Use of Camera Sensors to Deduce Relative Velocity of Bicycles to Prevent Accidents

FULVIO NARDI DEI DA FILICAIA DOTTI, University of Twente, The Netherlands

With the growth of e-bikes on bicycle lanes, there is a growing risk factor involving accidents related to bicycle cruising with a large difference in speed. This paper explores technologies, algorithms and methods involved in calculating and deducing the relative velocity from the perspective of a camera mounted on a bicycle and detecting imminent danger. The algorithms in question are three and involve the use of geometric proportions, polynomial fitting and stereo vision respectively. A mathematical background and accuracy evaluation based on all three methods are provided in the research.

Additional Key Words and Phrases: Computer Vision, Stereo Vision, Sensors, Relative Velocity

1 INTRODUCTION

In the Netherlands, there has been a growing popularity of e-bikes. In 2013 the number of e-bikes was approximately 1.2 million, in 2017 was 2 million resulting in a growth of 74% over the course of four years [4]. According to [8] the leading cause of accidents between two vehicles is the relative distance and difference in velocity. Calculating the relative velocity of other bicycles is essential in order to increase the safety of the bikers and e-bikers. Alerting dangerous relative velocity can prevent sudden crashes between a slow rider and a fast rider if the distance between the two is not large enough [3].

In order to alert imminent biking danger this paper replicates and expands the possible algorithms used by smart cars capable of using computer vision in order to deduce the relative velocity of other vehicles in the same driving lane.

Although the technology to calculate the relative velocity of other vehicles in the same lane already exists, this paper provides insight into the implications of applying algorithms to calculate the relative velocity of bicycles and the implications of implementing those algorithms into a bicycle. In addition, this paper evaluates and compares the performance and accuracy of a single camera relative velocity algorithm and a stereo vision algorithm in order to determine the best way to calculate the relative velocity of other bicycles.

The algorithms that are evaluated are **Calculated Bicycle Velocity Based on Distance Equation**, **Calculated Bicycle Velocity Based on Fitted Polynomial Distance Equation** and **Calculated Bicycle Velocity Based on Stereo Vision**.

2 PROBLEM STATEMENT

With an increasing number of e-bikes, there is a risk of the speed difference between an e-bike and a regular bicycle. One of the most common causes of an accident between a bicycle and an e-bike in the same cycling lane is the speed difference between the former two [14].

Therefore, bicycle accidents due to speed differences are a risk that should be addressed due to the increasing number of e-bikes. The risk of an accident is especially high between two bicycles (or e-bikes) if the difference in velocity is high and the relative distance between two is short [8].

This paper attempts to design a bicycle relative velocity detection system for short distances in order to detect and alert a high difference in velocity between two bicycles that has a high risk of causing a collision. The system is intended to be an embedded system mounted on bicycles in order to detect danger in real time using computer vision.

2.1 Research Question

In order to tackle the problem statement the following question with corresponding sub-questions has been formulated.

1. To what extent can vision-based solutions accurately detect dangerous situations between two bicycle riders cycling at varying speeds?
   - (a) What is the software and hardware development process for developing a vision-based relative velocity detection system for bicycles?
   - (b) To what extent is an algorithm using stereo vision more accurate than a simple bounding box projection into a 3D world coordinates algorithm in terms of relative velocity calculation?

3 RELATED WORK

Authors [5, 11] provide mathematical background and an equation to find the distance of an object given its height. Distance inferred from [5, 11] will translate into velocity by measuring the displacement Δd over a period of time Δt. Thus, comparing the assumed true height of a bicycle with the perceived height in the camera will yield an estimated distance of the bicycle rider from the camera sensor.

Another algorithm that this research takes into consideration for deducing the distance of a bicycle rider from a monocular camera is the application of a polynomial function. The polynomial function is estimated and fitted by collecting real data and uses the perceived dimension of the object in the camera as the independent variable and the true distance of the object as the dependent variable [13].

Research involving stereo vision is already mature and well-documented. This article takes into consideration the mathematical principle and methodology applied by other works [7, 9, 10, 12] in order to achieve a 3-dimensional depth estimation of a region of interest.

4 METHODS OF RESEARCH

This research covers the hardware aspect of the project in section 6, the implementation methods considered for the project in section 7, the testing procedure in section 8 and finally future work and conclusion in sections 10 and 11 respectively.

TScIT 38, February 3, 2023, Enschede, The Netherlands

The "Hardware Setup" section consists in finding, utilizing and assembling the most suitable hardware tailored for the research data collection and algorithms evaluation.

"Methods" covers the implementation techniques for the bicycle detection algorithm and bicycle object tracking. In addition, it presents the mathematical background and limitations for each algorithm presented to calculate the relative velocity of the target bicycle.

"Testing" is the last process of this research and consists of an evaluation of the accuracy of each algorithm with respect to the actual relative velocity of the bicycles. This stage will be performed with speedometers to detect the true speed of target bicycles.

5 EXPECTED RESULTS

According to the results from the experiment already conducted from [5] the mean error of the uncalibrated camera to measure depth is 6.25% using the depth equation 2. Therefore, it is expected that the result using equation 2 will have an accuracy between 85% and 95% taking into consideration the uncertainty of a moving bicycle camera focus.

Using the polynomial fitting approach suggested by [13] an accuracy as high as 98.76% is achievable. Therefore it will be expected that the experiment using this method will have an accuracy to estimate the distance between 90% and 97%.

Finally, according to "An Improved Sum of Squared Difference Algorithm for Automated Distance Measurement" the achieved accuracy for calculating object distance using stereo vision was 95%. For this research, it will be assumed an accuracy between 85% and 95%.

Even though an estimated accuracy was drawn for all 3 methods it is still hard to predict the actual accuracy that will be achieved throughout the research in order to estimate relative velocity. All the methods mentioned used different experiment environments and applying them on a moving bicycle may yield a completely different result than expected.

5.1 Project Requirements and Goals

The requirements and goals for this project are listed below.

(1) Reach the accuracy mentioned in the expected result section 5.
(2) Build a suitable hardware setup capable of running and collecting data from the bicycle velocity algorithms.
(3) Answer the research questions in subsection 2.1

6 HARDWARE SETUP

The hardware used for the project involves an off-the-shelf Jetson Nano and respective modules required to detect and process bicycles through computer vision technology. Such modules are shown in figure 1 with the corresponding index listed below.

(1) **Jetson Nano** to handle the processing of algorithms 7.3, 7.4 and 7.5.
(2) **Raspberry Pi Noir Camera V2** to capture frames for algorithms 7.3 and 7.4.
(3) Second **Raspberry Pi Noir Camera V2** to add perspective in order to allow Stereo Vision Algorithm in 7.5.

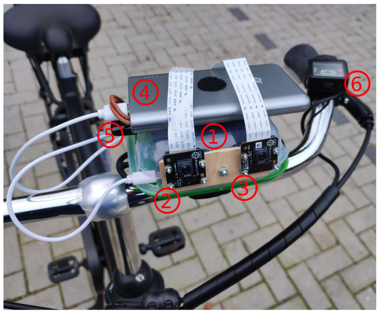

Fig. 1. Hardware Required to Process Relative Velocity

(4) **Power Bank** to aliment Jetson Nano during testing.
(5) **WiFi Module** to exchange data and start algorithms through Remote Desktop control.
(6) **Bicycle Speedometer** to read ground truth velocity data of the target bicycle.

7 METHODS

The methods involved in calculating the relative velocity of the target bicycle can be summarized in the following four steps: bicycle detection, bicycle distance calculation, bicycle tracking and filtering. An overview of these steps can be seen in figure 2 where bicycle distance calculation is split into the three submethods geometric equation, polynomial fitting and stereo vision approach.

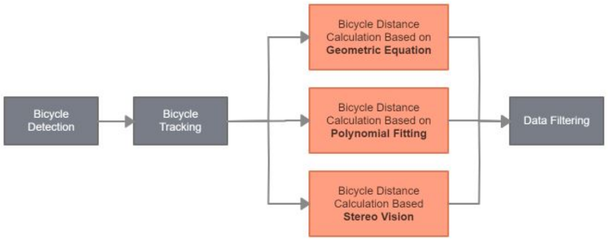

Fig. 2. Processing of Input Camera Image

7.1 Bicycle Detection

The object detection model used for the project is *ssd-mobilenet-v2*. *Ssd-mobilenet-v2* contains ninety labels, however, all of the labels except for the "bicycle" are ignored. For the entirety of this project, the detection threshold for bicycles was set to 0.4 confidence in order to decrease the chances of data loss due to undetected bicycles.

Finally, bounding box height is used to calculate the distance estimation of the bike using a geometric formula and polynomial fit approaches.

7.2 Bicycle Tracking

Bicycle tracking keeps track of the velocity measurement of every bicycle object detected in a given frame. Whenever a new frame is presented a simple ID assignment for every bicycle in the frame is applied. The ID applied to the detected bicycle object determines its row entry in a 2-dimensional array.

The column entry in the 2-dimensional keeps track, for a given bicycle, the previously calculated distance d_p and the current computed distance d_c. In addition, the column entry of the 2-dimensional array keeps track of the last s calculated velocities as shown in figure 3.

The distance is measured periodically between intervals of a few milliseconds and the velocity (between two bicycles) is thus calculated by evaluating the displacement $\Delta d = d_c - d_p$ over the interval of time Δt as seen in equation 1.

$$v = \frac{\Delta d}{\Delta t} \tag{1}$$

The resulting velocity v is placed in the last position V_s after the values in the array are shifted to the left by one position from V_s to V_1. The final output velocity is calculated using the Average Moving Window equation 11 from 7.6.1 with the values from V_1 to V_s.

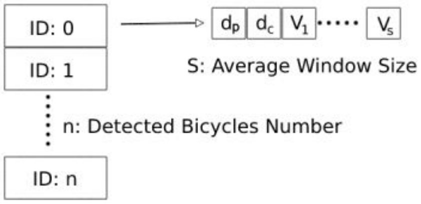

Fig. 3. Bicycle Tracking Logic and Data Structure

7.2.1 Bicycle Tracking Limitations. The bicycle tracking logic works best when bicycles do not cross each other in the x-axis of the frame. When bicycles cross each other on the x-axis the IDs in the row entries get swapped in the 2-dimensional array. In turn, the new bicycle's velocity reading overwrites the previous bicycle's reading resulting in a temporary inaccuracy of the relative velocity calculation for the two crossing bikes.

7.3 Calculated Bicycle Velocity Based on Distance Equation

According to [5] it is possible to apply equation 2 to derive the object's distance from the camera using its perceived pixel height.

$$d = \frac{f_{mm} H_{O_m} H_{Img_p}}{H_{O_p} H_{S_{mm}}} \tag{2}$$

"where f_{mm} is the focal length in the camera in mm, H_{O_m} is the object height in meters, H_{Img_p} the height of the frame in pixels, H_{O_p} the object in pixels, and $H_{S_{mm}}$ sensor size in mm." [5]

It is therefore possible to calculate the distance of a bicycle by applying equation 2 with the known specification of the Raspberry Pi Noir Camera V2. According to [1] the focal length (f_{mm}) of a Raspberry Pi Noir Camera V2 is 3.04 mm and its camera sensor height ($H_{S_{mm}}$) is 2.76 mm. The assumed true height of the target bicycle (H_{O_m}) detected through the camera is 1.1 m from the bottom of the wheel to the top of the saddle and the pixel height of a frame ($H_{O_{imgp}}$) is of 720. Using the value specification of the Raspberry Pi Camera V2 we can simplify equation 2 onto equation 3.

$$d \approx \frac{872.347}{H_{O_p}} \tag{3}$$

Using equation 3 we can read and plug in the pixel height of the bounding boxes present in the frame in order to calculate the distance of the detected bicycles in meters.

7.3.1 Distance Equation Limitations. The distance equation 2 uses assumed standard bicycle height in order to calculate its distance to the camera. Therefore, bicycle distance will be more accurate the closer the bicycle frame height is to the assumed height. Bicycle frames that are shorter than the assumed height will appear further away than their actual distance while bicycles that are taller than the assumed height will appear closer to the camera than their actual distance.

7.4 Calculated Bicycle Velocity Based on Fitted Polynomial Distance Equation

Following a similar approach as [13] this paper proposes a mapping between the pixel height of an object to its physical distance in meters. To map the bicycle distance, sixteen frames are captured using an iPhone 13 Pro camera from the rear angle of a bicycle from various distances. In each frame, the pixels are counted from the bottom of the rear bike to the top of the saddle to simulate the height measurement of a bounding box enclosing a bicycle. Using the bicycle height (px) as the independent variable and the distance (m) of the bicycle from the camera as the dependent variable, we can plot the relationship in an x-y plane as shown in 4. Using MATLAB it was possible to find the line of best-fit equation 4 that best represents the plotting between the bicycle height (px) and distance (m) of the bicycle to the camera.

$$d_p = 0.0021 \times \varphi^2 - 2.2475 \times \varphi + 714.7792 \tag{4}$$

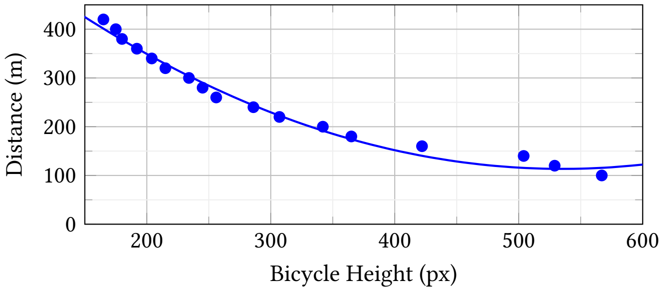

Fig. 4. Relationship Between Bicycle Pixel Height and Distance

Where d_p is the distance between the camera and the bicycle in meters and φ is the pixel height of the bicycle using an iPhone 13 Pro. Therefore plugging in the bicycle height in pixels, from the perspective of an iPhone 13 Pro camera to equation 4 will produce the distance between the target bicycle and the camera.

7.4.1 Translating Raspberry Pi V2 Camera Pixel Height to iPhone 13 Pro Camera Pixel Height. Because the pixel height of the bicycle was measured using frames captured from an iPhone 13 Pro camera the bicycle pixel height frames captured from a Raspberry Pi V2 camera will not produce the correct distance when plugged into equation 4.

We propose a method to translate the pixel height of the bicycle from a Raspberry Pi Noir Camera V2 to the camera of an iPhone Pro 13 in order to utilize formula 4 using the hardware mentioned in section 6.

Plugging the value from table 1 into formula 3 we get a set of equations 5 which ultimately allow deriving the relationship between the pixel height from iPhone 13 Pro Camera $H_{O_{iphone}}$ and the pixel height from Raspberry Pi Camera $H_{O_{raspberry}}$ in equation 7.

Varible	iPhone 13 Pro	Pi Camera V2
f_{mm}	5.70	3.04
H_{Img_p}	1080	720
$H_{S_{mm}}$	7.76	2.76

Table 1. Corresponding Values for Raspberry Pi Camera V2 and iPhone 13 Pro Camera.

$$\begin{cases} d = \frac{3.04 \cdot H_{O_m} \cdot 720}{H_{O_{raspberry}} \cdot 7.7} \\ d = \frac{5.70 \cdot H_{O_m} \cdot 1080}{H_{O_{iphone}} \cdot 7.76} \end{cases} \tag{5}$$

$$\frac{5.70 \cdot H_{O_m} \cdot 1080}{H_{O_{iphone}} \cdot 7.76} = \frac{3.04 \cdot H_{O_m} \cdot 720}{H_{O_{raspberry}} \cdot 7.7} \tag{6}$$

$$H_{O_{iphone}} \approx 2.79 \cdot H_{O_{raspberry}} \tag{7}$$

Therefore, we can use equation 7 in order to obtain the bicycle's pixel height from the perspective of an iPhone 13 Pro by plugging the read pixel height from the perspective of the Raspberry Pi Camera V2.

In order to calculate the distance of the target bicycle using equation 4 the following steps need to be taken into consideration:

1. Capture frame using Raspberry Pi V2 Camera.
2. Detect and measure the bounding box pixel height of bicycles.
3. Convert pixel height measured with Raspberry Pi V2 Camera to iPhone 13 Pro pixel height using equation 7.
4. Plug converted height into equation 5.

7.4.2 Polynomial Fitted Distance Equation Limitations. Similarly to 7.3 the polynomial fitted equation assumes the standard height of the target's bicycle in order to calculate its distance from the camera. In fact, the mapping between bicycle pixel height and distance from the camera was performed using a 1.1 meters tall bicycle. In addition, the accuracy of the calculated distance is heavily dependent on the correctness of the chosen model representing the plotted mapping between bicycle height and distance.

7.5 Calculated Bicycle Velocity Based on Stereo Vision

Stereo Vision image capturing is a method used to determine the 3D physical properties of a scene using two parallel-placed cameras [12]. In the case of this project, the distance between the camera and the target bicycle was deduced using two Raspberry Pi Cameras V2 placed 5 cm distant from each other.

7.5.1 Background. Stereo vision distance measurement is possible by applying the following equation 8 using the principle of similar triangles between triangle O_1, O_2, P and triangle x_1, x_2, P in figure 5.

$$\frac{b}{Z} = \frac{b + x_1 - x_2}{Z - f} \tag{8}$$

Where the baseline b is the distance between two cameras O_1 and O_2, x_1 and x_2 are the pixel distances of a point P from the centre of the respective cameras, Z is the distance between the object and cameras and f is the focal distance as shown in figure 5.

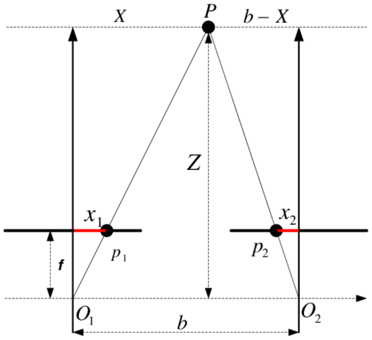

Fig. 5. Stereo Vision Triangulation Scheme [6]

Rearranging equation 8 it is possible to deduce equation 9 solving for the distance Z.

$$Z = \frac{f \cdot b}{x_2 - x_1} \tag{9}$$

Now, the only values required to compute the distance Z are the focal length f of the two cameras, the baseline b and the parallax pixel offset between the two points x_2 and x_1.

7.5.2 Calibration. Calibration is a required step for rectifying the two frames from the perspective of each camera. Rectification is the transformation of each image plane such that a point in the 3D space is mapped onto the same corresponding horizontal line and is mainly used to simplify the stereo correspondence problem [7, 9].

Calibration is performed by processing thirty images of a checkerboard pattern with known dimensions as shown in figure 6. During the calibration process, the following aspects need to be taken into account for better results.

- In each calibration picture the checkerboard should be tilted at various angles.
- The checkerboard should be maintained at a relatively short distance from the cameras.

- Each box should be visible from both of the cameras.
- Both cameras must have as similar tilt angle as possible.

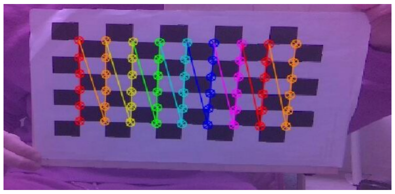

Fig. 6. Checkerboard for Cameras Rectification

7.5.3 Depthmap. A depthmap is a 2D representation of depth given an image; a function that translates the pixel of an image to a color that represents its distance in the 3D space as shown with equation 10 [2].

$$Color = 255 - \frac{(depth * 255)}{maxdepth} \tag{10}$$

According to [2] "the tasks required for creation of Depth-map are: (i) Capturing Images, (ii) Image Preprocessing, (iii) Depth Estimation, and (iv) Calculation of color value for all pixels." The result of the depthmap for distance calculation for my project can be shown in the figure 7. Finally, the depthman can be used in conjunction with object detection models in order to deduce the distance of a given detected object.

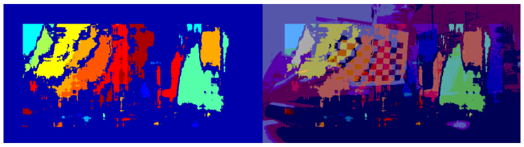

Fig. 7. Depthmap Representation with Jetson Nano

7.5.4 Stereo Vision Limitations. The limitations of the Stereo Vision algorithm are the requirement of two identical cameras, its calibration process and the accuracy of long distances. In fact, Stereo Vision accuracy is dependent on the quality of the calibration process and the baseline length. With a greater baseline, the parallax pixel offset is more noticeable making it more accurate in determining the distance of the target object.

7.6 Filtering

Due to the unpredictable and uncertain nature of the object detection models, it is possible that bounding boxes get assigned distorted and incorrectly to the intended object. In turn, distorted bounding boxes will lead to anomalous data calculating unrealistic velocity measurements of the target bicycle. To prevent outliers of this kind two methods are presented which consist of the Average Moving Window and Threshold Noise Filtering.

7.6.1 Average Moving Window. The Average Moving Window takes into consideration a set of consecutive measurements n and finds their average in order to smooth out the readings and outliers. The Average Moving Window can be represented with equation 11.

$$AWS = \frac{1}{n} \sum_{i=k-n+1}^{k} V_i \tag{11}$$

Where k is the total observed values and V_i are the read values. In the case of this project, V_i is the observed relative velocity of the target bicycle at frame i and the window size n was set as five.

7.6.2 Threshold Noise Filter. Another method used for filtering outliers is the use of Threshold Noise Filtering. By comparing two consecutive measurements we evaluate whether the difference in value between the two is not higher than a predefined threshold value. In the case of this project, I set the threshold value to 0.7 and applied threshold noise filtering to the last two calculated average velocities using 7.6.1. To calculate the difference index between two values I use equation 12.

$$\Delta V = \frac{|V_k - V_{k-1}|}{\frac{|V_k| + |V_{k-1}|}{2}} \tag{12}$$

Where V_k and V_{k-1} are the last two velocity measurements.

8 TESTING

Testing involves a comparative analysis of relative velocity calculation accuracy using the distance based on bicycle height, distance based on a fitted polynomial equation and distance based on stereo vision algorithms.

For each distance calculation method, three velocities are tested namely **3 km/h**, **6 km/h** and **9 km/h** repeated twice each.

8.1 Testing Conditions

Testing was conducted in a flat parking area with perfect weather conditions in order to maximise target bicycle visibility. In addition, the Jetson Nano processor and its required modules were placed on a tripod in a stationary position as shown in figure 8.

Using the bicycle's speedometer the bicycle was brought at a constant velocity of 3 km/h, 6 km/h and 9 km/h for a distance of about ten meters two trials for each velocity calculation algorithm. The data collected from the test are summarized in table 2. Where, for each distance algorithm, the average velocity over the course of ten meters is reported as $\hat{V}_1$ for the first trial and $\hat{V}_2$ for the second trial.

Ground Truth	**Distance Eq.**		**Polynomial Fit**		**Stereo Vision**	
V km/h	$\bar{V}_1$ km/h	$\bar{V}_2$ km/h	$\bar{V}_1$ km/h	$\bar{V}_2$ km/h	$\bar{V}_1$ km/h	$\bar{V}_2$ km/h
3.00	2.51	2.27	2.37	2.07	4.64	4.89
6.00	3.71	4.35	3.26	4.31	6.99	5.22
9.00	5.67	5.74	3.52	3.83	8.17	6.59

Table 2. Average Velocity Result Based on Applied Method

Fig. 8. Testing Setup

8.2 Evaluation Metrics

To evaluate the accuracy of each method 7.3, 7.4, 7.5 we use the mean absolute error equation 13 on both iterations of the test for each expected velocity (3 km/h, 6 km/h and 9 km/h).

$$\Delta V_{mean} = \frac{1}{2}\sum_{i=1}^{2} |\bar{V}_i - V| \tag{13}$$

Where ΔV_{mean} is the mean absolute error given the velocity and i is the iteration, which is either iteration 1 or 2 from table 2. We can then calculate the percentage relative error with equation 14.

$$\delta V_{mean} = \frac{\Delta V_{mean}}{\bar{V}_{mean}} \times 100 \tag{14}$$

Where δV_{mean} is the the relative error given then velocity and $\bar{V}_{mean}$ is the mean between $\bar{V}_1$ and $\bar{V}_2$. Finally, the average velocities, absolute errors and relative errors are reported in table 3.

Ground Truth	**Distance Eq.**		**Polynomial Fit**		**Stereo Vision**	
V km/h	ΔV_{mean}	δV_{mean} %	ΔV_{mean}	δV_{mean} %	ΔV_{mean}	δV_{mean} %
3.00	0.61	**25.60**	0.78	**35.14**	1.76	**36.94**
6.00	1.97	**48.89**	2.22	**58.65**	0.88	**14.41**
9.00	3.30	**57.84**	5.33	**145.03**	1.62	**21.95**

Table 3. Absolute and Relative Errors for Given Velocity

8.3 Error Sources

8.3.1 *Distance Equation Error Source.*

The source of error for the distance equation approach is due to the moving average window filter and low time interval. Since the moving average filter takes the average of the last five measured velocities, the filter will take into consideration values read from the previous trials. Even when the bicycle is not inside the frame the average moving window value will not reset. Therefore, if the average moving window array already contains values from the previous trial they will be taken into consideration during the current trial.

Because of a low time interval between one frame and another, any inaccuracy in the time interval Δt would cause the velocity to be inaccurate as well. One way to offset this error would be to increase the time interval between frames or by multiplying the velocity by an arbitrary factor to correct the error.

8.3.2 *Fitted Polynomial Equation Error Source.*

Fitted Polynomial Equation has the highest relative error $(35.14 - 145.03\%)$. The cause of the high error percentage may be due to the wrong-fitted equation used to estimate the relationship between pixel height and object distance. In fact, in figure 4, the line is flat between 600 and 500 pixels on the x-axis. The flat nature of the line would explain the low value in velocity; from pixel height 600 to pixel height 500 the object appears to have the same distance. In other words, there is no displacement between pixel height 600 and pixel height 500 causing the bicycle to appear stationary.

8.3.3 *Stero Vision Error Source.*

Even though Stereo Vision has a relatively low relative error $(14.41 - 36.94\%)$ it still does not meet the expected result from section 5. The causes of inaccuracy in Stereo Vision may be due to poor calibration, an unaligned array of cameras or a too narrow baseline. In fact with a greater baseline length, it is easier for the Stereo Vision to pick up parallax pixel offset which, in turn, makes it easier to deduce the distance of an object.

9 ADJUSTED CALCULATED BICYCLE VELOCITY BASED ON THE DISTANCE EQUATION

In order to compensate for the error sources in 8.3 we present an adjusted Calculated Bicycle Velocity Based on Distance Equation from subsection 7.3. In fact, it is apparent from 8.2 that the calculated relative velocity using the distance equation 2 is smaller than the target expected velocity by a factor of around 1.5. We, therefore, attempt to reevaluate the distance equation approach by multiplying the calculated velocity by a factor of 1.5.

9.1 Testing the Adjusted Calculated Bicycle Velocity Based on The Distance Equation

The testing conditions for this method are the same as used in section 8.1. However in this case, only one trial is evaluated and the tested velocities are **6 km/h**, **9 km/h** and **14 km/h**.

9.2 Results of the Adjusted Calculated Bicycle Velocity Based on The Distance Equation

The result and evaluation for the adjusted calculated bicycle velocity based on the distance equation can be found in table 4 where V is the expected ground truth velocity, $\hat{V}$ is the average measured velocity, ΔV is the absolute error and δV is the relative error. Results from table 4 show that the relative error is withing 8.83 and 12.57% which is within the expected accuracy mentioned in section 5.

Target	**Adjusted Distance Eq.**		
V km/h	$\bar{V}$ km/h	ΔV	δV %
6.00	6.53	0.53	8.83
9.00	10.00	1.00	11.11
14.00	12.24	1.76	12.57

Table 4. Absolute Error and Relative Error of Adjusted Calculated Bicycle Velocity Based on Distance Equation

10 FUTURE WORK

One of the future implementations for calculating bicycle relative velocity using the distance equation and fitted polynomial distance equation approach is **detecting the target's bicycle wheel height** instead of the target's bicycle frame height. In fact, the standard average height of bicycle wheels in the Netherlands is 28 inches. Applying "velocity calculation based on distance equation" and "velocity calculation based on fitted polynomial equation" on the bicycle's wheel would lead to more consistent results and less variation compared to bicycle frame size. Training a bicycle wheel detection model tailored for accurately finding their height would decrease the chances of missing detectable objects and would increase consistency in finding accurate relative velocities of target bicycles.

Another limitation mentioned in subsection 7.2.1 is the potential data loss due to the basic bicycle tracking logic implemented in section7.2. **Implementing algorithms such as the appearance model** for tracking bicycles would increase accuracy in tracking the velocity of multiple bicycles in the same frame.

Inaccurate velocity readings due to bounding boxes cut from the edge of the frame could also be solved in a future iteration. Some approaches to tackle this issue involve **the use of the bounding box width to calculate object distance** or **the use of cameras capable of wide angles frame capturing.**

Finally, **combining stereo vision and distance equation approaches** should lead to a more accurate reading of the relative velocity of target bicycles. This can be done by outputting the average readings between stereo vision and distance equation approaches or by alternating the use of them depending on their optimal range in terms of accurate readings.

11 CONCLUSION

This research explored three methods for calculating the relative velocity of bicycles and are based on a distance equation, fitted polynomial distance equation and stereo vision respectively. According to the results collected the stereo vision algorithm managed to score the highest in terms of relative velocity calculation accuracy. The reasons for this result can be attributed to the stereo vision's independence of bicycle height measurement. In other words, the stereo vision algorithm does not need to assume the height of the bicycle in order to deduce its depth in the 3D space. In fact, the stereo vision algorithm managed to score an accuracy error of 24.43% on average when calculating bicycle relative velocity. On the other hand, the relative velocity calculation based on a distance equation scored an accuracy error of 34.11% while the relative velocity calculation based on a fitted polynomial equation scored an accuracy error of 79.61%.

Nonetheless, monocular camera algorithms can surpass stereo-vision algorithms in terms of accuracy if the correct parameters are applied. In fact, the relative velocity calculated using a distance equation managed to score an accuracy error of 10.84% applying the appropriate adjustments.

ACKNOWLEDGMENTS

I would like to thank my project supervisor Akhil Pallamreddy for providing me with all the necessary tools and help during the development and writing of this research project.

REFERENCES

[1] [n. d.]. Raspberry Pi Documentation. https://www.raspberrypi.com/documentation/accessories/camera.html

[2] Asra Aslam and Mohd. Samar Ansari. 2019. Depth-Map Generation using Pixel Matching in Stereoscopic Pair of Images. https://doi.org/10.48550/ARXIV.1902.03471

[3] Lu Bai, Pan Liu, Yuguang Chen, Xin Zhang, and Wei Wang. 2013. Comparative analysis of the safety effects of electric bikes at signalized intersections. *Transportation Research Part D: Transport and Environment* 20 (2013), 48–54. https://doi.org/10.1016/j.trd.2013.02.001

[4] Mathijs de Haas, Maarten Kroesen, Caspar Chorus, Sascha Hoogendoorn-Lanser, and Serge Hoogendoorn. 2022. E-bike user groups and substitution effects: evidence from longitudinal travel data in the Netherlands. *Transportation* 49 (6 2022), 815–840. Issue 3. https://doi.org/10.1007/s11116-021-10195-3

[5] Sotirios Diamantas, Stefanos Astaras, and Aristodemos Pnevmatikakis. 2016. Depth estimation in still images and videos using a motionless monocular camera. *IST 2016 - 2016 IEEE International Conference on Imaging Systems and Techniques, Proceedings*, 129–134. https://doi.org/10.1109/IST.2016.7738210

[6] Du, Muslikhin, Hsieh, and Wang Zhaoli. 2020. Stereo Vision-Based Object Recognition and Manipulation by Regions with Convolutional Neural Network. *Electronics* 9 (01 2020), 210. https://doi.org/10.3390/electronics9020210

[7] Andrea Fusiello. [n. d.]. Tutorial on Rectification of Stereo Images Tutorial on Rectification of Stereo Images. http://www.dai.ed.ac.uk/CVonline/.1998.

[8] Joko Hariyono, Ajmal Shahbaz, Laksono Kurnianggoro, and Kang-Hyun Jo. 2016. Estimation of collision risk for improving driver's safety. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 901–906. https://doi.org/10.1109/IECON.2016.7793743

[9] Pasquale Lafiosca and Marta Ceccaroni. 2022. Rectifying homographies for stereo vision: analytical solution for minimal distortion. (2022). https://doi.org/10.48550/ARXIV.2203.00123

[10] Yue Lin, Yixun Gao, and Yao Wang. 2021. An Improved Sum of Squared Difference Algorithm for Automated Distance Measurement. *Frontiers in Physics* 9 (8 2021). https://doi.org/10.3389/fphy.2021.737336

[11] Robert McCraith, Lukas Neumann, and Andrea Vedaldi. 2021. Real time monocular vehicle velocity estimation using synthetic data. *IEEE Intelligent Vehicles Symposium, Proceedings* 2021-July, 1406–1412. https://doi.org/10.1109/IV48863.2021.9575204

[12] Y. M. Mustafah, A. W. Azman, and M. H. Ani. 2013. Object distance and size measurement using stereo vision system. *Advanced Materials Research* 622, 1373–1377. https://doi.org/10.4028/www.scientific.net/AMR.622-623.1373

[13] Ashfaqur Rahman, Abdus Salam, Mahfuzul Islam, and Partha Sarker. 2008. An image based approach to compute object distance. *International Journal of Computational Intelligence Systems* 1 (2008), 304–312. Issue 4. https://doi.org/10.1080/18756891.2008.9727627

[14] Chen Rongbao, Zhao Dan, and Wang Qianlong. 2017. Research on the algorithm of the front vehicle speed based on vehicle video. *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, 7719–7724. https://doi.org/10.1109/CCDC.2017.7978591