Video-Based Risk Assessment for Cyclists

Miguel Costa[1], Manuel Marques[2], João Paulo Costeira[3]

***Abstract*—Due to their zero pollution emissions, health improvements conditions 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 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**

I. Introduction

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. In this context, bicycles appear as a non-pollutable mean of transportation.

Bicycles have been shown to have cardiovascular benefits when practiced for around 30 minutes per day, for 5 days a week [1]. In fact, commuting by bicycle and walking showed that this type of commuting does enhance cardiorespiratory and metabolic fitness of sedentary adults [2].

However, despite such benefits, the number of cyclers have not increased as one would expect. Therefore, it is important to study risk situations that bicyclers face to help city planners plan better and safer bike infrastructures.

In our work we explore precisely this, we gather data directly from cycling journeys to study the risk that cyclers are faced each day, enabling the geographic localization of multiple dangerous scenarios to help urban planners address this issue.

On previous work, the work developed by Vieira [3] focused on developing a tool for elementary events recognition using a smartphone application and ultimately identify stressful events using biological signals.

Fig. 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).

Our approach improves this work by revamping some features of the developed application (app). Using this new version of the application we propose a novel method to identify risky events based solely on video analysis. Taking advantage of optical flow and Focus of Expansion (FOE) estimation we develop a risk descriptor based on the image scene and then assess the level of risk based on different risk criteria (see Fig. 1).

Thus, we highlight the following contributions of our work:

- We improved the previously available Android app. This new app is now able to capture new sets of data signals and developed a backend server enabling the upload of data directly from the smartphone;
- Detection of stressful or risky events 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 work.

Next, we explore related work in Section II. Following, in Section III we describe the improvements made to the smartphone app. In Section IV we describe our risk assessment framework and in Section V we present some results of our developed work. Finally, in Section VI we conclude our work and present some possibilities for future work.

II. Related Work

Our work in focused on three main areas: firstly, cycling, secondly, gathering data using smartphones, and lastly, using computer vision to do risk analysis.

[1] Miguel Costa is a Master student with the ECE department, Instituto Superior Técnico, 1049 Lisboa, Portugal, miguel.n.costa@tecnico.ulisboa.pt
[2] Manuel Marques is a Researcher with the ECE department, Instituto Superior Técnico, 1049 Lisboa, Portugal, manuel@isr.ist.utl.pt
[3] João Paulo Costeira is an Associate Professor with the ECE department, Instituto Superior Técnico, 1049 Lisboa, Portugal, jpcosteira@tecnico.ulisboa.pt

A. Cycling in Urban Areas

As previously mentioned, 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. Vandenbulcke et al. [4], discretize that 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. Policies that address these determinants can result in an increasing bike use. However, issues like the lack of cycling routes, too high or too low traffic volumes and even low quality of traffic signaling 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 [5].

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, [6]. In 2013, the City of Boston release a study [7], 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 [8], [9], where it shows the decrease in risk for cyclers by the implementation of protected bicycle corridors on its main avenues.

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 [10]. 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 to promote tourism [11]. Weston and Mota [12] summarizes what are the principal implications that cycling would take in the sustainability of tourism.

B. Sensing: Getting Data from Smartphones

Today smartphones are cheap and widely used by most population in cities. 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.

In [13] a Context Pyramid with raw sensor data is used to estimate ubiquitous position, recognize motion and human behavior. Mitchel et al. [14] use the smartphone accelerometers to automatically identify a sporting activity. And although Avci et al. [15] 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 [16] tracks users' physical activities and provide feedback in order to foment healthier habits and lifestyle. Su et al. [17] does an overall view on different activity recognitions processes, focusing on data that support the main activity recognition algorithms.

In [18], smartphone's sensors (accelerometer, gyroscope, magnetometer, GPS, video) are used to characteristically divide driving as non-aggressive or aggressive. Eren et al. [19] use sensory data to obtain position, speed, acceleration and deceleration to estimate commuting safety by analyzing driver behavior. More work on analyzing driver's behavior have been done in in [20], 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. [21], 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., [22], 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 [23] accelerometers are used to estimate the vehicle's speed and then sense the traffic volume at a location.

Previous work in [3], also analyzed stress from ECG data by analyzing 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 [3].

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 [3] also records inertial, sound and GPS data, we take advantage of the information only available through the captured images.

C. Computer Vision on Risk Analysis

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 [24], 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 [25] presents the problem solution as a unique neural network pipeline, and thus increasing speed. SSD [26] 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 [27]. 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 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. 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 [28]. More recent works present the problem of solving dense optical flow problems to better get a denser and accurate motion estimation, [29], [30].

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.

III. Data Acquisition System

To gather data from the smartphone an Android app was used. This app was built over the previously made app in [3]. 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 enables the recording of audio and video signals directly from the app.

Additionally, some personal information is also asked when the user first uses the application, such as: Username, Password, Gender, Age, Biking Experience, and whether the bike has a suspension or not. This information is used to create an account on the server that stores the captured data.

This new feature (user registration) is also complemented with the possibility of uploading the recorded data to a server. This allows for a faster, simpler and more organized way to get access to all the recorded data from each user, instead of having to copy all recorded files via a USB connection between the smartphone and the computer as before.

IV. Risk Assessment using the Focus of Expansion

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 is the Focus of Expansion. This point in the image frame represent the convergence of the depicted motion in the image (which 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.

A. Static Scene

Let's first consider the optical case where we have a certain amount of motion described by the cyclist between two consecutive video frames. This motion is resultant only from the cyclist's movement in the scene. In this case, the optical flow calculated is optimal, as it perfectly describes the motion between two frames. Such optimal motion vectors point towards only one location in the image, and the intersection of 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.

Given the motion vectors in the image v_i, with $i = 0, \dots, N$, calculated using the Lucas-Kanade method, 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 $L_i = am + bn + c\}$, with

$a, b, c \in \mathbb{R}$. The distance between point x and L_i can be calculated using (1):

$$f(x, \mathrm{L}_i) = \frac{|am + bn + c|}{\sqrt{a^2 + b^2}} \quad (1)$$

Therefore, to view the point $\tilde{x}$ closest to all lines L_i one can solve the following optimization problem:

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

However, we assume a priori that the optical flow computed across the image has some error associated in its calculations. Due to this we choose to weigh over the computed optical flow vectors. This way we can distinguish some certain 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 center 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 [31]) according to its magnitude and how close it is to the mean of the area they are in. This way, we can discretize the weights m_i we assign to each vector v_i. We assign m_i in accordance to (3). Additionally, if the magnitude is below a certain threshold we assign it the lowest m_i score possible ($m_i = 0.1$).

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

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. Fig. 2 shows a representation of the weights assigned to each vector found, and the division of the image using the concentric circles.

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

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} m_i \cdot f(x, L_i) \quad (4)$$

B. *Dynamic Scene*

Doing this weighting would be enough if we knew that there were no objects in the scene. However, in our scenarios that is rarely the case. In most instances, there are other moving object in the scene like cars, people on the sidewalk or other bikes.

Fig. 2. 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.

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 have to 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 [24] with the image we are analyzing 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 Neural Network (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]).

Thus, 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). So, we weight these vectors that correspond to objects, giving each vector v_i a weight o_i for object k.

$$o_i = e^{-s_k}, \quad (5)$$

where s_k is the confidence score outputted by the NN to a given object k.

Fig. 3. Object associated weights given the confidence score outputted by the NN. Red vectors correspond to $s_k = 0$, blue vectors $s_k = 0.711$, cyan vectors $s_k = 0.741$, and green vectors to $s_k = 0.827$.

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 ($o_i = 1$). Fig. 3 shows a representation of the weights o_i.

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:

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} o_i \cdot f(x, L_i) \tag{6}$$

C. 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. Again, this can be done by solving one optimization problem. We use both previously calculated weights (magnitude weights and object presence weights) to better estimate which vectors are better used in the optimization Plus, we choose to solve the optimization problem using the Huber Loss distance as a distance metric.

This problem can be formulated as in (7) and as in (8), with the contribution of the weights w_i.

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}_\delta(f(x, L_i)) \tag{7}$$

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}_\delta(w_i \cdot f(x, L_i)), \tag{8}$$

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

$$\mathcal{L}_\delta(a) = \begin{cases} \frac{1}{2}a^2 & for\ |a| < \delta, \\ \delta(|a| - \frac{1}{2}\delta) & otherwise. \end{cases} \tag{9}$$

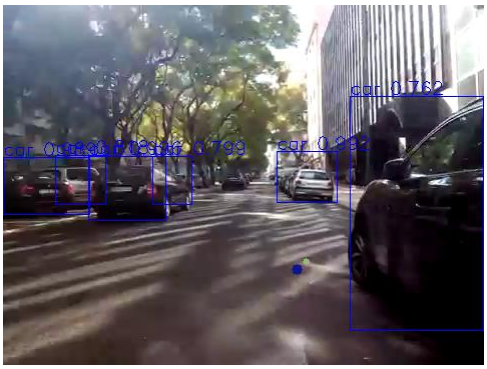
and weights w_i of a given OF vector v_i can be calculated using (10). Note that because both m_i and o_i vary in the interval [0,1], also w_i varies in this interval.

$$w_i = m_i \cdot o_i \tag{10}$$

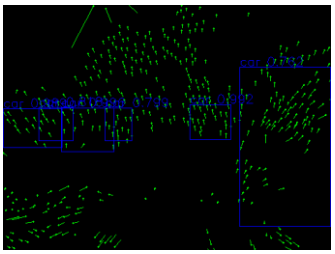
D. Iterative Object Weights Refinement

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 than the one the cyclist is taking. Thus, this re-weighting of the object associated weights is performed taking this in consideration.

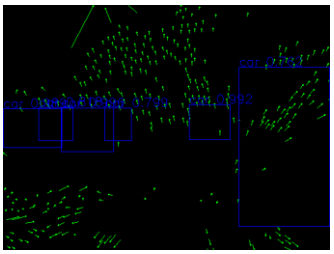
So, we perform an iterative process where each optical flow vector which is calculated on an object is re-weighted (only on the o_i weight) according to its direction. That is, if a vector is pointing away from the $\hat{x}_{FOE}$ and its direction is in line with the FOE, we consider that the object 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). 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. Fig. 4 shows the resulting FOE after this procedure, as well as the vectors weights o_i.



(a)



(b)



(c)

Fig. 4. Results of the refinement of weights o_{v_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).

E. Weighted Average of the Foci of Expansion

Following this, we apply the 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 (11).

$$x_t = \frac{\sum_{j=t-M}^{t} e^{-\tau(t-j)} \cdot \tilde{x}_j}{\sum_{j=t-M}^{t} e^{-\tau(t-j)}}, \tag{11}$$

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.

Fig. 5 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.

Fig. 5. The weighted average of the previous and current FOE (small colored points) and the result of this operation (large red point in the center of the image).

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.

F. 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. We further sub-divide each of the 5 areas on the image into 5 horizontal (see Fig. 1 and Fig. 6), 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 (12)).

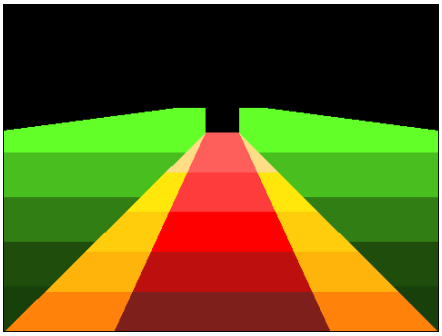


Fig. 6. Division of the 5 different risk zones into 25 sub-regions to promote proximity encoding.

$$d_t = [d_t^1 \quad \cdots \quad d_t^l \quad \cdots \quad d_t^{25}] \tag{12}$$

To compute the value of the risk descriptor d_t^l on sub-region l we use (13), 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.

$$d_t^l = \sum_{k=object_0}^{object_K} r_t^{l,k}, \tag{13}$$

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

$$r_t^{l,k} = \alpha_k \cdot s_k \cdot \gamma_l \cdot \frac{a_t^{l,k}}{b_t^l}, \tag{14}$$

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 described 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 a simpler risk assessment framework. We encompass this as a supervised classification problem, where we use the Earth Mover's Distance (EMD) [32] 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 color 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 neighbors 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 can 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.

V. Results

A. 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

(a) (b) (c) (d)

Fig. 7. Procedure to compute the optical flow: given a frame (a), divide it in 16 (b), apply the CLAHE filter to each and calculate the Shi-Tomasi feature detection (c) and finally compute the optical flow to the previous frame (d).

and to have this points spread out evenly through the whole image we have a different way of discovering these points. Firstly, we start by applying a similar method as the one used in [33]. We start by dividing the image into 16 different sub-images. On each one, we apply a histogram equalization filter (CLAHE [34]), which improves contrast and edge characterization. We then apply the Shi-Tomasi method [35] 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 important point across the image, whereas before all the points could only present themselves in a section of the image and thus compromising the FOE estimation. Fig. 7 show this procedure.

Weights used in the optimization process are calculated as stated before. For the Huber Loss optimization method ((7) and (8)) we used the Python Toolbox *sklearn* and the module *HuberRegressor* [36]. Thus, the result of the optimization step of the Huber Loss distance can be seen in Fig. 8. In Fig. 9 we present a heatmap with the distance between each line L_i and the solution for the optimization problem.

Fig. 8. Example 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.

Fig. 9. 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. Fig. 4 present this step. In fact, as seen, 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 portion of its optical flow vectors). In image frames where there are many objects that occupy a good portion of the image the 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 regarding the previous iteration. 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.

Finally, the last step consists in performing the weighted average of the current estimate for the FOE and the estimates of the previous frames. Results for this can be seen in Fig. 5.

B. 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 Fig. 6) 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

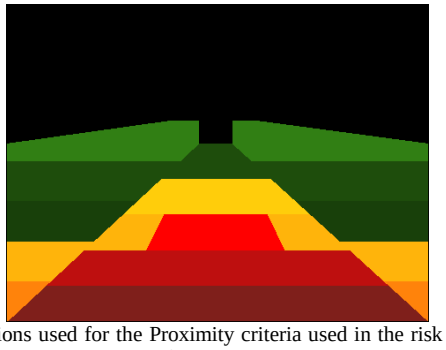


Fig. 10. Regions used for the Proximity criteria used in the risk assessment. The colours indicate the level of risk: red – highest risk level (3); yellow – intermediate risk level (2), and; green – lowest risk level (1).

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 Fig. 10) and again define the levels of risk as before.

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 neighboring 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). In order to solve the EMD, we used the Python Toolbox *pyemd* [37], [38].

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

$$\alpha_k = \begin{cases} 1.0 & if \text{ object } k \text{ is motorized,} \\ 0.8 & if \text{ object } k \text{ is a bike,} \\ 0.6 & if \text{ object } k \text{ is a person.} \end{cases} \quad (15)$$

Furthermore, the region and sub-region factor used in (14) is defined as

$$\gamma_l = \varphi_l \cdot \psi_l \,, \quad (16)$$

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

$$\varphi_l = \begin{cases} 1.0 & if \text{ region } l \text{ is of risk level 3,} \\ 0.75 & if \text{ region } l \text{ is of risk level 2,} \\ 0.30 & if \text{ region } l \text{ is of risk level 1.} \end{cases} \quad (17)$$

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 this box better projects the object on the ground and in the risk zones (which correspond to regions on the ground).

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.

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.

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

VI. 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.

In the future, 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.

REFERENCES

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

[2] 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.

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

[4] 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.

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

[6] 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.

[7] City of Boston, "Cyclist Safety Report," City of Boston, Boston, 2013.

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

[9] 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].

[10] 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].

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] D. Ballard and C. Brown, "Focus of Expansion," in Computer Vision, Prentice Hall, First Edition, May 1982, p. 199.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] M. Grundmann, V. Kwatra, D. Castro and I. Essa, "Calibration-Free Rolling Shutter Removal," in IEEE ICCP, 2012.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.