IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Vision-Based Estimation of Motorcycle Attitude

Obaida Alrazouk[1], Amine Chellali[1], Hicham Hadj-Abdelkader[1], and Hichem Arioui[1]

***Abstract*— This paper presents a novel, efficient, vision-based technique to estimate the attitude of Powered Two-Wheeled Vehicles (P2WV) with respect to the road surface. This method is based only on a monocular camera fixed on the vehicle. The estimation is done in two steps. The first step relies on a novel technique that utilizes perspective distortion and Inverse Perspective Mapping to measure the sub-pixel width difference of the lane markers and estimate the roll relative angle. The second step uses the vanishing point technique from the lane markers to estimate the yaw and the pitch. These last are then converted to the same Euler convention, and the rotation matrix of the camera is recovered. This permits estimating the vehicle attitude.**

The presented method was tested on several scenarios with different high speeds and road shapes using the BikeSim framework. The results show a very low error of estimation and robustness against camera pose change. The main advantage of this method is that the estimation is not accumulative, thus eliminating the estimation drift error over time.

I. INTRODUCTION

Motorcycling is an exhilarating and rewarding activity but entails certain risks and hazards. The National Highway Traffic Safety Administration (NHTSA) reports that motorcyclists are 28 times more likely than passenger car occupants to die in a crash per vehicle mile traveled [1].

While following safety guidelines can help riders reduce the risk of accidents, they are not always sufficient to prevent them. Human errors, distraction, and other factors can impair the rider's ability to react quickly to hazards on the road. Therefore, developing rider assistance systems, that can enhance his awareness and performance, is essential.

Rider assistance systems can help motorcyclists avoid collisions by responding to various riding situation. However, to do so, these systems need to measure the motorcycle's attitude (roll, pitch, and yaw angles) relative to the road. These rotations affect the motorcycle's stability, balance, and maneuverability and must be accurately estimated.

To estimate the motorcycle attitude, two main approaches exist: model-based and vision-based techniques. The model-based techniques [3], [2], [4] rely on the use of the vehicle's dynamic or kinematic models for the differential state estimator. This makes the estimation directly related to the intricacy and the complexity of the model and its parameters. These models require understanding of wheel interaction with the road, adding to the computational burden due to their highly non-linear nature. This complexity poses challenges in real-time scenarios.

Vision-based techniques [9], [7] rely on the principle of visual odometry, using cheap and widely available camera sensors. They estimate the translation and rotation of the camera using some visual features of the scene or the road texture. While these methods have been shown to be accurate in short scenarios, they also suffer from error accumulation over time since they count on incremental localization. In addition, feature extraction, matching, and outlier rejection make vision-based techniques computationally expensive.

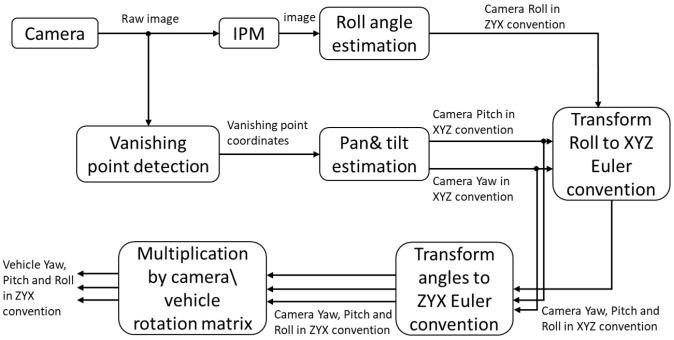


Fig. 1: Steps of the proposed algorithm

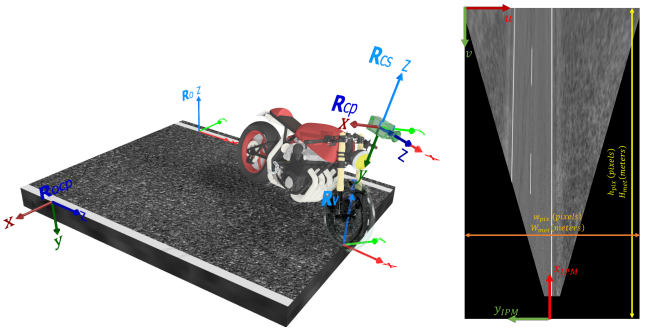
This work was motivated by the need for an efficient and model-independent method based on a monocular camera. The proposed algorithm measures the roll, pitch, and yaw of a P2WV relative to the road, even at high speed without suffering from time drift with low estimation error.

The subsequent section of the paper introduces the system and defines the frames. Section III describes the construction process using IPM techniques with a virtual camera. In section IV, the concept of using pixel width difference to recover the roll angle is explained. The estimation of yaw and pitch angles using the vanishing point technique follows, and all angles are converted to the same Euler convention. This allows for the representation of the camera's rotation matrix and the reconstruction of the vehicle's attitude in terms of roll, pitch, and yaw relative angles. Figure 1 illustrates the steps of the proposed algorithm.

II. SYSTEM DESCRIPTION

Consider the case of a P2WV with a monocular camera installed, traveling on a planar road with two visible lane markers as illustrated in figure 2.a. Let $\mathcal{R}_o$ be the ground reference frame, where its x and y axes are in the road plane. Its z-axis is pointing upwards, and it follows the road lane marker as its x-axis is always tangent to the lane marker. $\mathcal{R}_v$ is the vehicle body frame, defined at the front wheel contact point when there is no steering. $\mathcal{R}_{cs}$ is the camera frame which results from rotating $\mathcal{R}_v$ around it y-axis with a known fixed angle μ. Let $\mathcal{R}_{cp}$ be another camera frame that follows the pinhole camera frame definition, where its origin

[1] All authors are affiliated with IBISC Lab, University of Paris-Saclay, Evry-Courcouronnes, France obaida.alrazouk@univ-evry.fr



(a) Frames definition (b) BEV

Fig. 2: Frames definition and BEV of the road

is the optical point, and its z-axis is the optical axis. Its x and y axes are the same orientation of the image's plane x and y directions, and finally, let $\mathcal{R}_{ocp}$ be a local reference frame where its z-axis is always tangent to the road lane marker, its x-axis is perpendicular to the lane marker to the inside of the road, and its y-axis is pointing downwards. It can be seen easily that the rotation matrix from $\mathcal{R}_{ocp}$ to $\mathcal{R}_o$ and from $\mathcal{R}_{cp}$ to $\mathcal{R}_{cs}$ are given by:

$$ {}^{o}\mathbf{R}_{ocp} = {}^{cs}\mathbf{R}_{cp} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \tag{1} $$

The last two reference frames are defined in this way to facilitate the estimation of the yaw and pitch angles.

In order to apply the algorithm, first, one needs a top view of the road using the camera image from its current pose, known as Bird-Eye-View (BEV), and it is constructed from the original camera image.

III. BEV and Virtual Camera Reconstruction

Reconstructing a BEV from the original image has been a discussion problem for many years. The easiest and most adopted method in the literature is reconstructing the BEV using the Inverse Perspective Mapping (IPM) [5], where a virtual camera is constructed to look down on the road from the same height as the original camera. In order to have a good BEV image, the camera height, pitch, and roll angles are needed as input to the IPM techniques.

However, if provided with a wrong roll angle, the reconstructed lanes will appear distorted, and one lane will appear larger than the other. Ultimately, the distortion in the lane itself enables the estimation of the roll angle. In an ideal Bird's Eye View (BEV) representation, where there is no roll, the lane markers would have equal widths.

While the IPM technique will provide an image, it will not provide the properties of the newly constructed virtual camera. This section aims to calculate the intrinsic parameters of the virtual camera.

When defining the IPM, the region of interest is expressed in meters, taking a width of W_{met} meters centered around the camera and H_{met} meters starting from the camera and ahead. This definition translates to defining a new frame $\mathcal{R}_{IPM}$ that has the exact origin as $\mathcal{R}_{cs}$ but with the orientation of $\mathcal{R}_o$. This will be important for converting word length to pixel length and *vice versa*.

When executing the IPM, the virtual camera frame $\mathcal{R}_{vc}$ will be written as a rotation from $\mathcal{R}_{IPM}$ as ${}^{vc}\mathbf{R}_{IPM} = {}^{o}\mathbf{R}_{ocp}$. Starting with the virtual camera projection equation:

$$ s \cdot \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathbf{K} \times [\mathbf{I}_3|0] \times \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \tag{2} $$

Where u and v are the pixel coordinates. X, Y and Z are the 3D coordinated in $\mathcal{R}_{vc}$ frame, s is the scale which is a number, to say the two sides of the equations are equal to scale. $\mathbf{K} = \begin{bmatrix} f_{vx} & 1 & u_0 \\ 0 & f_{vy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ is the virtual camera intrinsic parameter of the camera, where f_{vx} is the focal length measured of unites of pixel's width, and f_{vy} is the focal length measured of unites of pixel's height, u_0, v_0 are the pixel coordinates of the principle point. However, since only the 3D coordinates in $\mathcal{R}_{IPM}$ frame are known, multiplying equation (2) by the rotation matrix ${}^{vc}\mathbf{R}_{IPM}$ will allow to write X, Y, Z as the 3D coordinated in $\mathcal{R}_{IPM}$.

$$ s \cdot \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathbf{K} \times [{}^{vc}\mathbf{R}_{IPM}|0] \times \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \tag{3} $$

Since the virtual and the actual camera share the same optical point, there is no translation between them.

According to the definition of the region of interest of the IPM, the principle point in the IPM image is at $u_0 = \frac{w_{pix}}{2}, v_0 = h_{pix}$ where w_{pix} and h_{pix} are the width and height of the IPM image in pixels, respectively. By substituting all the mentioned into equation (3):

$$ s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_{vx} & 0 & \frac{w_{pix}}{2} \\ 0 & f_{vy} & h_{pix} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} $$

This leads to the following:

$$ \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \begin{bmatrix} \frac{-Y \cdot f_{vx}}{Z} + \frac{w_{pix}}{2} & \frac{-X \cdot f_{vy}}{Z} + h_{pix} & 1 \end{bmatrix}^T \tag{4} $$

Since Z is the distance between the road plane and the camera origin in $\mathcal{R}_{vc}$, Z can be replaced by $Z = -h$ for all points in the road plane. Also, the IPM region of interest in pixels and meters is known, plugin in the top-right corner with the pixel coordinates of $\begin{bmatrix} w_{pix} & 0 \end{bmatrix}^T$ and the same point coordinates in meters in $\mathcal{R}_{IPM}$ frame are $\begin{bmatrix} H_{met} & \frac{-W_{met}}{2} & -h \end{bmatrix}^T$. Substituting this point in (4) will allow to isolate and calculate both f_{vx} and f_{vy}, hence:

$$ f_{vx} = \frac{h \cdot w_{pix}}{W_{met}}, f_{vy} = \frac{h \cdot h_{pix}}{H_{met}} \tag{5} $$

f_{vx} is needed for estimating the roll angle, as explained in the next section.

IV. Roll Angle Estimation

A. Definition

Take the case of a camera $\mathcal{C}_1$ with no pitch and yaw, just roll angle looking at the road. Let us take a cross-section of the road that passes through the camera's principle point, as illustrated in figure 3 (this section is also the ZX plane

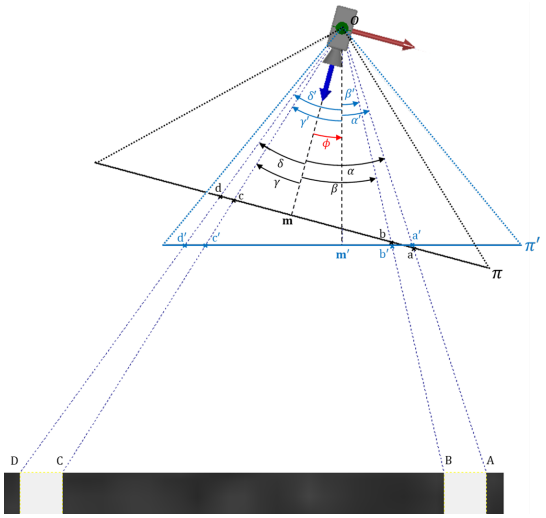


Fig. 3: BEV camera with the virtual no roll camera and the lane points projected on their image planes

in $\mathcal{R}_{cp}$ frame of the camera). Let π be the image plane projection in this section. Let us consider a second camera $\mathcal{C}_2$ -with the exact specification of the previous camera- with no roll angle by rotating the previous camera around its y-axis with a ϕ angle, let π' be its image plane projection of this camera in this cross-section. Let the edges of the white lanes on the surface of the roads be A, B and C, D for each lane marker as illustrated in the figure 3, assuming that these points are visible to the camera, then they have a projection on its image plane. Let a, b, c, d be the projection in pixels in the $\mathcal{C}_1$ camera's image plane, and a', b', c', d' be the projection in pixels in the $\mathcal{C}_2$ camera's image plane. While the pixel position in each image plane differs for the same point between the two cameras $\mathcal{C}_1$ and $\mathcal{C}_2$, the angles between them are still the same since both cameras share the same optical point. Let α be the angle from the optical axis of the $\mathcal{C}_1$ camera $\overrightarrow{Om}$ to $\overrightarrow{Oa}$, β the angle from $\overrightarrow{Om}$ to $\overrightarrow{Ob}$, γ the angle from $\overrightarrow{Om}$ to $\overrightarrow{Oc}$, δ the angle from $\overrightarrow{Om}$ to $\overrightarrow{Od}$. In the same way for the second camera $\mathcal{C}_2$, let α' be the angle from $\overrightarrow{Om'}$ to $\overrightarrow{Oa'}$, β' the angle from $\overrightarrow{Om'}$ to $\overrightarrow{Ob'}$, γ' the angle from $\overrightarrow{Om'}$ to $\overrightarrow{Oc'}$, δ' the angle from $\overrightarrow{Om'}$ to $\overrightarrow{Od'}$. These angles have a positive value if the direction with rotation is positive around the y-axis of $\mathcal{R}_{cp}$ (right-hand rule) and negative in the opposite direction.

B. Concept

In reality, the lane markers have the same width in meters. Also, if the camera has zero roll angle with respect to the road surface, then the lanes will appear with the same width in pixels. Since the original camera has some roll angle, the width of the lane markers will not be equal, one lane will appear wider and the other smaller than they actually are. Taking advantage of the width differences of these distorted lane markers and the knowledge that they are equal in a non-rolled camera, one is able to calculate the roll angle of the camera. The idea is to calculate the width of each lane marker in terms of ϕ in the non-rolled virtual camera $a'b'$ and $c'd'$, using fact that they have an equal length in the virtual camera image plane, we have $\overline{a'b'} = \overline{c'd'}$ that enables to calculate the value of the roll angle ϕ.

Starting by writing the equations for lengths $\overline{a'b'}$ and $\overline{c'd'}$:

$$\overline{a'b'} = f_{vx} \cdot (\tan\alpha' - \tan\beta'), \overline{c'd'} = f_{vx} \cdot (\tan\gamma' - \tan\delta')$$

Where f_{vx} is the focal length measured in pixel width. For the equation of $\overline{c'd'}$, the order of angles is reversed because points c, d lie on the opposite side of a, b as seen from the origin m. Thus, they have negative pixel values. After slight calculation, this leads to the following:

$$\tan\alpha' - \tan\beta' - \tan\gamma' + \tan\delta' = 0 \tag{6}$$

From the definition and as illustrated by figure (3) one can write $\alpha', \beta', \gamma', \delta'$ in terms of ϕ and $\alpha, \beta, \gamma, \delta$ as follows:

$$\alpha' = \alpha - \phi, \beta' = \beta - \phi, \gamma' = \gamma - \phi, \delta' = \delta - \phi \tag{7}$$

Substituting the last in (6)

$$\tan(\alpha - \phi) - \tan(\beta - \phi) - \tan(\gamma - \phi) + \tan(\delta - \phi) = 0 \tag{8}$$

By using simple triangulation, angles $\alpha, \beta, \gamma, \delta$ can be measured from the $\mathcal{C}_1$ camera with:

$$\begin{aligned} \alpha = atan2(a', f_{vx}), \beta = atan2(b', f_{vx}) \\ \gamma = atan2(c', f_{vx}), \delta = atan2(d', f_{vx}) \end{aligned} \tag{9}$$

Noting that the pixel coordinates can be positive or negative.

In the special case where a center lane is visible, it can greatly enhance the estimation since the distance between the center lane and the edge lanes are significantly larger. Doing the same analysis by replacing lane width with the distance between the edge lanes and the center lane, one can get:

$$\tan(\alpha_1 - \phi) - 2\tan(\alpha_2 - \phi) + \tan(\alpha_3 - \phi) = 0 \tag{10}$$

Where $\alpha_1, \alpha_2, \alpha_3$ represent the pixel coordinates of the left, center, and right lanes, respectively.

Equations (10) and (8) will enable the calculation of the value of ϕ directly. However, the solution must be done numerically as it can not be solved analytically, which introduces some computational costs.

C. Polynomial approximation

While the solution of equation (8) can be found easily numerically, we can speed up the calculation even further by using Padé approximation of the $\tan$, which can transform the problem to calculate a root of a polynomial.

The Padé approximation of $\tan x$ when $-\frac{\pi}{2} < x < \frac{\pi}{2}$ is given by the formula: $pade(x) = \frac{N(x)}{D(x)} = \frac{x - \frac{1}{9}x^3 + \frac{1}{945}x^5}{1 - \frac{4}{9}x^2 + \frac{1}{63}x^4}$

Under the assumptions that :

$-\frac{\pi}{2} < \alpha - \phi, \beta - \phi, \gamma - \phi, \delta - \phi < \frac{\pi}{2}$

These assumptions always hold because the camera field of view is less than π. Applying the Padé approximation on equation (8), one can get :

$$\begin{aligned} N(\alpha-\phi)\cdot D(\beta-\phi)\cdot D(\gamma-\phi)\cdot D(\delta-\phi) \\ -D(\alpha-\phi)\cdot N(\beta-\phi)\cdot D(\gamma-\phi)\cdot D(\delta-\phi) \\ -D(\alpha-\phi)\cdot D(\beta-\phi)\cdot N(\gamma-\phi)\cdot D(\delta-\phi) \\ +D(\alpha-\phi)\cdot D(\beta-\phi)\cdot D(\gamma-\phi)\cdot N(\delta-\phi) = 0 \end{aligned}$$

Where N and D are the nominator and denominator of the Padé function previously defined, this reduces the problem of finding the real root closest to zero of this 17^{th} degree polynomial. While this is still not solvable analytically, it is lighter computationally. We can further enhance the performance by choosing the initial roll as the last estimated one because we know that the roll change in two successive frames is small. This method will provide the camera roll angle of frame $\mathcal{R}_{cs}$ in the ZYX Euler convention.

D. Error analysis

Let us define the width difference between the two lane markers in pixels as a function f:

$$f = \tan\left(\arctan\left(\frac{a'}{f_{vx}}\right) - \phi\right) - \tan\left(\arctan\left(\frac{b'}{f_{vx}}\right) - \phi\right) - \tan\left(\arctan\left(\frac{c'}{f_{vx}}\right) - \phi\right) + \tan\left(\arctan\left(\frac{d'}{f_{vx}}\right) - \phi\right)$$

Let us define the function g as:

$$g(p, \phi) = \tan\left(\arctan\left(\frac{p}{f_{vx}}\right) - \phi\right)$$

From this definition, it can be concluded from the error propagation rule that: $\delta g = |\frac{\partial g}{\partial p}|\delta p + |\frac{\partial g}{\partial \phi}|\delta\phi$, then:

$$\delta g = \frac{\frac{f_{vx}}{p^2+f_{vx}^2}\delta p}{\cos^2\left(\arctan\left(\frac{p}{f_{vx}}\right) - \phi\right)} + \frac{\delta\phi}{\cos^2\left(\arctan\left(\frac{p}{f_{vx}}\right) - \phi\right)}$$

To make the writing easier, let us define a function q as $q(p) = \frac{1}{\cos^2\left(\arctan\left(\frac{p}{f_{vx}}\right) - \phi\right)}$, that means $\delta g = \frac{f_{vx}}{p^2+f_{vx}^2} q(p) \cdot \delta p + q(p) \cdot \delta\phi$, then f can be expressed as:

$$f = g(a', \phi) - g(b', \phi) - g(c', \phi) + g(d', \phi) \tag{11}$$

Assuming the error in pixel position is the same for all the points a', b', c', d', one can write $\delta p = \delta a', \delta b', \delta c', \delta d'$. Following the error propagation rule, one can write:

$$\delta f = \left(\frac{f_{vx}}{a'^2 + f_{vx}^2}.q(a') + \frac{f_{vx}}{b'^2 + f_{vx}^2}.q(b') + \frac{f_{vx}}{c'^2 + f_{vx}^2}.q(c') + \frac{f_{vx}}{d'^2 + f_{vx}^2}.q(d')\right).\delta p + \left(q(a') + q(b') + q(c') + q(d')\right).\delta\phi$$

Which allows us to write:

$$\delta\phi = \frac{1}{q(a') + q(b') + q(c') + q(d')}.\left(\delta f + \left(\frac{f_{vx}}{a'^2 + f_{vx}^2}.q(a') + \frac{f_{vx}}{b'^2 + f_{vx}^2}.q(b') + \frac{f_{vx}}{c'^2 + f_{vx}^2}.q(c') + \frac{f_{vx}}{d'^2 + f_{vx}^2}.q(d')\right).\delta p\right)$$

There are two terms that contribute to the roll angle estimation error. The first one is due to the difference in width between the lane markers in reality. The second is due to the pixel position error in selecting the lane edges.

However, it can be proven that $\delta\phi$ has an upper limit. Let us take each term individually, starting from the second term. it is known that $\frac{f_{vx}}{a'^2+f_{vx}^2}, \leq \frac{1}{f_{vx}}$ Applying this for the rest of the points b', c', d' leads to:

$$\frac{\frac{f_{vx}}{a'^2+f_{vx}^2}q(a') + \frac{f_{vx}}{b'^2+f_{vx}^2}q(b') + \frac{f_{vx}}{c'^2+f_{vx}^2}q(c') + \frac{f_{vx}}{d'^2+f_{vx}^2}q(d')}{q(a') + q(b') + q(c') + q(d')} \leq \frac{1}{f_{vx}} = \frac{W_{met}}{h \cdot w_{pix}}$$

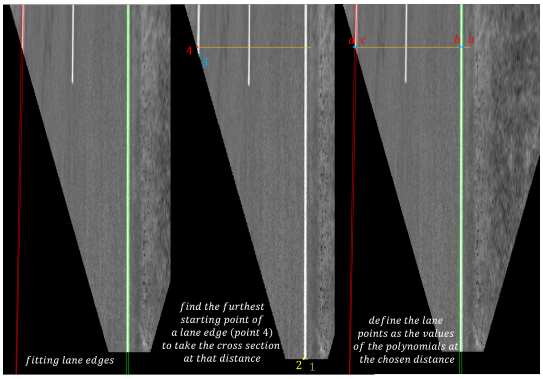


Fig. 4: Selecting the cross-section and lane points

For the first term, from the definition $q(p) = \frac{1}{\cos^2\left(\arctan\left(\frac{p}{f_{vx}}\right) - \phi\right)} \geq 1$, then:

$$\frac{1}{q(a') + q(b') + q(c') + q(d')} \leq \frac{1}{4} \tag{12}$$

Substituting these inequalities in $\delta\phi$ equation leads to:

$$\delta\phi \leq \frac{1}{4}.\delta f + \frac{W_{met}}{h \cdot w_{pix}}.\delta p \tag{13}$$

However, δf is taken in pixels. In the real world, one is interested in the width difference between the lane markers in meters, returning to figure 3 and from trigonometry, it is easy to find that $\delta f = \frac{\delta w.f_{vx}}{h} = \frac{\delta w.w_{pix}}{W_{met}}$ where δw is the error of the lane width difference given in meters. Thus, giving the final error equation for the roll angle

$$\delta\phi \leq \frac{w_{pix}}{4W_{met}}.\delta w + \frac{W_{met}}{h \cdot w_{pix}}.\delta p \tag{14}$$

E. Picking lane points in practice

The method for calculating the roll angle relies essentially on the points a, b, c and d. Their extraction from the image is critical for the estimation process. We provide a method for estimating one valid set of these points with sub-pixel coordinated. Sub-pixel coordinates are essential as the width of the lanes may differ by less than one pixel.

After getting the distorted IPM image, it is binarized and filtered, then the two lane markers are detected, and only the edges of each lane marker are kept. After that, the edges are filtered and fitted to a second-degree polynomial (this fitting is what provided the sub-pixel coordinates), then the cross-section is chosen at a distance as the furthest of the starting point of each lane marker edge (figure 4), and then assign the points as the values of the corresponding polynomial.

V. Estimating angles from vanishing point

Vanishing point techniques have been used in the literature for pose estimation [8], [6]. However, from a single vanishing point, estimating "pan" and "tilt" angles was not very useful since they are not estimated in the standard ZYX Euler convention. However, if the roll angle is available, it is possible to transform all the angles into the same convention.

For a pinhole camera, we know that the formula of perspective projection in homogeneous coordinates is:

$$s \cdot \begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathbf{K} \times [{}^{cp}\mathbf{R}_{ocp} | \mathbf{t}] \times \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \tag{15}$$

Where u, v are the pixel coordinates, and X, Y, Z are the 3D coordinates written in $\mathcal{R}_{cp}$, $\mathbf{K}$ is the 3x3 calibration matrix, ${}^{cp}\mathbf{R}_{ocp}$ is the 3x3 rotation matrix from the camera frame to the world frame, $\mathbf{t}$ is the translation vector from the camera frame to the world frame.

In this case, the interest is in the vanishing point where the two road lanes intersect, which is in the z direction in the world frame $\mathcal{R}_{ocp}$. This vanishing point is expressed in the homogeneous coordinate system as $\mathbf{V}_z = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$ The zero at the last component indicates that the point lies at infinity. Assuming that v_{pix} is the coordinates in pixel of the vanishing point in the image plane, one gets:

$$s \cdot \mathbf{v}_{pix} = \mathbf{K} \times [\mathbf{R} | \mathbf{t}] \times \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{16}$$

Which leads to $s \cdot \mathbf{v}_{pix} = \mathbf{K} \times r_3$ Where r_3 is the 3^{rd} column of the rotation matrix $\mathbf{R}$, knowing that r_3 should have a unit length, that will get rid of the scaling factor s, and one can write

$$r_3 = \frac{\mathbf{K}^{-1} \times \mathbf{v}_{pix}}{\left\| \mathbf{K}^{-1} \times \mathbf{v}_{pix} \right\|} \tag{17}$$

Let α be the pan angle between $\mathcal{R}_{cp}$ and $\mathcal{R}_{ocp}$ around y-axis of $\mathcal{R}_{cp}$, then let β be the tilt angle between $\mathcal{R}_{cp}$ and $\mathcal{R}_{ocp}$ around x-axis of the resulting frame, following the standard YXZ Euler angles order convention.

Because of the way the camera is defined, it is only possible to get the last row of the rotation matrix ${}^{cp}\mathbf{R}_{ocp}$ from the vanishing point, in order to estimate pan and tilt, a representation that enables to calculate the two angles just from the last column has to be chosen. Thus the use of the YXZ Euler representation, where the rotation matrix takes the following form:

$${}^{cp}R_{ocp} = \begin{bmatrix} c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma & c_\gamma s_\alpha s_\beta - c_\alpha s_\gamma & s_\alpha c_\beta \\ c_\beta s_\gamma & c_\beta c_\gamma & -s_\beta \\ -c_\gamma s_\alpha + c_\alpha s_\beta s_\gamma & c_\alpha c_\gamma s_\beta + s_\alpha s_\gamma & c_\alpha c_\beta \end{bmatrix}$$

Where s_θ is $\sin\theta$, and c_θ is $\cos\theta$.

It can be seen that the last column is just composed of the pan and tilt angles. From this, the connection between the vanishing point and the pan and tilt angles can be concluded: $r_3 = \begin{bmatrix} \sin\alpha \cdot \cos\beta & -\sin\beta & \cos\alpha \cdot \cos\beta \end{bmatrix}^T$ which means that the equations of both α and β can be found as follows: $\alpha = atan2(r_3(1), r_3(3)), \beta = -asin(r_3(2))$ where $r_3(1), r_3(2), r_3(3)$ are the components of r_3.

A. Camera angles from YXZ to XYZ convention

The previous section explained how to get the pan and tilt angle of the camera. However, they are written in YXZ representation, which is not helpful. In this section, we prove that α is exactly the yaw angle of the frame $\mathcal{R}_{cs}$ from $\mathcal{R}_o$, and β is precisely the pitch angle of the frame $\mathcal{R}_{cs}$ from $\mathcal{R}_o$ following an XYZ Euler angles representation.

What is required is to get the angles of $\mathcal{R}_{cs}$ from frame $\mathcal{R}_o$ first using XYZ Euler convention to have all angles in the same representation, and they can be transformed easily to ZYX Euler representation which is the most used in vehicles. In order to prove that α and β are the yaw and pitch angles in XYZ representation, we need to write ${}^{o}\mathbf{R}_{cs}$ in terms of α and β and compare it with the rotation matrix in the XYZ convention. Starting from ${}^{cp}\mathbf{R}_{ocp}$:

$${}^{o}\mathbf{R}_{cs} = {}^{o}\mathbf{R}_{ocp} \times {}^{ocp}\mathbf{R}_{cp} \times {}^{cp}\mathbf{R}_{cs} \tag{18}$$

From the definition, ${}^{o}\mathbf{R}_{ocp} = {}^{cp}\mathbf{R}_{cs}^T$ Moreover, it is known that ${}^{ocp}\mathbf{R}_{cp} = {}^{cp}\mathbf{R}_{ocp}^T$ Substituting these in (18) one get:

$${}^{o}\mathbf{R}_{cs} = {}^{cp}\mathbf{R}_{cs}^T \times {}^{cp}\mathbf{R}_{ocp}^T \times {}^{cp}\mathbf{R}_{cs} \tag{19}$$

Furthermore, one can write:

$${}^{cp}\mathbf{R}_{ocp} = \begin{bmatrix} c_\alpha c_\beta & -s_\alpha c_\beta & s_\beta \\ c_\gamma s_\alpha - c_\alpha s_\beta s_\gamma & c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma & c_\beta s_\gamma \\ -c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\beta c_\gamma \end{bmatrix}$$

On the other hand, we have the rotation matrix constructed from the XYZ Euler convention: ${}^{o}\mathbf{R}_{cs} = \mathbf{R}_x(-\phi) \times \mathbf{R}_y(\theta) \times \mathbf{R}_z(\psi)$ By doing the matrices product, one gets the following

$${}^{cp}\mathbf{R}_{ocp} = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\theta & s_\theta \\ c_\phi s_\psi - c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\theta s_\phi \\ -c_\psi s_\theta c_\phi - s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix}$$

Since $-\frac{\pi}{2} < \alpha, \beta, \gamma < \frac{\pi}{2}$ the rotation is unique, that means there is only one set of (α, β, γ) that constructs this rotation, thus $\psi = \alpha, \theta = \beta, \phi = \gamma$. This proves that α is indeed the yaw angle and β is the pitch angle of the camera in the XYZ Euler convention.

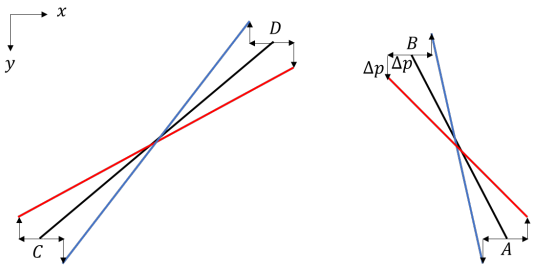


Fig. 5: lines that define the vanishing point with their error

B. Error analysis

Since the yaw and pitch estimation relies on the vanishing point, the error analysis of the vanishing point position is necessary. Let each line be comprised of two points, A, B for the right, and C, D for the left. Let δp be the maximum projection error for the four points. Thus, the worst case is for each point to have $\mp\delta p$ on each coordinate. According to this definition, the error in the vanishing point coordinates can be bounded by the maximum and minimum values for each of the x and y coordinates of the intersection. Note that the $y - axis$ is pointing downwards.

The maximum x coordinates for the intersection point is to take the right line with the largest slope and the left line with the smallest, as illustrated in figure 5 by the blue right line with the red left line giving the maximum x. for the minimum x it is the inverse, the smallest slope right line and the largest slope left line (right red with left blue lines on figure 5).

The largest x is the intersection of these two lines:

$$y = y_A + \delta p + \frac{(-y_A + y_B - 2\delta p)}{(-x_A + x_B + 2\delta p)}(x - x_A + \delta p)$$
$$y = y_C - \delta p + \frac{(-y_C + y_D + 2\delta p)}{(-x_C + x_D + 2\delta p)}(x - x_C + \delta p)$$

Which gives:

$$x_{max} = \frac{a_0 + a_1\delta p + a_2\delta p^2}{a_3 + a_4\delta p + a_5\delta p^2}$$

Where $a_0, ..., a_5$ are comprised only of $x_A, x_B, x_C, x_D, y_A, y_B, y_C, y_D$
The smallest x can be given in the same way as:

$$x_{min} = \frac{b_0 + b_1\delta p + b_2\delta p^2 + b_3\delta p^3}{b_4 + b_5\delta p + b_6\delta p^2}$$

Where $b_0, ..., b_6$ are comprised only of $x_A, x_B, x_C, x_D, y_A, y_B, y_C, y_D$ knowing this, one can define $\delta x = |x_{max} - x_{min}|$.

For the y coordinates, in the same way, the maximum y is the intersection of the two smallest slope lines from the left and right (the two red lines in figure 5). Moreover, the smallest y is taken by intersecting the two largest slope lines from the left and right (the two blue lines in figure 5).

$y_{min} = \frac{c_0 + c_1\delta p + c_2\delta p^2 + c_3\delta p^3}{c_4 + c_5\delta p + c_6\delta p^2}$
$y_{max} = \frac{d_0 + d_1\delta p + d_2\delta p^2 + d_3\delta p^3}{d_4 + d_5\delta p + d_6\delta p^2}$
Where $d_0, ..., d_6, c_0, ..., c_6$ are composed only of $x_A, x_B, x_C, x_D, y_A, y_B, y_C, y_D$.
One can define $\delta y = |y_{max} - y_{min}|$
Now for the error in yaw and pitch

$$r_3 = \begin{bmatrix} \sin\psi \cdot \cos\theta \\ -\sin\theta \\ \cos\psi \cdot \cos\theta \end{bmatrix} = \frac{\mathbf{K}^{-1} \times \mathbf{v}_{pix}}{\left\|\mathbf{K}^{-1} \times \mathbf{v}_{pix}\right\|} \tag{20}$$

It is known that $\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{fx} & 0 & -\frac{u0}{fx} \\ 0 & \frac{1}{fy} & -\frac{v0}{fy} \\ 0 & 0 & 1 \end{bmatrix}$ thus

$$r_3 = \frac{1}{\sqrt{1 + \frac{(u0-x)^2}{fx^2} + \frac{(v0-y)^2}{fy^2}}} \begin{pmatrix} \frac{-u0+x}{fx} & \frac{-v0+y}{fy} & 1 \end{pmatrix}$$

$$\psi = \arctan\left(\frac{r_3(1)}{r_3(3)}\right) = \arctan\left(\frac{-u0 + x}{fx}\right)$$

$$\theta = \arcsin\left(r_3(2)\right) = \arcsin\left(\frac{-v0 + y}{fy\sqrt{1 + \frac{(u0-x)^2}{fx^2} + \frac{(v0-y)^2}{fy^2}}}\right)$$

For the yaw angle, by applying the error propagation rule, one can get $\delta\psi = \frac{f_x}{{f_x}^2 + (u_0 - x)^2}\delta x \leq \frac{1}{f_x}\delta x$ thus the yaw error is only related to the error of the x position of the vanishing point.

By applying the error propagation rule for the pitch angle:

$$\delta\theta = h_1(u_0, v_0, x, y, f_x, f_y)\delta x + h_2(u_0, v_0, x, y, f_x, f_y)\delta y$$

$$h_1 = \frac{(u_0 - x)(-v_0 + y)}{f_x^2 f_y (1 + \frac{(u_0-x)^2}{f_x^2} + \frac{(v_0-y)^2}{f_y^2})^{\frac{3}{2}}} \cdot \frac{1}{\sqrt{1 - \frac{f_x^2(v0-y)^2}{f_y^2(u_0-x)^2 + f_x^2(f_y^2 + (v_0-y)^2)}}}$$

$$h_2 = \frac{1}{(f_y^2(u_0-x)^2 + f_x^2(f_y^2 + (v_0-y)^2))} \cdot \frac{1}{\sqrt{1 + \frac{(u_0-x)^2}{f_x^2} + \frac{(v_0-y)^2}{f_y^2}}} \cdot \frac{f_y(f_x^2 + (u_0-x)^2)}{\sqrt{1 - \frac{f_x^2(v_0-y)^2}{f_y^2(u_0-x)^2 + f_x^2(f_y^2 + (v_0-y)^2)}}}$$

VI. Roll angle to XYZ and construction of camera rotation matrix

The roll angle is in the desired ZYX Euler convention. However, because the pitch and yaw are in XYZ convention, it is easier to convert the roll to XYZ convention, then construct the rotation matrix, and then convert everything back to the desired ZYX convention.

The rotation matrix is the same no matter what convention we choose. Its rows and columns do not change with the representation. According to that, we can write:

$\mathbf{R}_x(\phi_1) \times \mathbf{R}_y(\theta_1) \times \mathbf{R}_z(\psi_1) = \mathbf{R}_z(\psi_2) \times \mathbf{R}_y(\theta_2) \times \mathbf{R}_x(\phi_2)$

It can be found by isolating c_{θ_2} from $c_{\psi_1}c_{\theta_1} = c_{\psi_2}c_{\theta_2}$ that $c_{\theta_2} = \frac{c_{\psi_1}c_{\theta_1}}{c_{\psi_2}}$, by substituting the last in $s_{\psi_1}s_{\theta_1}c_{\phi_1} + c_{\psi_1}s_{\phi_1} = c_{\theta_2}s_{\phi_2}$, and by isolating ϕ_1, it can be shown that:

$$\phi_1 = atan2(\cos\theta_1 \cdot \tan\phi_2 - \sin\psi_1 \cdot \sin\theta_1, \cos\psi_1) \tag{21}$$

Which transforms the roll angle from the ZYX convention to the XYZ convention using the yaw and pitch from the XYZ convention.

Now that all the roll, pitch, and yaw angles are known in the same convention, the transformation matrix of the camera frame $\mathcal{R}_{cs}$ with respect to the ground frame $\mathcal{R}_o$ can be written as follows:

$\mathbf{R} = {}^o\mathbf{R}_{cs} = \mathbf{R}_x(\phi_1) \times \mathbf{R}_y(\theta_1) \times \mathbf{R}_z(\psi_1)$

VII. From camera to vehicle

The goal is to recover the vehicle's rotation matrix or rotation angles, not the camera's. However, knowing the camera rotation matrix is the hardest part. Then it is multiplied by the rotation matrix between the vehicle and the camera to get to the vehicle rotation matrix.

From the definition ${}^v\mathbf{R}_{cs} = \begin{bmatrix} \cos(\mu) & 0 & \sin(\mu) \\ 0 & 1 & 0 \\ -\sin(\mu) & 0 & \cos(\mu) \end{bmatrix}$

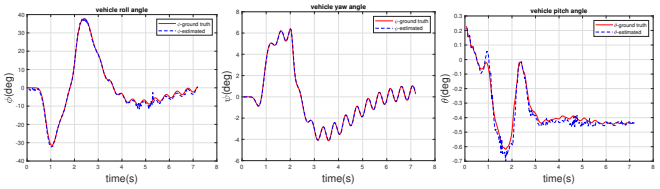
Then:

$${}^o\mathbf{R}_v = {}^o\mathbf{R}_{cs} \times {}^v\mathbf{R}_{cs}^T \tag{22}$$

Assuming that $\mathbf{R} = {}^o\mathbf{R}_v = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ one can extract the yaw, pitch, and roll using the ZYX convention for the vehicle as follows:

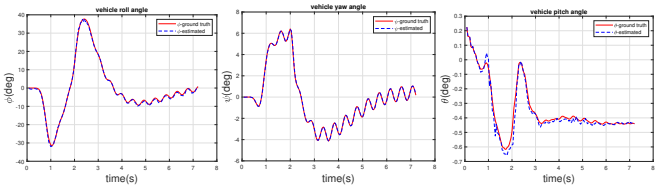
$\psi = atan2(r_{21}, r_{11}), \theta = -asin(r_{31}), \phi = atan2(r_{32}, r_{33})$

In practice, an initial calibration process is crucial to get ${}^v\mathbf{R}_{cs}$ and $\mathbf{K}$, in addition to setting the nominal camera height. This calibration process is done only once when installing the system.



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 6: DLC on a straight road at $150km/h$ in HD



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 7: DLC on a straight road at $150km/h$ in FHD

VIII. Simulation and Results Analysis

Three different scenarios were created using BikeSim to asses the proposed approach with different road shapes, speeds, duration, and image resolutions. Each is key to evaluating one aspect of the algorithm.

It is essential to state that the roads are empty with a well-lit environment, with no strange objects in the scene. The camera was installed at the height of $h = 1.1m$ and tilted $\mu = 15°$ in the virtual environment. The only input to the algorithm is the image. All these tests were done on a Matlab R2020b environment, installed on Windows 10, on a mid-range laptop with an AMD Ryzen 3750H chip.

A. Double Lane Change in a short straight road

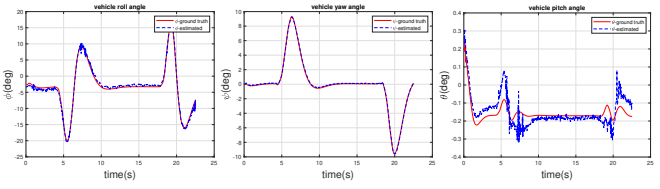
A Double Lane Change (DLC) maneuver was executed at a high speed of $150km/h$. The test was done using two different camera resolutions, one at Full-HD (1920×1080) at $24FPS$, and the other at HD resolution (1280×720) at $30FPS$. The camera Field of View was $80°$ for both resolutions.

The estimation is excellent with very low Root Mean Squared Errors (RMSE), as shown in table I and figures 6 and 7.

B. Double Lane Change in a short curved road

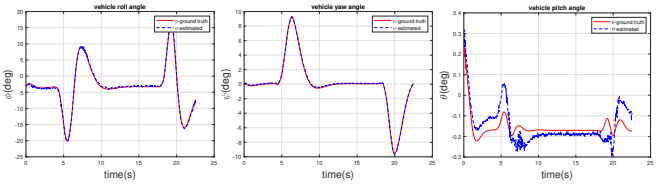
This test is critical, as it evaluates the algorithm on a more challenging curved road shape while executing a DLC maneuver. This test was executed at a speed of $80km/h$. The test was done using similar camera resolutions to the first test, FHD at $24FPS$ and HD at $30FPS$. The camera Field of view was $120°$ for both resolutions. This is very important as seeing the curve as close as possible to the camera helps with the vanishing point estimation, thus helping with the yaw and pitch estimation.

The estimation was also very good with very low RMSE, as shown in table I.Figures 8 and 9 illustrate these results.



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 8: DLC on a curved road at $100km/h$ in HD



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 9: DLC on a curved road at $100km/h$ in FHD

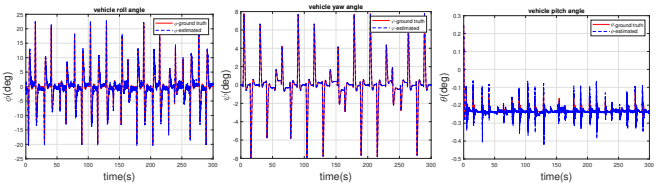
C. Very long straight road with repeatable DLC maneuvers

As this algorithm estimates the absolute vehicle's angles (relative to ground), it is essential to show that there is no drift over time, because the estimation is not accumulative in the algorithm as in other odometry algorithms. There should be no drift over time. This test is critical, as it shows the added value of the algorithm. This test was executed at a high speed of $100km/h$. The test was done using similar camera resolutions to the first test, FHD at $24FPS$ and HD at $30FPS$. The camera Field of view was $80°$ for both resolutions.

The estimation was also very good with very low RMSE as shown in table I.Figures 10 and 11 illustrate these results.

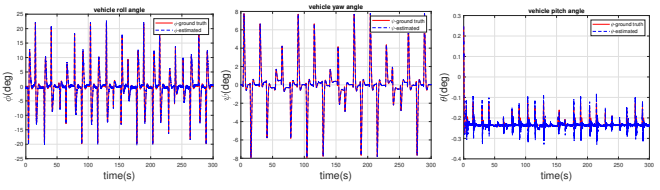
D. Special case when the center lane is available

As mentioned in section IV.B, the estimation of the roll angle can be enhanced by the inclusion of the center lane. This significantly improves the accuracy of the roll angle



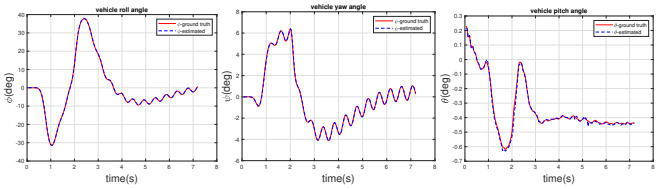
(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 10: DLC on a straight road at $80km/h$ in HD



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 11: DLC on a straight road at $80km/h$ in FHD



(a) Roll angle (b) Yaw angle (c) Pitch angle

Fig. 12: DLC, straight, $150km/h$, HD, centerlane visible

Scenario	Roll(°)	Pitch(°)	Yaw(°)
1 at $30FPS$ HD with center line	0.1905	0.0563	0.0124
2 at $30FPS$ HD with center line	0.194	0.0897	0.0532
3 at $30FPS$ HD with center line	0.2189	0.0522	0.005
1 at $24FPS$ FHD	0.5159	0.0557	0.03
2 at $24FPS$ FHD	0.314	0.0949	0.0591
3 at $24FPS$ FHD	0.4324	0.0517	0.0107
1 at $30FPS$ HD	0.9361	0.0555	0.0347
2 at $30FPS$ HD	0.6114	0.0893	0.0568
3 at $30FPS$ HD	0.7928	0.0515	0.0173

TABLE I: RMSE of angles estimation in different scenarios

just at HD $30FPS$. The results are far better than just the two lane markers with a stable RMSE of around $0.2°$ across all tests achieving the lowest error and best fit as illustrated in table I and figure 12.

IX. Discussion

As shown in the results, the approximation is excellent. While the error for the yaw and pitch remains relatively the same across all tests, the roll angle error benefits significantly from higher resolution and higher frame rate. Increasing the resolution from HD to FHD doubled the accuracy. In the special case where the center lane is present in the scene, the accuracy was improved four to five times in some scenarios, and it was stable on about $0.2°$ across all scenarios.

In figure 10, it can be seen that the estimation of the pitch is not perfect. However, in this scenario, the road was curved, which produces some estimation error, plus the error from the roll angle is propagated when constructing the camera rotation matrix. Nevertheless, the scale of the pitch in that scenario is very small, and the RMSE value of the error is still similar to other scenarios.

While the pitch and the height of the camera are important inputs to the IPM technique, they are not provided as inputs to the algorithm. They are entered as fixed values inside the algorithm with their initial values $\mu = 15°$ for the pitch angle and $h = 1.1m$ for the height as the roll estimation is tolerant against pitch change since we are only taking points on the road on the same cross-section. This means the same distance, and the change in pitch has the same effect on all points in the same cross-section at the same distance from the camera. Thus, the pitch effects will cancel out for these points. It is also robust against height change since it will only affect the scaling of the image, which will not affect the results.

X. Conclusions

In this paper, a real-time novel technique is proposed to estimate the relative roll, pitch, and yaw angles of the vehicle to the road using only a monocular camera fixed on the vehicle body. The technique relies on a novel approach to estimate the roll angle from the two lane-markers distortion in the BEV image. The roll and pitch angles were estimated using the vanishing point technique. Then all the angles were transformed to be written in the same Euler angle convention. This allowed the writing of the camera rotation matrix and the extraction of the vehicle rotation angles. The algorithm was tested using several scenarios with different road shapes and speeds. The results were indeed very promising. The advantage of this algorithm is that it is independent of the vehicle's dynamic model, and it can be applied to other road vehicles. In addition, the estimation is absolute to the road, as there is no drift over time. Finally, it tolerates camera height and pitch changes, in addition to the real-time performance. However, the method is currently hard to use in traffic areas because it requires lane markers to be visible on the scene.

References

[1] "Traffic Safety Facts : Motorcycles". Crashstats. National Highway Traffic Safety Administration.

[2] D. Maceira, A. Luaces, U. Lugrís, M. Á. Naya, and E. Sanjurjo, "Roll Angle Estimation of a Motorcycle through Inertial Measurements," Sensors, vol. 21, no. 19. MDPI AG, p. 6626, Oct. 05, 2021.

[3] H. Slimi, H. Arioui, and S. Mammar, "Motorcycle lateral dynamic estimation and lateral tire-road forces reconstruction using sliding mode observer," 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, Oct. 2013.

[4] P.-M. Damon, H. Dabladji, D. Ichalal, L. Nehaoua, and H. Arioui, "Estimation of lateral motorcycle dynamics and rider action with Luenberger observer," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, Nov. 2016.

[5] P.-M. Damon, H. Hadj-Abdelkader, H. Arioui, and K. Youcef-Toumi, "Inverse Perspective Mapping Roll Angle Estimation for Motorcycles," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, Nov. 2018.

[6] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast Vanishing-Point Detection in Unstructured Environments," IEEE Transactions on Image Processing, vol. 21, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 425–430, Jan. 2012.

[7] R. Labayrade and D. Aubert, "A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision," IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683). IEEE.

[8] V. Babaee-Kashany and H. R. Pourreza, "Camera Pan and Tilt Estimation in Soccer Scenes Based on Vanishing Points," 2010 Fourth UKSim European Symposium on Computer Modeling and Simulation. IEEE, Nov. 2010.

[9] Y. Yang, Q. Shen, J. Li, Z. Deng, H. Wang, and X. Gao, "Position and Attitude Estimation Method Integrating Visual Odometer and GPS," Sensors, vol. 20, no. 7. MDPI AG, p. 2121, Apr. 09, 2020.