to the user (i.e., the top-most sub-region corresponds to $\psi_l = 0.6$, the one bellow that $\psi_l = 0.7$, until the bottom-most sub-region that is equivalent to $\psi_l = 1$) and φ_l is defined as

$$\varphi_l = \begin{cases} 1.0 & if \text{ region } l \text{ is of risk level 3 (red area)}, \\ 0.75 & if \text{ region } l \text{ is of risk level 2 (yellow area)}, \\ 0.30 & if \text{ region } l \text{ is of risk level 1 (green area)}. \end{cases} \tag{21}$$

Moreover, for the objects bounding box, because we consider that every object is in contact with the ground, we, instead of using the bounding box given by the NN, use an alternative bounding box. We use the width of the box given by the NN and the height given by $\max(10\, pixels, 0.2 \cdot NN's\ bouding\ box\ height)$. Using

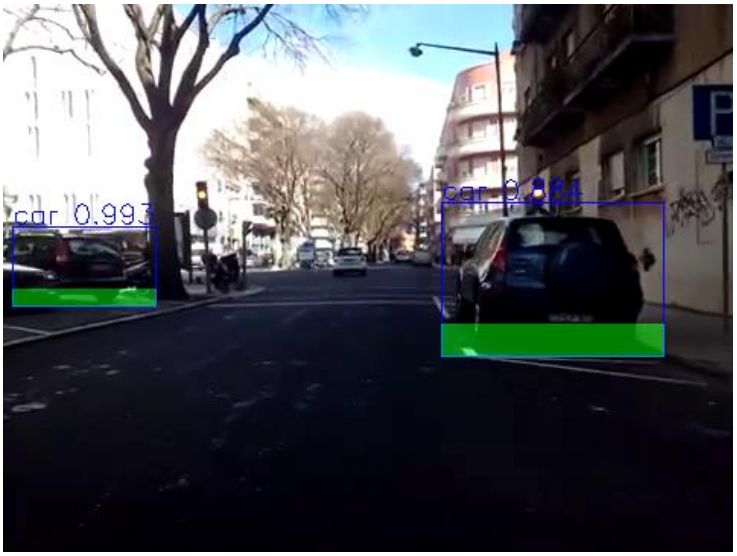
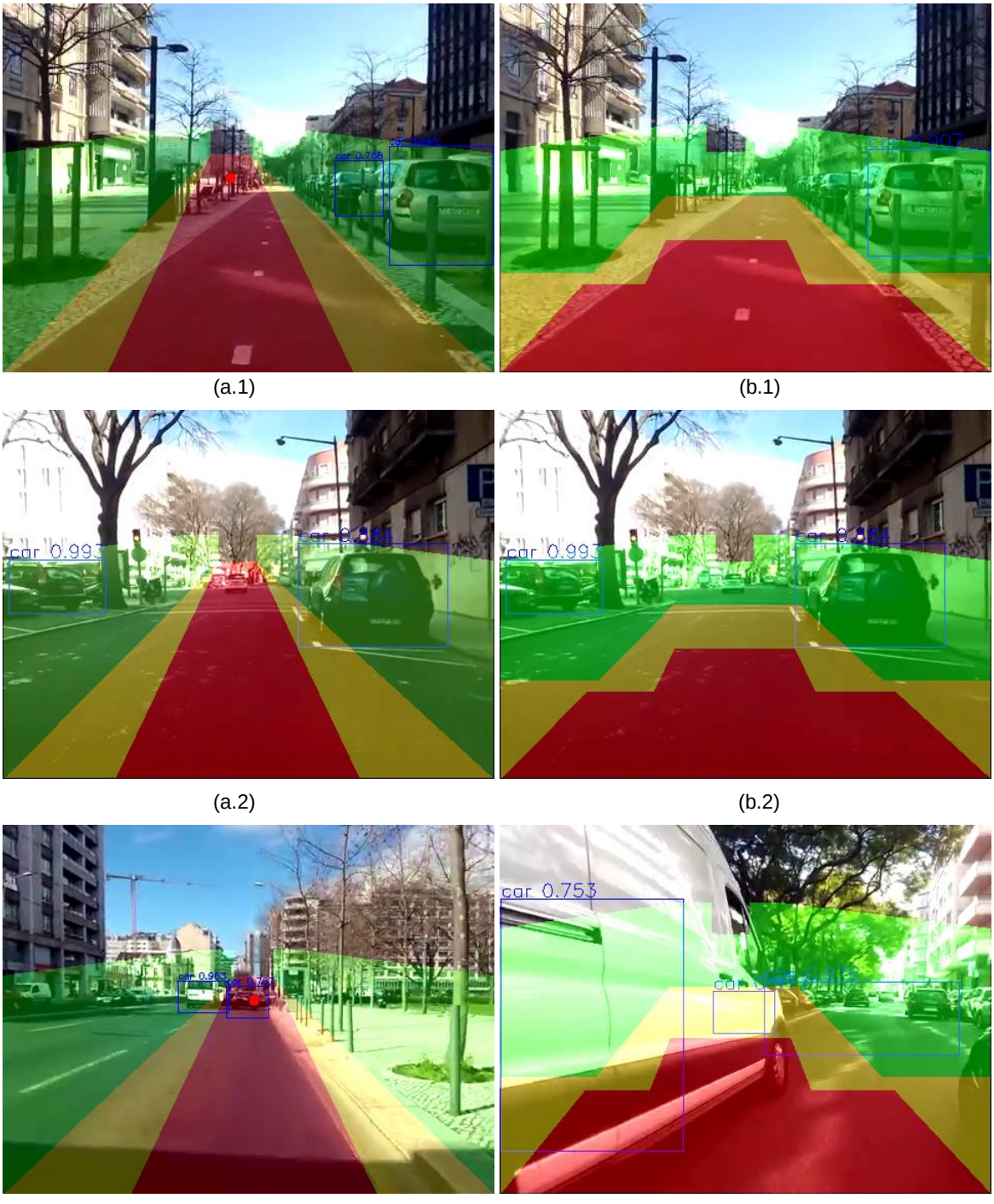


Figure 4.2.3 – Boxes of the objects. In blue, the bounding box given by the NN is shown, whereas in green it is shown the box of the alternative box used as a projection of the object on the ground.

this box better projects the object on the ground and in the risk zones (which correspond to regions on the ground). Figure 4.2.3 shows the used bounding boxes in relation to the ones outputted by the NN.



(a.1) (b.1)

(a.2) (b.2)

(a.3) (b.3)

Figure 4.2.4 – Risk Assessment framework. Images (a.1)-(a.3) represent the Path Occupation criteria, whereas images (b.1)-(b.3) correspond to the Proximity criteria. Images (1) correspond to risk level 1, (2) to risk level 2, and lastly, (3) to risk level 3.

Examples of situations for each level of risk for each of the two criteria used are shown in Figure 4.2.4.

Given all this, we ran our classifier for 60 images frames, composing our test data. We present our results as a confusing matrix in Table 1 for the Path Occupation classifier and in Table 2 for the Proximity classifier.

Table 1 – Confusion Matrix for the Path Occupation classifier.

		Predicted Class		
		1	2	3
True Class	1	80.0	20.0	0.0
	2	9.1	81.8	9.1
	3	0.0	25.0	75.0

Table 2 – Confusion Matrix for the Proximity classifier.

		Predicted Class		
		1	2	3
True Class	1	66.7	33.3	0.0
	2	10.7	82.1	7.2
	3	0.0	41.2	58.8

We conclude that there is no misclassifications between risk levels 1 and 3 and, as such, we have a good separation between these two classes. The accuracy for the Path Occupation classifier is relatively high, showing an error rate of around 20-25% for each class. However, for the Proximity classifier the results show some misclassification between risk levels 3 and 2, which we consider as a result of objects being too close to the limits of both red and yellow regions during the labelling phase, and thus resulting in some erroneous classifications. Likewise, because we have discretised our risk levels throughout the defined regions, i.e., the risk levels are not continuous, it is expected that there are some risk misclassifications when objects are positioned near or on top of the boundaries of each zone.

4.3 Elementary Event Classification

As a side effect of the discovery of the Focus of Expansion we are also able to discover elementary events as it was done in [27]. However, a different approach is done and thus a direct comparison between how the two methods perform will not be done. Instead, a qualitative assessment is done to show how the method performs using this novel approach to detect elementary events.

Vieira had as an objective the mapping of basic events that happened to the cyclist, namely turn left, turn right, among others. While in [27] an area on the image is defined to map the regions to do the OF calculations to then classify events, our approach uses the FOE calculated from the optical flow calculated as described above. Explicitly, we primarily use the Focus of Expansion discovered on the current frame and compare its changes to the previously calculate FOEs and then estimate an event based on the variation that occurred. Similar to [27], we detect the following elementary events:

- Turn Left: the cyclist turns left;
- Turn Right: the cyclist turns right;
- Ground Irregularities: there are some irregularities on the ground;
- Being Overtaken: the cyclist is being overtaken by some object in his left, and;
- Clear Path: there is no obstacles in the path the cyclist is taking.

Following is a brief description and results of how we proceed to detect the above mentioned elementary events.

A. Turn Left

The Turn Left event can be seen as the movement described by the cyclist when turning left at an intersection for example. In order to describe the Turn Left event, let us start by considering two additional movements: Smooth Turn Left and Sharp Turn Left. We defined these two supplementary movements because the former can be used to characterise small bike handling movements, and thus it is not a proper turn left as described above, and the latter because sometimes the user must do a sharp turn to avoid some unexpected obstacle, and thus, again, it is not a proper Turn Left.

To discover the Smooth Turn Left event, we start by looking at the current computed FOE and comparing it to the FOEs computed in the previous $M-1$ frames (i.e., we use (15) with $j = 1, \dots, M$). If the horizontal variation (x axis in the image) is above a certain threshold and the new FOE it to the left of the previous FOEs, we consider it to be a Smooth Turn Left movement. Again, to discover the Sharp Turn Left movement we proceed in the same manner, except for the used threshold being much higher than the one used in the Smooth Turn Left.

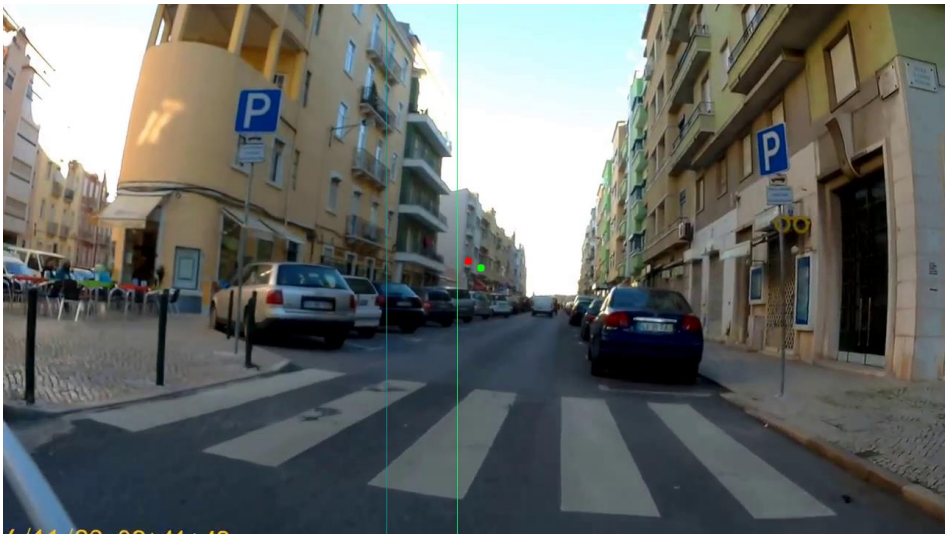


Figure 4.3.1 – Turn Left event. It is shown the current FOE (red point), the previously averaged FOEs (green point) and the thresholds for the Smooth Turn Left (green line) and for the Sharp Turn Left (cyan line).

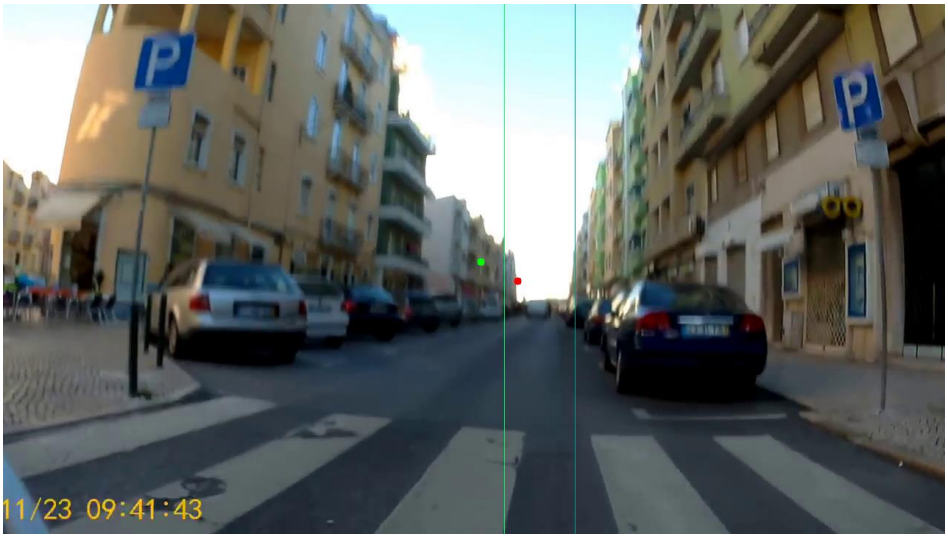


Figure 4.3.2 - Turn Right event. It is shown the current FOE (red point), the previously averaged FOEs (green point) and the thresholds for the Smooth Turn Right (green line) and for the Sharp Turn Right (cyan line). It is seen, that because the red point is to the right of the green line, there is currently a Smooth Turn Right movement.

To conclude, we use the Smooth Turn Left movement to discover the actual Turn Left event. We consider that if three Smooth Turn Left movements happen in a row, a Turn Left actually happened.

Figure 4.3.1 shows the thresholds used for the detection of the above described movements.

B. Turn Right

To discover the Turn Right event, we proceed exactly like described in the Turn Left event, considering the same extra Smooth Turn Right and Sharp Turn Right movements.

(a.1)

(a.2)

(a.3)

Figure 4.3.3 – Elementary Events: (a) Turn Left. Three Smooth Turn Left/Right events must happen prior to a Turn Left or Turn Right event. In sequence (a) the cyclist is turning left to pass the stopped white van.

However, while before, we must have considered the horizontal variation of the current FOE to the left of the previous FOEs, here we check if the new FOE is to the right of the previous FOEs. Again, we only consider that there was a Turn Right event if three Smooth Turn Right movements happened in a row. Figure 4.3.2 shows the thresholds used for the detection of the described movements.

Figure 4.3.3 shows an example of a Turn Left when the cyclist tries to overtake a stopped van.

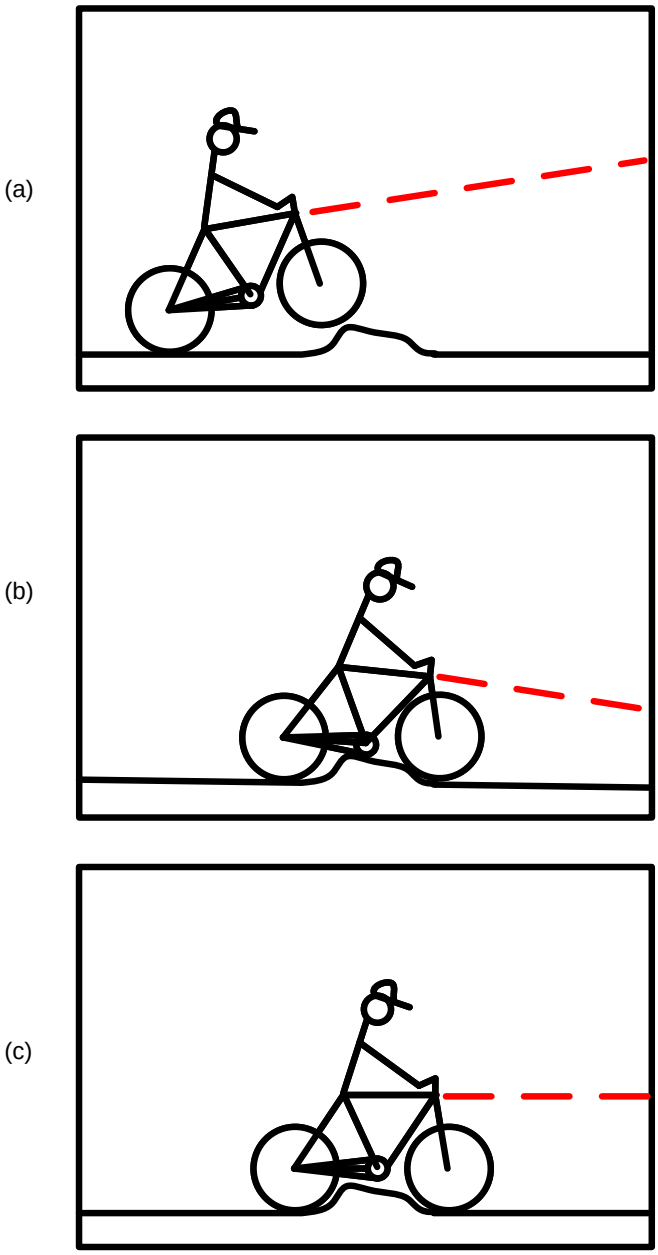


Figure 4.3.4 – Ground Irregularities event. On the left frames of a video are shown, while on the right a scheme of what is happening is presented. Green lines show the thresholds used to detect the vertical variation of the current FOE (red point) to the previous FOEs (green point).

C. Ground Irregularities

When there are ground irregularities, a specific series of event usually occur (see Figure 4.3.4). First, we see a huge variation in the vertical component of the FOE (y axis of the image). Following that variation, we see an approximate variation in the vertical opposite direction and finally, another variation is seen in the same vertical direction of the first variation. To check when this happens, we check the currently discovered FOE and check if its vertical component has surpassed a certain threshold and save this occurrence. In the next frame, we check whether the vertical component of the FOE has changed in the opposite direction of the saved state by a certain threshold and save this state. We finally re-test using the same test as the one used in the first saved state. If the occurrences tick all three tests, then we say there are some irregularities on the ground.

D. Being Overtaken

As shown in [27], when being overtaken, the optical flow which is calculated on the left side of the image shows a distinct behaviour: its direction is opposite to the flow resulting from the movement of the cyclist. Given this, we exploit two aspects of our work to detect the Being Overtaken event: we see if there is any object to the left of the FOE and we see to where the optical flow calculated in that area is pointing to.

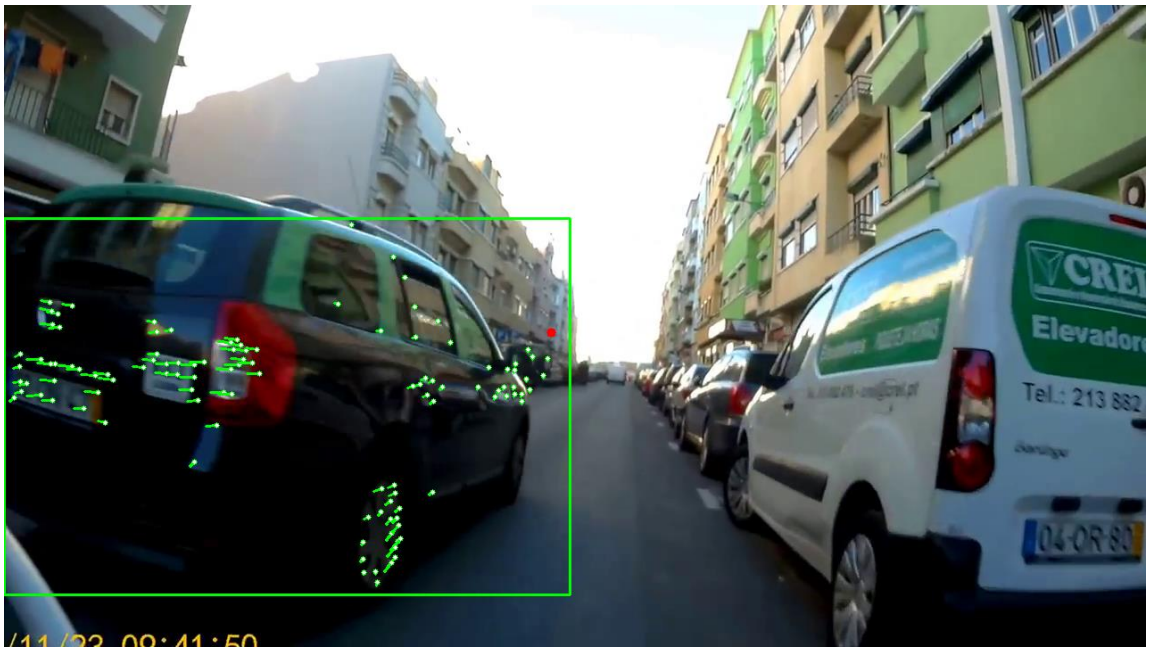
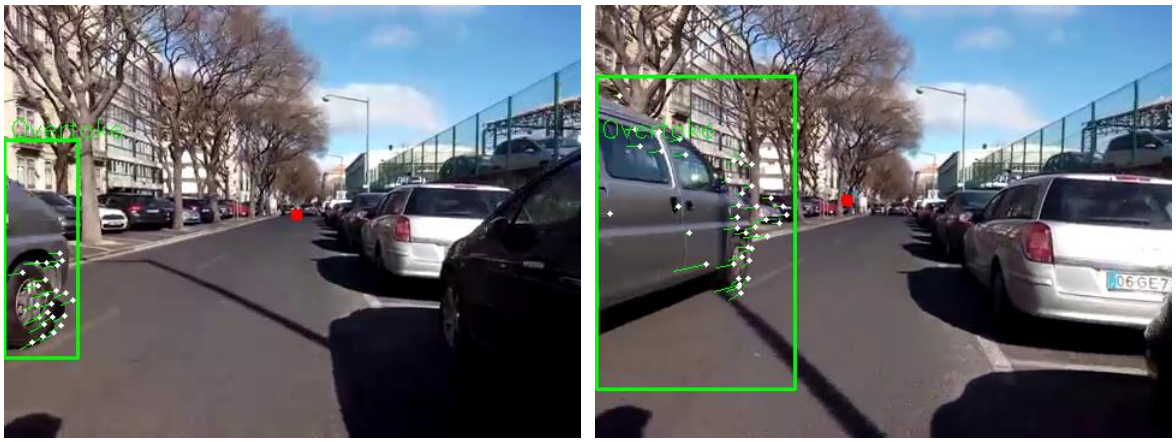


Figure 4.3.5 – Being Overtaken event. The green box contemplates the bounding box of the object as given by the NN and the green lines represent the optical flow pointing towards the FOE (red point).

First, we check if there is any object (car, bus, motorbike or bike) to the left of the FOE. If so, we then check whether the majority of the flow being calculated in the object reflects the movement of the user or if it points in the opposite direction and towards the FOE. If the latter is true, we assume that the cyclist is being overtaken, as shown in Figure 4.3.5 and Figure 4.3.6.

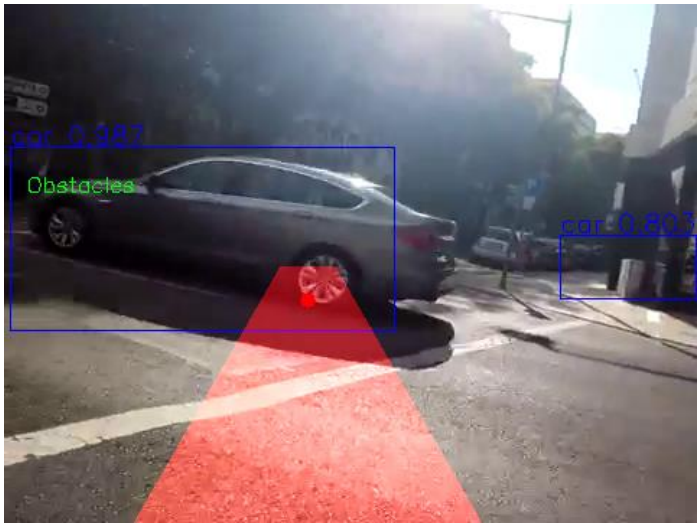


(a) (b)

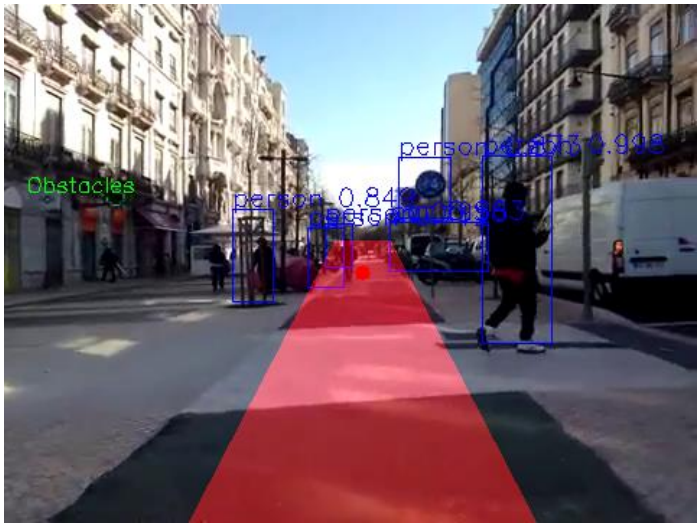
Figure 4.3.6 – Primal Event: Being Overtaken. When being overtaken, the optical flow calculated in the object points towards the FOE and not away from it as normally. Images (a) and (b) show precisely this.

E. Clear Path

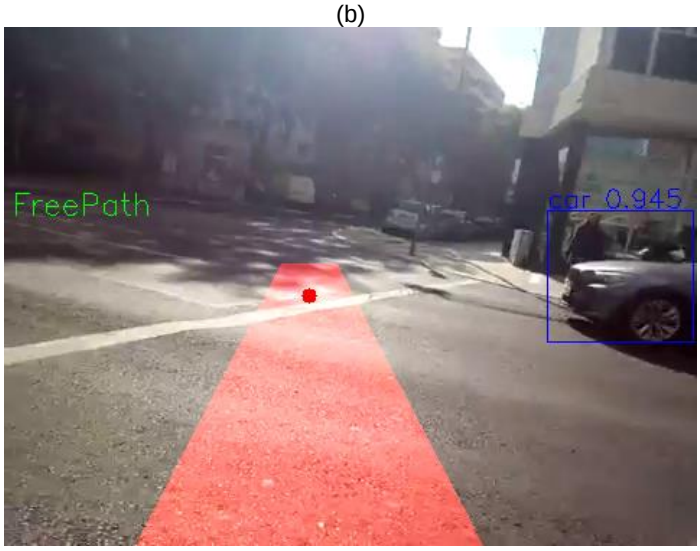
For the Clear Path event, we take advantage of the risk areas defined above. Here we simply check if there is any object in the image that is inside the red area. If so, we can conclude the path is occupied. Otherwise we can assess that the path is clear and not occupied by any object.



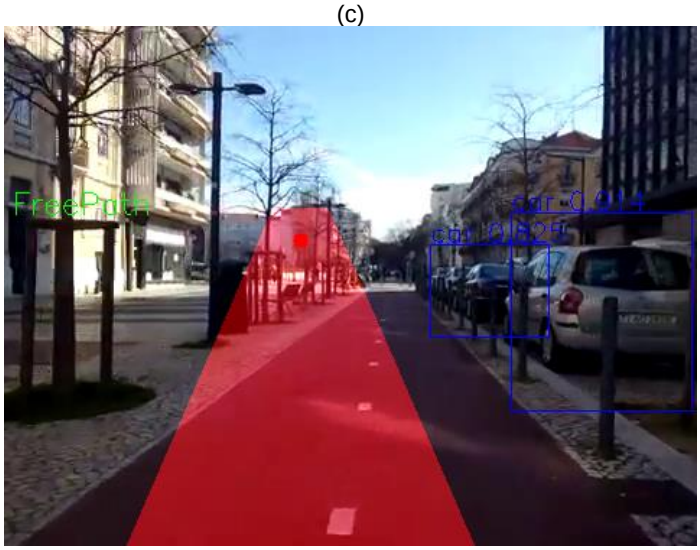
(a)



(b)



(c)



(d)

Figure 4.3.7 – Elementary event: Free Path. As shown in images (a) and (b) there are objects in the user's trajectory, whereas in (c) and (d) the path is clear and so we have a Free Path event.

Chapter 5

Conclusions

In this chapter, we present the conclusions taken from the developed work and what future work can be done following this thesis.

5.1 Conclusions

The main objectives for this work was to prove that with captured images from a smartphone mounted on a bicycle's handlebar it was possible to determine the direction of the cycler and to establish a risk assessment criteria that would enable the analysis of situations that bicyclists face each day.

The improved data capture system proved to be useful, as it enabled the capture of video directly from the developed smartphone's application, along with other sensory data.

The videos captured proved that information related to motion is can still be gathered in the form of optical flow vectors, that can then be transformed into the Focus of Expansion, which determines the direction of the cycler. This important point in the image can be easily computed, despite the constant shakiness of the handlebar and the bike itself. This is due to the computed weights of both the magnitude of vectors and objects in the image and the weighted average of the previous Foci of Expansion of previous frames in the video.

Concerning the risk assessment, two different criteria were developed regarding the occupation of the cycler's path and the proximity to objects. Both criteria are useful in assessment the amount of danger the cycler faces in each situation along its ride because one focus on obstacles along its journey, whereas the second the distance to each object. This makes it that the developed work can be used in mapping geographic locations where constant danger situations happen, and thus help urban planners plan better cycling infrastructures that ultimately contribute to a healthier and safer mean of transportation.

5.2 Future Work

Regarding future work, it would be interesting to distribute the improved developed app to a large number of users to better assess where and what dangers cyclists face each day.

Another route that can be taken is to improve the detection of objects as these take a major role throughout this work. In fact, it what could be done is take image samples from our data capture system and use these as training data for the neural network, as this way, the training would directly affect the results, as the network would be trained for the kind of images that we capture along any ride and not images of cars which are placed in a completely different situation.

In regard to the risk descriptor, firstly it would be interesting to assess danger situations using other metrics that can be more useful in certain situation and give more information to city planners. Secondly, using an

example, it would also be interesting to assess how the current road and cycling infrastructures works being conducted in the city of Lisbon at this moment would impact the difference in risk that cyclers face when riding in these work locations, because currently there is no concrete method to evaluate how this construction works would benefit bicyclists.

References

[1] G. Vandenbulcke, C. Dujardin, Thomas, Isabelle, B. d. Geus, B. Degraeuwe, R. Meeusen and L. I. Panis, “Cycle commuting in Belgium: Spatial determinants and ‘re-cycling’ strategies,” *Transportation Research Part A: Policy and Practice,* vol. 45, no. 2, pp. 118-137, February 2011.

[2] R. J. Shepard, “Is Active Commuting the Answer to Population Health?,” *Sports Med,* vol. 38, no. 9, pp. 751-758, 2008.

[3] P. Oja, I. Vuori and O. Paronen, “Daily walking and cycling to work: their utility as health-enhancing physical activity,” *Patient Education and Counseling,* vol. 33, no. 1, pp. S87-S94, April 1998.

[4] I. L. Husting, “EU actions on sustainable tourism and EU funding for tourism 2014-2020,” 2014. [Online]. Available: http://www.eurovelo.org/home/eurovelo-greenways-and-cycling-tourism-conferences/conference-2014-lessons-from-the-demarrage-project/. [Accessed 28 February 2017].

[5] European Cyclist' Federation, [Online]. Available: http://www.eurovelo.org/home/what-is-eurovelo/. [Accessed 28 February 2017].

[6] R. Weston and J. C. Mota, “Low Carbon Tourism Travel: Cycling, Walking and Trails,” *Tourism Planning & Development,* vol. 9, no. 1, pp. 1-3, February 2012.

[7] J. Pucher, “Cycling Safety on Bikeways vs. Roads,” *Transportation Quarterly,* vol. 55, no. 4, pp. 9-11, September 2001.

[8] National Highway Traffic Safety Administration, National Center for Statistics and Analysis, “Traffic Safety Facts, Research Note,” U.S. Department of Transportation, Washington, DC, August 2016.

[9] City of Boston, “Cyclist Safety Report,” City of Boston, Boston, 2013.

[10] New York City, “Bicyclists Network and Statistics,” [Online]. Available: http://www.nyc.gov/html/dot/html/bicyclists/bikestats.shtml#crashdata. [Accessed 1 March 2017].

[11] New York City Department of Transportation, “Protected Bicycle Lanes in NYC,” September 2014. [Online]. Available: http://www.nyc.gov/html/dot/downloads/pdf/2014-09-03-bicycle-path-data-analysis.pdf. [Accessed 1 March 2017].

[12] Canalys, “Smart phones overtake client PCs in 2011,” 3 February 2012. [Online]. Available: https://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011. [Accessed 23 February 2017].

[13] IDC, “IDC: 87% Of Connected Devices Sales By 2017 Will Be Tablets And Smartphones,” 12 September 2013. [Online]. Available: http://www.forbes.com/sites/louiscolumbus/2013/09/12/idc-87-of-connected-devices-by-2017-will-be-tablets-and-smartphones/#758960982472. [Accessed 23 February 2017].

[14] Lifewire, “How Many Apps Are in the App Store?,” [Online]. Available: https://www.lifewire.com/how-many-apps-in-app-store-2000252. [Accessed 20 February 2017].

[15] AppBrain, “Number of Android applications,” [Online]. Available: https://www.appbrain.com/stats/number-of-android-apps. [Accessed 20 February 2017].

[16] L. Pei, R. Guiness, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, L. Chen and J. Kaistinen, “Human Behavior Cognition Using Smartphone Sensors,” *Sensors,* vol. 13, no. 2, pp. 1402-1424, 2013.

[17] E. Mitchell, D. Monaghan and N. E. O'Connor, “Classification of Sporting Activities Using Smartphone Accelerometers,” *Sensors,* vol. 13, no. 4, pp. 5317-5337, 2013.

[18] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga, “Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey,” in *23th International Conference on Architecture of Computing Systems 2010*, Hannover, Germany, 2010.

[19] A. Anjum and M. U. Ilyas, “Activity Recognition Using Smartphone Sensors,” in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2013.

[20] X. Su, H. Tong and P. Ji, “Activity Recognition with Smartphone Sensors,” *TSINGHUA SCIENCE AND TECHNOLOG,* vol. 19, no. 3, pp. 235-249, 2014.

[21] D. A. Johnson and M. M. Trivedi, “Driving style recognition using a smartphone as a sensor platform,” in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, 2011.

[22] H. Eren, S. Makinist and E. Akin, “Estimating Driving Behaviour by a Smartphone,” in *2012 Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, 2012.

[23] F. Seraj, K. Zhang, O. Turkes, N. Meratnia and P. J. M. Havinga, “A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior,” in *UbiComp/ISWC'15 Adjunct Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers* , Osaka, Japan, 2015.

[24] R. Araújo, Â. Igreja and R. d. Castro, “Driving coach: A smartphone application to evaluate driving efficient patterns,” in *2012 IEEE Intelligent Vehicles Symposium*, Alcala de Henares, 2012.

[25] J. Strauss, L. F. Miranda-Moreno and P. Morency, “Mapping cyclist activity and injury risk in a network combining smartphone GPS data and bicycle counts,” *Accident Analysis & Prevention,* vol. 83, pp. 132-142, October 2015.

[26] S. Panichpapiboon and P. Leakkaw, “Traffic Sensing Through Accelerometers,” *IEEE Transactions on Vehicular Technology,* vol. 65, no. 5, pp. 3559-3567, May 2016.

[27] P. M. S. Vieira, “Percepção do risco em ambiente rodoviário urbano,” M.S thesis, Dept. Elect. Eng., UTL, IST, Lisbon, 2015.

[28] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems, NIPS*, 2015.

[29] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *ECCV*, 2016.

[31] D. Ballard and C. Brown, “Focus of Expansion,” in *Computer Vision*, Prentice Hall, First Edition, May 1982, p. 199.

[32] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” *Proceedings of Imaging Understanding Workshop,* pp. 121-130, 1981.

[33] M. W. Tao, J. Bai, P. Kohli and S. Paris, “SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm,” *Computer Graphics Forum (Eurographics 2012),* vol. 31, no. 2, May 2012.

[34] P. Weinzaepfel, J. Revaud, Z. Harchaoui and C. Schmid, “DeepFlow: Large displacement optical flow with deep matching,” in *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[35] C. Zhang, H. Li, X. Wang and X. Yang, “Cross-scene crowd counting via deep convolutional neural networks,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015.

[36] M. Everingham, L. Van~Gool, C. K. I. Williams, J. Winn and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,* http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[37] Y. Rubner, C. Tomasi and L. J. Guibas, “The Earth Mover’s Distance as a Metric for Image Retrieval,” *International Journal of Computer Vision,* vol. 40, no. 2, pp. 99-121, 2000.

[38] M. Grundmann, V. Kwatra, D. Castro and I. Essa, “Calibration-Free Rolling Shutter Removal,” in *IEEE ICCP*, 2012.

[39] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman and K. Zuiderveld, “Adaptive histogram equalization and its variations,” *Computer Vision, Graphics, and Image Processing,* vol. 39, no. 3, pp. 355 - 368, 1987.

[40] J. Shi and C. Tomasi, “Good Features to Track,” in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593-600.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[42] O. Pele and M. Werman, “A linear time histogram metric for improved SIFT matching,” *Computer Vision - ECCV 2008,* pp. 495-508, 2008.

[43] O. Pele and M. Werman, “Fast and robust earth mover's distances,” *Proc. 2009 IEEE 12th Int. Conf. on Computer Vision,* pp. 460-467, 2009.

[44] M. S. Andersen, J. Dahl and L. Vandenberghe, “CVXOPT: A Python package for convex optimization, version 1.1.9,” 2 December 2016. [Online]. Available: cvxopt.org.

[45] M. S. Andersen, J. Dahl and L. Vandenberghe, “CVXOPT L1-norm approximation,” 02 December 2016. [Online]. Available: http://cvxopt.org/examples/mlbook/l1.html.

[46] Statista, “Number of smartphone users worldwide from 2014 to 2020 (in billions),” [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. [Accessed 20 February 2017].

[52] J. Lee and A. C. Bovik, “Estimation and analysis of urban traffic flow,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, November 2009.

[53] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM,* vol. 24, no. 6, pp. 381-395, June 1981.

[54] D. A. Forsyth and J. Ponce, “RANSAC,” in *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference, 2002, pp. 480-483.

Appendixes

Appendix A – Mobility in Cities: App's usage

The app is divided in two main menus, each depicted in

Figure A.1:

1. Gather Data: menu responsible for acquiring data;
2. Upload Data: menu for uploading the previously acquired data to a server.

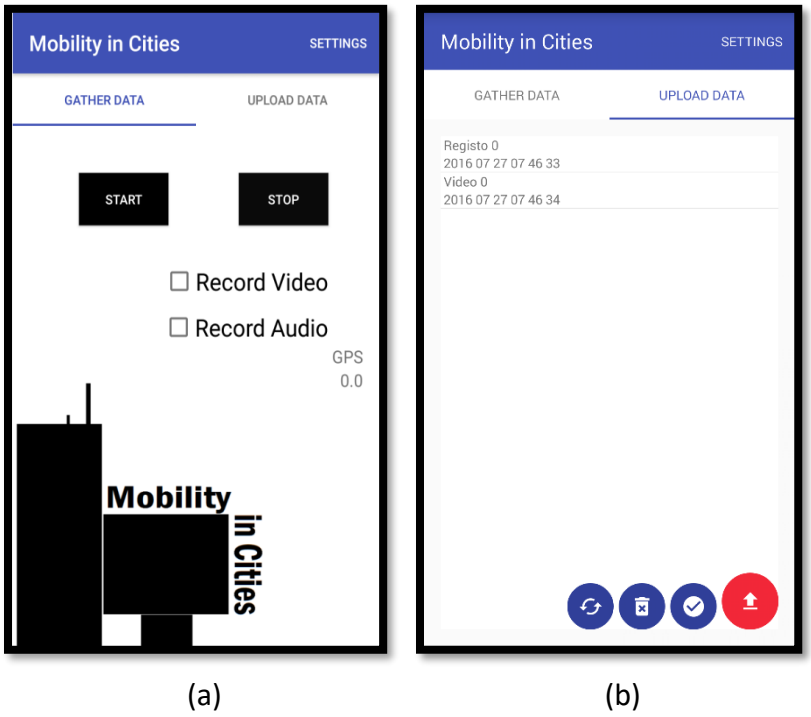


Figure A.1 – Screenshots of the App's menus: (a) Gather Data and (b) Upload Data.

Concerning the data acquisition process we can consider the flowchart shown in Figure A.2 which illustrates how the App works after the Start button is pressed.

After "Start" is pressed, it is checked if it is a sensor only acquisition or a sensor, film or audio acquisition. The sensor data is written in a text file with all the variables described above. A writing cycle is created and a measurement is taken every $100\ ms$. However, because each writing to the file takes on average 20 to $30\ ms$, we end up with a variable writing frequency of 7 to $9\ Hz$. If it is also a movie acquisition, a service for

the recording is started according to the user desirable recording quality and frames-per-second recording, which are explained further down. The recording is started inside the service and a MP4 file is created to save the movie. If an audio recording is also desired, a new service is started and a new 3GP audio file is created.

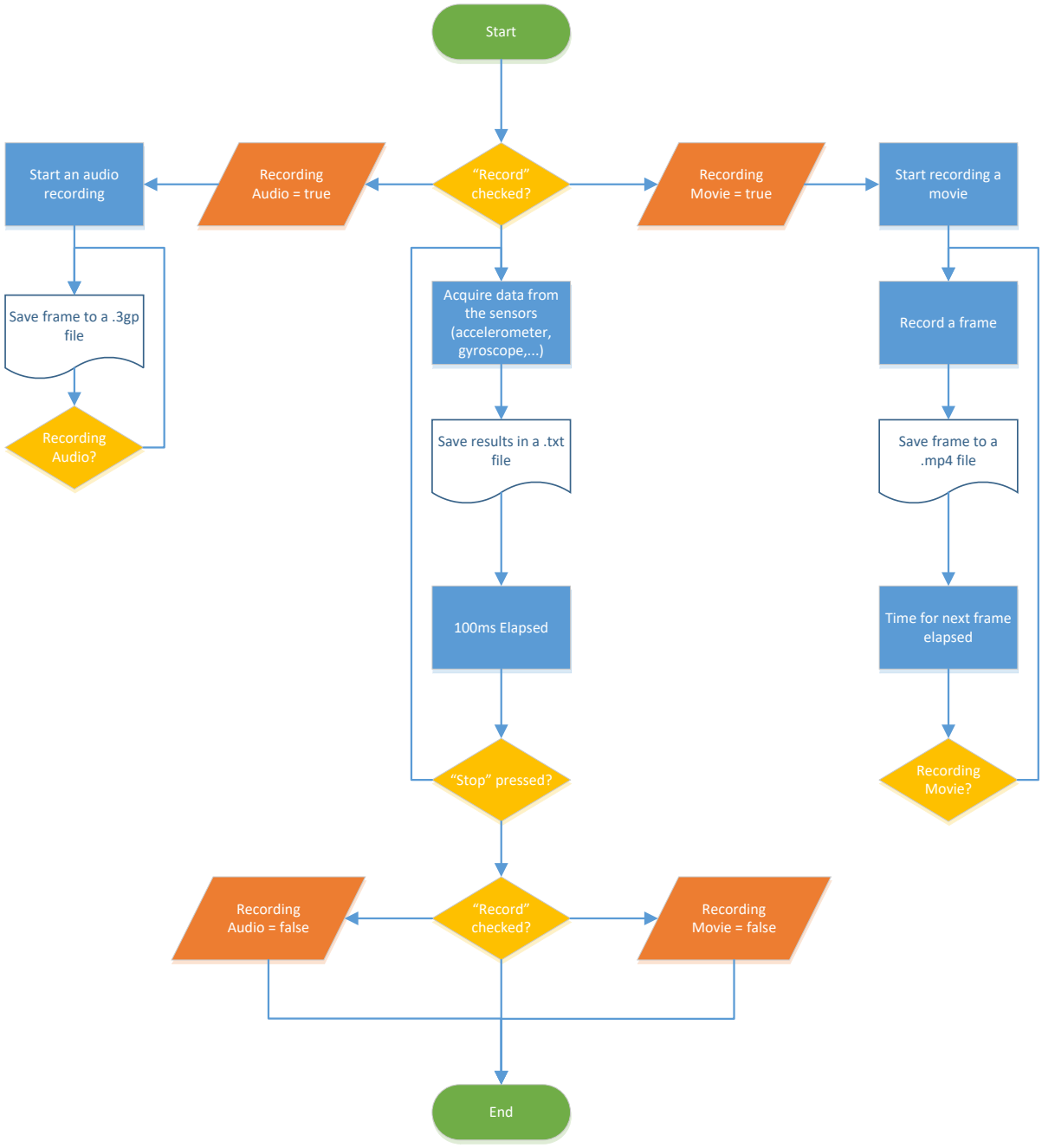


Figure A.2 – Flowchart of the data acquisition of the App.

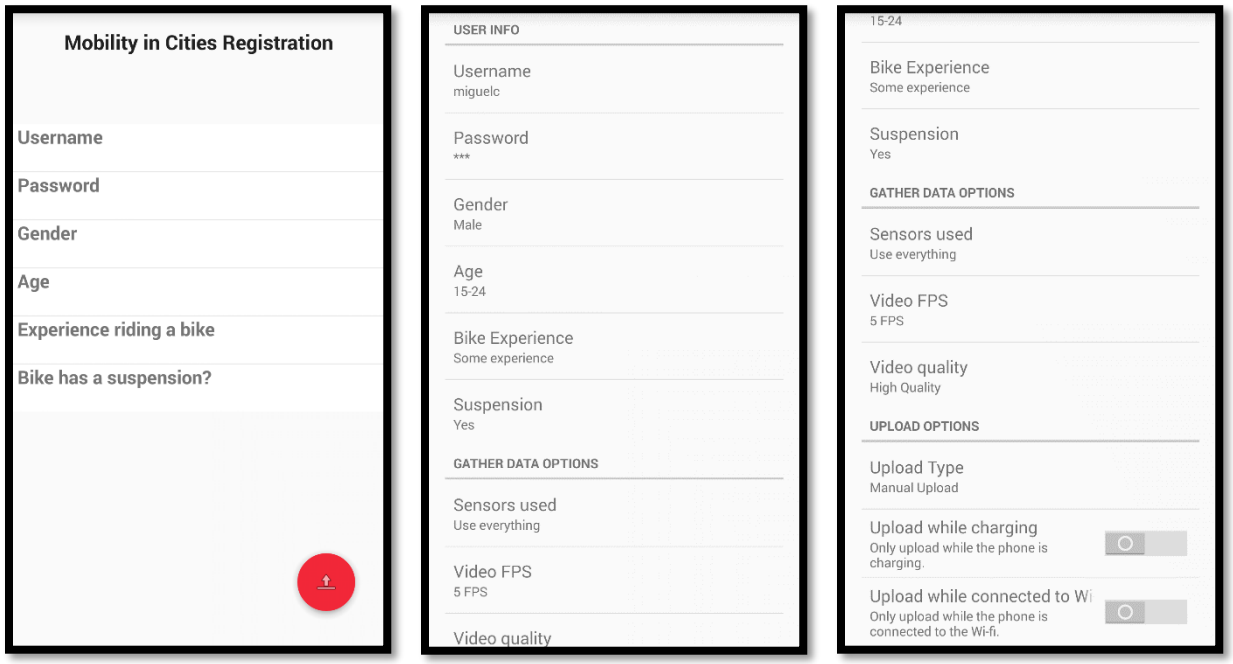
When "Stop" is pressed the sensor acquisition is halted and a message is sent to the previously started services (if it was the case) to finish the recordings. Both the sensor text file, the MP4 movie file and the

3GP is saved under an app directory in the flash memory of the smartphone. All files share the same name so that it is known which files match each other.

On the Upload Data menu, a list of all saved files is shown, and the user can select or deselect files to upload. Four buttons are presented (enumerated from left to right):

- Refresh the current list: searches for new recordings in the app's folder in the flash memory;
- Delete all already sent files: deletes all sent files to the server to save memory space;
- Select all unsent files: select all files that have yet to be set to the server, and;
- Upload selected files: uploads the selected files to the server.

The app has also some settings which need to be configured prior to acquiring some data. Figure A.3 (a) portraits the First Time Configuration menu, where the user is asked to input some information about itself. This information will later be used to create an individual user account, where the user can view its walking, biking or driving recorded paths, and other information which will later be accessible.



(a) (b.1) (b.2)

Figure A.3 - Screenshots of the App's menus: (a) First Time Configuration (b) Settings.

Other app settings are also important for the correct functioning of the acquisition process. These other settings are accessible in the Settings menu shown in Figure A.3 (b). These other settings include:

- Video FPS: choose between recording at 1, 5, 15 or 30 FPS;

- Video Quality: choose between recording at Low Quality or High Quality, where each one of these depend on the smartphone hardware;
- Upload while charging: choose between only being able to upload a file while the device is being charged, and;
- Upload only while connected to Wi-Fi: choose between only being able to upload a file while the device is connected to Wi-Fi.

Other settings which were not released in the current version of the App include an automatically uploading feature, which would upload the acquisition files directly when the smartphone is connected to the Internet and a smaller sensor acquisition, which would decrease the amount of captured data to only acquire essential data and therefore decrease the time it takes to write the captured sensor values in the file.

Appendix B – Different Methods to calculate the Focus of Expansion

There are other ways to compute the Focus of Expansion in an image frame. Below is the formulation of other two optimization problems, as well as the description of two RANSAC initialized procedures.

A. Least Squares Optimization Problem

Least Squares (or l_2-norm) is a optimization problem formulation which provides a stable solution that can be found using analytical computation. However, one must consider that outliers may take a huge role in disrupting the final result, as they get amplified in this formulation.

The optimization problem using the least squares can be formulated as in (22) and in (23) considering the weights w_i.

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} f(x, L_i)^2 \tag{22}$$

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} w_i \cdot f(x, L_i)^2 \tag{23}$$

B. Manhattan Distance Optimization Problem

The Manhattan Distance or l_1-norm, improves the l_2-norm (or Least Squares) on the fact that is more robust to outliers as it does not amplify them. However, it does not provide a close form solution and thus can be computational expensive when compared to the l_2-norm.

The optimization problem using the l_1-norm can be formulated as in (24) and in (25) considering the weights w_i.

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} |f(x, L_i)| \tag{24}$$

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} w_i |f(x, L_i)| \tag{25}$$

Figure B.1 – A generic view on a one-dimension variable for Least Squares (red), Manhattan Distance (green) and Huber Loss with $\delta = 1$ (blue) functions. shows a representation of the 3 functions used in the three optimization problems (Least Squares Distance, Manhattan Distance and Huber Loss Distance) using a one-dimension variable.

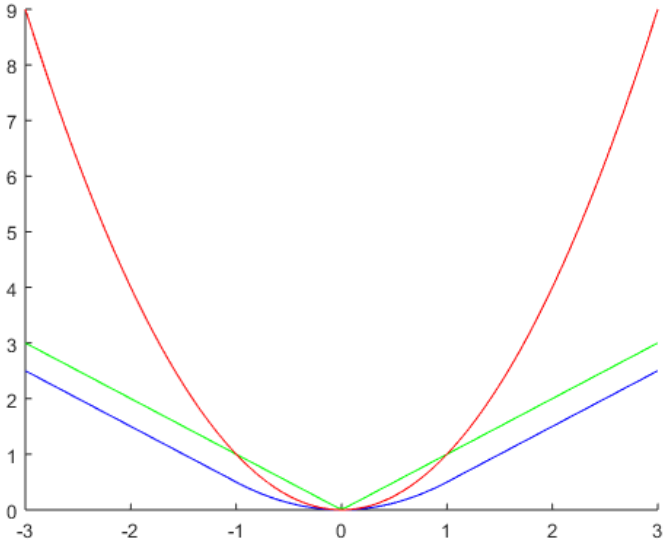


Figure B.1 – A generic view on a one-dimension variable for Least Squares (red), Manhattan Distance (green) and Huber Loss with $\delta = 1$ (blue) functions.

C. RANSAC

The Random Sample Consensus (RANSAC) is an iterative method to try and estimate a model from data samples. Its objective is to try and find a minimum number of points from the data that model a certain scenario. In our case, it is to try and find a minimum number of optical flow vectors that converge to a certain location in the image (the FOE). To do this, we start by randomly selecting two optical flow vectors from the ones found above. The intersection of the two lines given by the extension of these two optical flow vectors is our estimated FOE. We then check how many other OF vectors also point towards this estimated FOE. If a certain minimum number of vectors respects this FOE, we accept the estimated FOE and output x_{FOE}. However, in the event that not a minimum number of vectors is found that respect the found FOE, we restart and repeat the process, choosing another set of two random initial vectors.

We consider that to have an acceptable estimation of the focus of expansion we must have at least a ratio of 60% of vectors that point towards the same FOE from the whole set of optical flow vectors, and we consider these vectors as inliers if the angle between the line from the FOE and the vector and the line of the extension of the FOE is below a certain threshold (e.g. $15°$, see). A pseudocode of this method can be found in Algorithm 1.

Algorithm 1 RANSAC

Data: OF vectors
Result: FOE

for $\_MaximumNumberIterations$ **do**
  $subset \leftarrow$ randomlyChooseTwoVectors($Data$)
  $x \leftarrow$ intersectionLines($subset$)
  $inliers \leftarrow$ findInliers($Data$, x)
  $ratioInliers \leftarrow \#inliers \,/\, \#Data$
  if $ratioInliers > \_MinimumInliersRatio$ **then**
    return x
end

D. Modified RANSAC

The last method we test to find the FOE is what we denominated by modified RANSAC, as it starts by using the RANSAC iterative method but then tries to optimize each RANSAC iteration.

Again, we start by randomly selecting two OF vectors from the set of available vectors. Given these two vectors, we find their line's extension intersection. Using this intermediate intersection point we then find what vectors are considered as inliers (again using the method described above). Now, instead of checking if the ratio of inliers is above an acceptable value, we try and solve the least square optimization problem using all the available inliers at the current iteration. Specifically, we solve the method described in A. Least Squared Optimization Problem, but now, only using the vectors we consider as inliers and not the whole set of OF vectors. Since all the vectors are considered as inliers, we consider that there is almost to no gain in using the other optimization methods described (l_1-norm or Huber Loss) and as such we can achieve a faster solution. After discovering this new point, we check again if there are new vectors that have become inliers to the newly discovered point. If there are, we re-ran the optimization problem including the newly added inliers and discover a new FOE. We repeat the process until we have converged, i.e., the difference between the newly discovered point and the previously discovered point is below a certain threshold, or until a maximum number of iteration have passed. However, after this iteration method, we still check whether the ratio of inliers to the number of vectors is above a certain threshold (e.g. 60%). If it is, we accept this point as the FOE. If not, we restart the whole process by randomly choosing another set of two OF vectors. A pseudocode for this modified RANSAC method can be found in Algorithm 2.

At first glance, all five methods (including the Huber Loss Distance optimization) are able to estimate well the FOE. However, the last two explored methods present some inconsistencies. In the RANSAC and Modified RANSAC approaches we set two parameters: the inlier acceptance angle as $\theta_{inlier} \leq 15°$ and the acceptable inlier's ratio to 60%. Because of the fact that we only accept FOE estimations where we have an inlier's to total number of vectors ratio of at least 60% and have this inlier's condition, this makes it that

Algorithm 2 Modified RANSAC

Data: OF vectors
Result: FOE

for $\_MaximumNumberIterations$ **do**
  $subset \leftarrow$ randomlyChooseTwoVectors($Data$)
  $x \leftarrow$ intersectionLines($subset$)
  $x_1 \leftarrow 0$
  for $x - x_1 > \_MinimumDistanceForConvergence$ **and** $inliers \neq \emptyset$ **do**
    $x \leftarrow x_1$
    $inliers \leftarrow$ findInliers($Data$, x)
    $x_1 \leftarrow$ leastSquaresOptimizationProblem($inliers$)
    $ratioInliers \leftarrow \#inliers \ / \ \#Data$
    if $ratioInliers > \_MinimumInliersRatio$ **then**
      return x_1
  end
end

not all frames have a solution to our problem. In fact, in our tests, only about of 20-40% of the frames have a solution for these two approaches. As such, we decided to forgo these two methods, as it is of the upmost importance to estimate the FOE in a more general and consistent scenario.

Next, we set to study the differences of estimating the FOE using one of the three optimization aforementioned methods. In order to solve (22) and (23) we used the Python Toolbox $cvxopt$ [44] and the custom module $L1\text{-}norm\ approximation$ [45]. Figure B.2 show a heatmap of the distance between the optical flow lines L_i and the Focus of Expansion, $f(\tilde{x}, L_i)$. Figure B.3 shows a comparison between all three approaches. As shown, all three methods are able to do a good estimation of the Focus of Expansion. However, when there are numerous miscalculations on the optical flow, the Least Squares approach is expected to perform worse than the other two, as it exponentiates outliers. The other two method's results are very similar to one another and behave similarly, as one can inspect in Figure B.3.

Being so similar in terms of results, in the end, we chose as the final and best estimator the Huber Loss method, as it provides the best of both worlds, being robust to outliers, but still penalizing them in the process. Thus, what follows is in respect to the Focus of Expansion estimation using the Huber Loss Distance Optimization method.

(a.1)

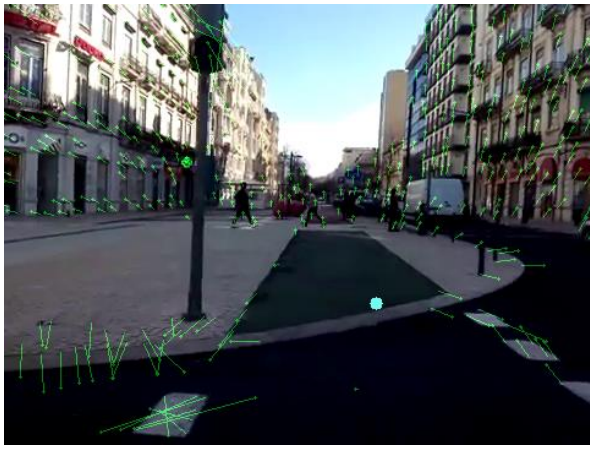
(a.2)

(b.1)

(b.2)

(c.1)

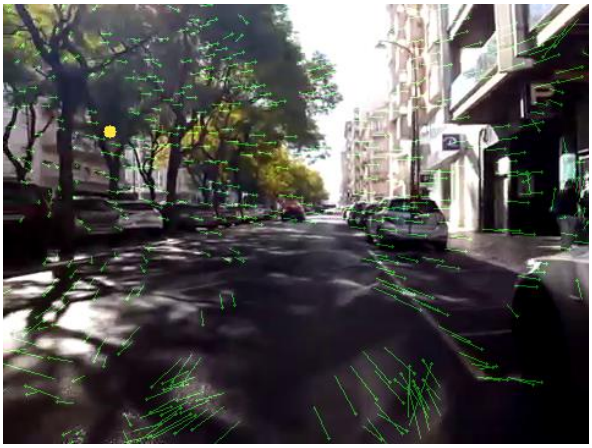
(c.2)



(d.1)

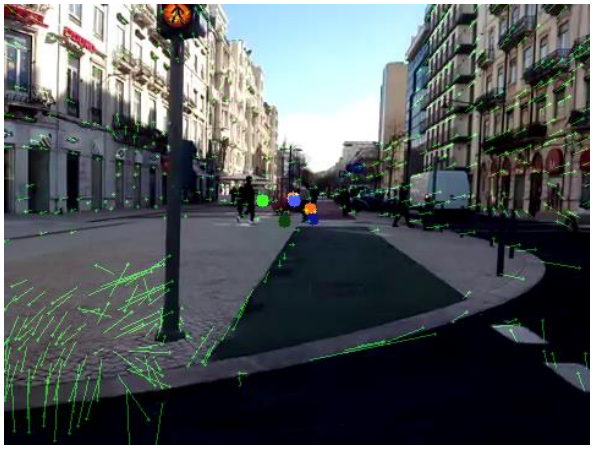
(d.2)

(e.1)

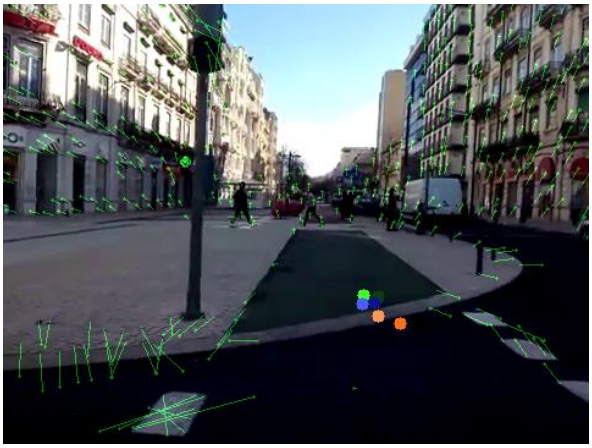


(e.2)

Figure B.2 - Examples of the FOE estimation methods: (a) - $l1$ Norm Optimization, (b) – Least Squares Optimization, (c) – Huber Loss Optimization, (d) – RANSAC, and (e) – Modified RANSAC. For (a), (b) and (c) the lighter colour point represent the solution without any weight consideration, whereas the darker colour point represents the optimization using weights.



(a)



(b)

(c)

(d)

(e)

(f)

Figure B.3 – Estimation of the FOE using all three optimization methods: green – Least Squares Optimization; orange – Manhattan Distance Optimization, and; blue – Huber Loss Optimization. Again, lighter colour point represents the weightless optimization and darker colour the weighted optimization problem solutions.

Appendix C – Distance Matrices used in the Earth Mover's Distance

Distance Matrix for EDM in the Lane/Path Occupation criteria:

$$C_{PathOccupation} =$$

$$\begin{bmatrix}
0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 10 & 2 & 4 & 6 & 8 & 10 & 8 & 12 & 16 & 20 & 24 & 8 & 12 & 16 & 20 & 24 \\
1 & 0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 4 & 2 & 4 & 6 & 8 & 12 & 8 & 12 & 16 & 20 & 12 & 8 & 12 & 16 & 20 \\
2 & 1 & 0 & 1 & 2 & 6 & 4 & 2 & 4 & 6 & 6 & 4 & 2 & 4 & 6 & 16 & 12 & 8 & 12 & 16 & 16 & 12 & 8 & 12 & 16 \\
3 & 2 & 1 & 0 & 1 & 8 & 6 & 4 & 2 & 4 & 8 & 6 & 4 & 2 & 4 & 20 & 16 & 12 & 8 & 12 & 20 & 16 & 12 & 8 & 12 \\
4 & 3 & 2 & 1 & 0 & 10 & 8 & 6 & 4 & 2 & 10 & 8 & 6 & 4 & 2 & 24 & 20 & 16 & 12 & 8 & 24 & 20 & 16 & 12 & 8 \\
2 & 4 & 6 & 8 & 10 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 10 & 2 & 4 & 6 & 8 & 10 \\
4 & 2 & 4 & 6 & 8 & 1 & 0 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 4 & 2 & 4 & 6 & 8 \\
6 & 4 & 2 & 4 & 6 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 2 & 6 & 4 & 2 & 4 & 6 & 6 & 4 & 2 & 4 & 6 \\
8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 0 & 1 & 8 & 6 & 4 & 2 & 4 & 8 & 6 & 4 & 2 & 4 \\
10 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 0 & 10 & 8 & 6 & 4 & 2 & 10 & 8 & 6 & 4 & 2 \\
2 & 4 & 6 & 8 & 10 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 10 & 2 & 4 & 6 & 8 & 10 \\
4 & 2 & 4 & 6 & 8 & 1 & 0 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 3 & 4 & 2 & 4 & 6 & 8 & 4 & 2 & 4 & 6 & 8 \\
6 & 4 & 2 & 4 & 6 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 2 & 6 & 4 & 2 & 4 & 6 & 6 & 4 & 2 & 4 & 6 \\
8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 0 & 1 & 8 & 6 & 4 & 2 & 4 & 8 & 6 & 4 & 2 & 4 \\
10 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 0 & 10 & 8 & 6 & 4 & 2 & 10 & 8 & 6 & 4 & 2 \\
8 & 12 & 16 & 20 & 4 & 2 & 4 & 6 & 8 & 10 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\
12 & 8 & 12 & 16 & 20 & 4 & 2 & 4 & 6 & 8 & 4 & 2 & 4 & 6 & 8 & 1 & 0 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 3 \\
16 & 12 & 8 & 12 & 16 & 6 & 4 & 2 & 4 & 6 & 6 & 4 & 2 & 4 & 6 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 2 \\
20 & 16 & 12 & 8 & 12 & 8 & 6 & 4 & 2 & 4 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 0 & 1 \\
24 & 20 & 16 & 12 & 8 & 10 & 8 & 6 & 4 & 2 & 10 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 0 \\
8 & 12 & 16 & 20 & 24 & 2 & 4 & 6 & 8 & 10 & 2 & 4 & 6 & 8 & 10 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\
12 & 8 & 12 & 16 & 20 & 4 & 2 & 4 & 6 & 8 & 4 & 2 & 4 & 6 & 8 & 1 & 0 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 3 \\
16 & 12 & 8 & 12 & 16 & 6 & 4 & 2 & 4 & 6 & 6 & 4 & 2 & 4 & 6 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 & 2 \\
20 & 16 & 12 & 8 & 12 & 8 & 6 & 4 & 2 & 4 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 0 & 1 \\
24 & 20 & 16 & 12 & 8 & 10 & 8 & 6 & 4 & 2 & 10 & 8 & 6 & 4 & 2 & 4 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 0
\end{bmatrix}$$

(26)

Distance Matrix for EDM in the Proximity criteria:

$$C_{Proximity} =$$

$$\begin{bmatrix}
0 & 2 & 4 & 6 & 8 & 1 & 4 & 6 & 16 & 20 & 1 & 4 & 6 & 16 & 20 & 2 & 3 & 4 & 10 & 12 & 2 & 3 & 4 & 10 & 12 \\
2 & 0 & 2 & 4 & 6 & 4 & 1 & 2 & 6 & 8 & 4 & 1 & 2 & 6 & 8 & 6 & 4 & 6 & 4 & 5 & 6 & 4 & 7 & 4 & 5 \\
4 & 2 & 0 & 1 & 2 & 12 & 4 & 2 & 2 & 3 & 12 & 4 & 2 & 2 & 3 & 16 & 12 & 8 & 6 & 8 & 16 & 12 & 8 & 6 & 8 \\
6 & 4 & 1 & 0 & 1 & 16 & 6 & 4 & 1 & 2 & 16 & 6 & 4 & 1 & 2 & 20 & 16 & 12 & 4 & 6 & 20 & 16 & 12 & 4 & 6 \\
8 & 6 & 2 & 1 & 0 & 20 & 8 & 6 & 2 & 1 & 20 & 8 & 6 & 2 & 1 & 24 & 20 & 16 & 6 & 4 & 24 & 20 & 16 & 6 & 4 \\
1 & 4 & 12 & 16 & 20 & 0 & 2 & 4 & 12 & 16 & 0 & 2 & 4 & 12 & 16 & 1 & 2 & 3 & 8 & 10 & 1 & 2 & 3 & 8 & 10 \\
4 & 1 & 4 & 6 & 8 & 2 & 0 & 1 & 4 & 6 & 2 & 0 & 1 & 4 & 6 & 4 & 2 & 4 & 3 & 4 & 4 & 2 & 4 & 3 & 4 \\
6 & 2 & 2 & 4 & 6 & 4 & 1 & 0 & 2 & 4 & 4 & 1 & 0 & 2 & 4 & 6 & 4 & 2 & 2 & 3 & 6 & 4 & 2 & 2 & 3 \\
16 & 6 & 2 & 1 & 2 & 12 & 4 & 2 & 0 & 1 & 12 & 4 & 2 & 0 & 1 & 16 & 12 & 8 & 2 & 4 & 16 & 12 & 8 & 2 & 4 \\
20 & 8 & 3 & 2 & 1 & 16 & 6 & 4 & 1 & 0 & 16 & 6 & 4 & 1 & 0 & 20 & 16 & 12 & 4 & 2 & 20 & 16 & 12 & 4 & 2 \\
1 & 4 & 12 & 16 & 20 & 0 & 2 & 4 & 12 & 16 & 0 & 2 & 4 & 12 & 16 & 1 & 2 & 3 & 8 & 10 & 1 & 2 & 3 & 8 & 10 \\
4 & 1 & 4 & 6 & 8 & 2 & 0 & 1 & 4 & 6 & 2 & 0 & 1 & 4 & 6 & 4 & 2 & 4 & 3 & 4 & 4 & 2 & 4 & 3 & 4 \\
6 & 2 & 2 & 4 & 6 & 4 & 1 & 0 & 2 & 4 & 4 & 1 & 0 & 2 & 4 & 6 & 4 & 2 & 2 & 3 & 6 & 4 & 2 & 2 & 3 \\
16 & 6 & 2 & 1 & 2 & 12 & 4 & 2 & 0 & 1 & 12 & 4 & 2 & 0 & 1 & 16 & 12 & 8 & 2 & 4 & 16 & 12 & 8 & 2 & 4 \\
20 & 8 & 3 & 2 & 1 & 16 & 6 & 4 & 1 & 0 & 16 & 6 & 4 & 1 & 0 & 20 & 16 & 12 & 4 & 2 & 20 & 16 & 12 & 4 & 2 \\
2 & 6 & 16 & 20 & 24 & 1 & 4 & 6 & 16 & 20 & 1 & 4 & 6 & 16 & 20 & 0 & 1 & 2 & 6 & 8 & 0 & 1 & 2 & 6 & 8 \\
3 & 4 & 12 & 16 & 20 & 2 & 2 & 4 & 12 & 16 & 2 & 2 & 4 & 12 & 16 & 1 & 0 & 1 & 4 & 6 & 1 & 0 & 1 & 4 & 6 \\
4 & 6 & 8 & 12 & 16 & 3 & 4 & 2 & 8 & 12 & 3 & 4 & 2 & 8 & 12 & 2 & 1 & 0 & 2 & 4 & 2 & 1 & 0 & 2 & 4 \\
10 & 4 & 6 & 4 & 6 & 8 & 3 & 2 & 2 & 4 & 8 & 3 & 2 & 2 & 4 & 6 & 4 & 2 & 0 & 1 & 6 & 4 & 2 & 0 & 1 \\
12 & 5 & 8 & 6 & 4 & 10 & 4 & 3 & 4 & 2 & 10 & 4 & 3 & 4 & 2 & 8 & 6 & 4 & 1 & 0 & 8 & 6 & 4 & 1 & 0 \\
2 & 6 & 16 & 20 & 24 & 1 & 4 & 6 & 16 & 20 & 1 & 4 & 6 & 16 & 20 & 0 & 1 & 2 & 6 & 8 & 0 & 1 & 2 & 6 & 8 \\
3 & 4 & 12 & 16 & 20 & 2 & 2 & 4 & 12 & 16 & 2 & 2 & 4 & 12 & 16 & 1 & 0 & 2 & 4 & 2 & 1 & 0 & 2 & 4 & 6 \\
4 & 7 & 8 & 12 & 16 & 3 & 4 & 2 & 4 & 12 & 3 & 4 & 2 & 8 & 12 & 2 & 1 & 0 & 2 & 4 & 2 & 1 & 0 & 2 & 4 \\
10 & 4 & 6 & 4 & 6 & 8 & 4 & 2 & 2 & 4 & 8 & 3 & 2 & 2 & 4 & 6 & 4 & 2 & 0 & 1 & 6 & 4 & 2 & 0 & 1 \\
12 & 5 & 8 & 6 & 4 & 10 & 4 & 3 & 4 & 2 & 10 & 4 & 3 & 4 & 2 & 8 & 6 & 4 & 1 & 0 & 8 & 6 & 4 & 1 & 0
\end{bmatrix}$$

(27)