*Proceedings of the International Conference on Machine Vision and Machine Learning*
*Prague, Czech Republic, August 14-15, 2014*
*Paper No. 127*

# Monocular Vision Based Autonomous Navigation for Arbitrarily Shaped Urban Roads

**Vinit Gela, M S Suraj, Kshitij Patil, Bijil Prakash**
Birla Institute of Technology and Sciences, K.K.Birla Goa campus,
Goa, India.
f2011016@goa.bits-pilani.ac.in; f2011033@goa.bits-pilani.ac.in; f2012024@goa.bits-pilani.ac.in;
bijil@goa.bits-pilani.ac.in

***Abstract*** **-** In this paper we present a method for urban road navigation for Autonomous Land Vehicle (ALV). The method does not make any assumptions pertaining to the shape of the road. Earlier work based on monocular vision navigation involved estimating the road centre by finding the vanishing point. However such methods cannot be used for arbitrarily shaped roads and impose a constraint on the shape of the road region. Our contribution is finding a method that works for such arbitrarily shaped, nonhomogeneous roads. First we segment out the road region and then find a reference point on the segmented binary image. In section 3 we describe an algorithm to find a suitable reference point, an alternative to vanishing point for arbitrarily shaped roads. We tested our navigation algorithm on the simulation software, Webots.

***Keywords*****:** Arbitrarily shaped, Monocular vision, Vanishing point, Webots.

## 1. Introduction

Monocular vision based navigation is an explored topic of research. The methods used previously were based on the estimation of the road centre by finding the vanishing point (Siagian et al., 2013), (Chang et al., 2012), (Kong et al., 2010). However, such methods assume that the road region is triangular or calculate the vanishing point based on contour segments using a group of line. We propose a method that is independent of the shape of the road. Frew et al.(2004), worked on vision based road following using a small aircraft. Their approach was to segment the road and then follow the central part of the road using PID. Our approach is similar in construction. However in their implementation, the camera mounted on the plane was at a considerable height from the ground and was facing downwards, so finding the road centre was not a tough task. In our setup, due to limited span of the centrally mounted camera on the ground robot, it is difficult to estimate the true road centre. We assume that the camera is centrally mounted on the robot, facing towards the instantaneous direction of the motion. Our approach is comprised of two parts. The first part is segmenting out the road region. The second part is finding a reference point on the segmented image. The reference point is chosen such that when the robot tries to align itself with respect to the point, in the course of its motion, it always stays on the road. The input to the second part of the algorithm is a binary image which is comprised of road and non-road pixels (black region indicates the road pixels and white region indicates the background). Colour histogram based road model (Tan et al., 2006), illuminant-invariant road model (Alvarez et al., 2008) are two road segmentation methods that do not constrain the shape of the road.Both of these methods work for arbitrarily shaped, nonhomogeneous urban roads. However, both fail when the robot starts near the road boundary, where the bottom part of the image consists of both the road and the background. This is one of the assumptions for our implementation as well. To find the reference point, we consider a line $l$:V=$c$, on the binary image $I$, according to the U-V image coordinate system (Fig. 2.b), where c is a constant. We analyse the different possible cases which occur depending upon the intersection of the line $l$ and the segmented road region (Fig.1). The midpoint of the intersection points of the line $y$=$c$ and the road region, on the binary image, is the calculated reference point. Under certain conditions and a suitable constant c, a reference point can always be found. The next section describes the algorithm and its motivation.

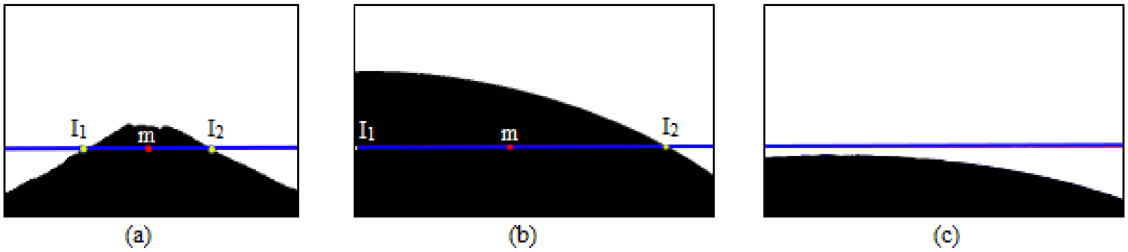

Fig. 1. The black region of the image indicates the road. In each of the figures, the line *l:* V = *c* is represented by a blue line.

## 2. Theory

As discussed earlier, the first part of our implementation is road segmentation. The output of the road segmentation is a binary image, comprised of road and non-road pixels. We implement the illuminant-invariant road segmentation algorithm. We prefer this method over the colour histogram based method, as the former does not require generating a background model to quantify pixels. Also the latter method is much more complex due to updating of the road model set and scanning through all the models to determine multiple road probabilities. The segmented image *I* is used as an input to the second part of the algorithm. To understand the implementation of the second part of the algorithm, we consider the pinhole camera model. According to pinhole camera model, if P(X, Y, Z) is a point with respect to the camera coordinate system, and p(u,v), is its representative point on the image plane (assuming P is visible) where u and v are the image coordinates with respect to the image coordinate system (both coordinate systems shown in Fig. 2.b), then X, Y, Z and u, v are related by equation (1) and (2) where $f_x$ and $f_y$ are the scaled focal lengths, and $u_0$ and $v_0$ are the coordinates of the centre of the image *I*.

$$u=f_x\frac{X}{Z}+u_0 \tag{1}$$

$$v=f_y\frac{Y}{Z}+v_0 \tag{2}$$

We assume that the entire road surface is at a uniform level, thus the road surface is a flat plane. This is a decent assumption for urban roads. Consider a line L at certain depth $Z_0$ from the camera on the road surface, parallel to the image plane and perpendicular to the focal axis. Let this line intersect the road at points $P_1(X_1, Y_1, Z_1)$ and $P_2(X_2, Y_2, Z_2)$ with respect to camera coordinate system (Fig. 2.a). All the road points lying on the line L will have the same Y coordinate equal to -*H*, where *H* is the height of the focal centre with respect to the road. So from (2), we see that v remains constant for every point on the line L that is visible to the camera, as $f_y$, Y, $Z_0$ and $y_0$ are constant.

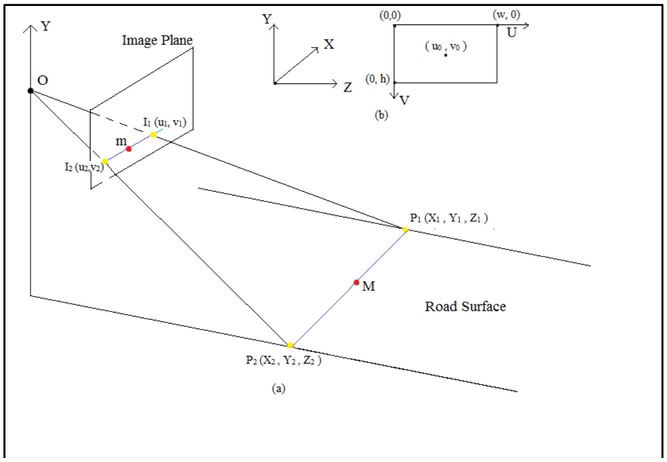

Fig. 2. (a) Depicts the projection of line segment ($P_1$, $P_2$) onto the image plane, where $I_1$ and $I_2$ are projections of $P_1$ and $P_2$, respectively. (b) Denotes the camera and the image coordinate system. U and V are the axis in the image coordinate system.

Let us consider the first case where the entire line segment ($P_1$, $P_2$) lies within the span of the camera at depth $Z_0$ (Fig 3.a). So for case (1), let $I_1(u_1, v_1)$ and $I_2(u_2, v_2)$ be the respective image points of $P_1$ and $P_{2.}$ Let M be the mid-point of $P_1$, $P_2$ and m be the mid-point of $I_1$, $I_2$ on image plane (Fig 2.a). It is simple to prove that the point on the image plane corresponding to M is m. When we substitute points $I_1$, $I_2$ in (1), we get

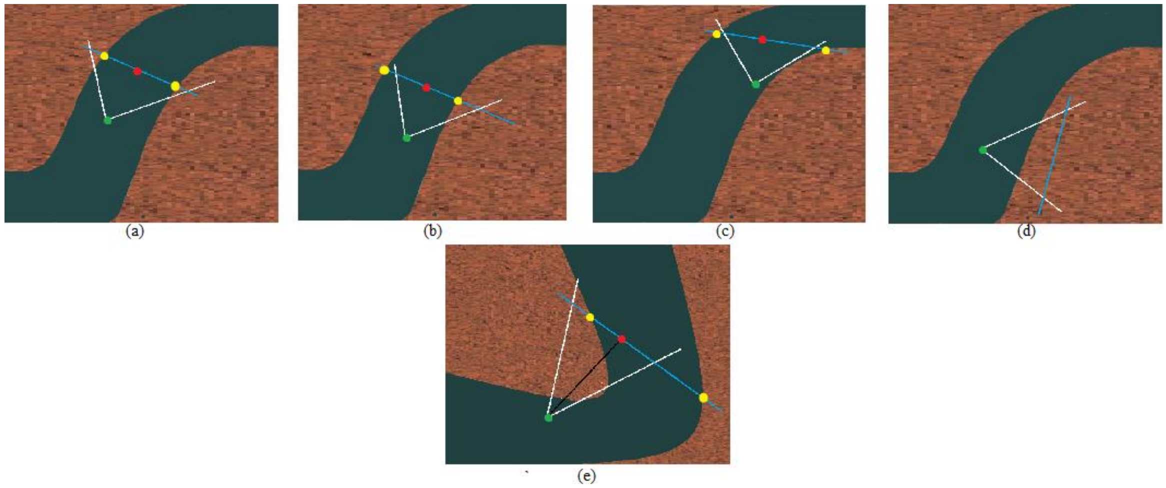

Fig. 3. The region between the white lines is the visible region for the camera. The green point indicates the position of the camera, the blue line indicates the line L, yellow points indicate the points $P_1$ and $P_2$, and the red point indicates the point M. The line passing through the green point and perpendicular to line L, is the focal axis of the camera. (a) Segment ($P_1$, $P_2$) is completely visible. (b), (c) Segment ($P_1$, $P_2$) is partly visible. (d) Segment ($P_1$, $P_2$) does not exist. (e) Line segment visible but line joining the robot and the reference point partly lies outside the road.

$$u_1=\frac{X_1 f_x}{Z}+u_0 \tag{3}$$

$$u_2=\frac{X_2 f_x}{Z}+u_0 \tag{4}$$

By adding (3) and (4) and dividing the result by 2, we get

$$\frac{u_1+u_2}{2}=\left(\frac{X_1+X_2}{2}\right)\times\frac{f_x}{Z}+u_0 \tag{5}$$

Thus (5) can be written as

$$m_u=M_x\times\frac{f_x}{Z}+u_0 \tag{6}$$

$M_x$ is the X-coordinate of M and $m_u$ is the U-coordinate of m. Similarly as V-coordinate of the entire line segment projected on the image plane is same, we have $v_1 = v_2$. Also the Y-coordinate of the entire road segment is same. Therefore we can write

$$m_v=M_y\times\frac{f_y}{Z}+v_0 \tag{7}$$

$m_v$ is the V-coordinate of the point m and $M_y$ is the Y-coordinate of the point M. Thus by (1), (2) and (6), (7), m is the corresponding projected image point of M. However we need to consider cases where the line segment ($P_1$, $P_2$) does not entirely lie within the span of the camera. The other possible cases are case (2), when one of $P_1$ and $P_2$ is visible, so only a part of the line segment is visible (Fig 3.b), and case

(3), when both $P_1$ and $P_2$ are not visible, and a part of the line segment is visible (Fig 3.c).There might even be a case (4) where such a line segment does not exist(Fig 3.d).However we address this problem later. For cases (2) and (3), the midpoint of the line segment cannot be found. However an approximate midpoint can be considered. Considering the description discussed above we frame thealgorithm as follows. Let $V_0$ be such that $h > V_0 > v_0$, where h is the height of the image frame (Fig 2.b), $v_0$ is the V-coordinate of the centre of the image. $V_0$ has to be greater than $v_0$, for the line L (which is projected as *l* in the image) to intersect the road. For $V_0 = v_0$, the line L will be at infinity. Draw Line *l:* $V= V_0$ on the binary image *I*.If case (1) occurs then entire line segment ($P_1$, $P_2$) is visible, therefore the line *l* intersects the segmented road region at $I_1(u_1, v_1)$ and $I_2 (u_2, v_2)$ such that $0 < u_1 < u_2 < w$, where w is the width of the image *I* (Fig 2.a). If case (2) occurs, the line segment is partly visible and one of $P_1$, $P_2$ is visible. If $P_1$ is visible then $I_1(u_1, v_1)$ will be the intersection point of *l* and road region in *I.* Similarly if $P_2$ is visible, then$I_2(u_2, v_2)$ will be the intersection point of *l* and road region in *I.* Then either $u_1= 0$ and $u_2 < w$ or $u_1> 0$ and$u_2 = w$, depending upon which one of $P_1$, $P_2$ is visible. Due to limited span, the other intersection point cannot be calculated and hence the intersection of the line *l* with the image frame is taken as the other point (Fig 1.b). For case (3), the line segment is partly visible and both $P_1$ and $P_2$ are not visible. The intersection points of *l* with the image frame are considered for calculating the reference point. Therefore $u_1 = 0$ and $u_2 = w$.For all cases naturally $v_1 = v_2 = V_0$. Let m be the midpoint of $I_1(u_1, v_1)$ and$I_2(u_2, v_2)$. We consider m as our reference point. The algorithm described gives a reference point for the three cases, where some part of the line segment is visible. But there might be a case where such a line segment does not exist (Fig. 3.d). In order to avoid such a situation we need to choose a suitable value for $V_0$. There is no fixedalgorithm to find the value, as it depends on the shape of the road. There is also a case (5) (Fig. 3.e) where a part of the line segment is visible. However the line joining the robot's position and the reference point is not completely inside the road. Higher the value of $V_0$, lower will be the depth $Z_0$ and therefore less chances of cases (4) or (5) occurring. Thus, we keep $V_0$ as high as possible. We take the value of $V_0$ to be around 80-90 percent of the image height. Increasing the value of $V_0$ makes the robot move more towards the edges. This is because more the value of $V_0$, less is the depth $Z_0$. Thus case (3) will occur, until it reaches near the edge at which case (2) occurs and the robot starts to turn. Thus the parameter $V_0$ determines how early a transition from case (3) to case (2) occurs, assuming the robot started at an initial position which had the configuration of case (3). Figure 4 depicts the shift towards the road boundary when $V_0$ is increased. Also, a point to note is that for case (3), the robot will maintain is original direction, as the reference point will be aligned with the line $l_1$:$U = u_0$, which is the central axis of the image. This is assuming that the camera is centrally mounted on the robot, and the direction of the robot is same as the direction of the camera's focal axis at every instance.

So far in the paper we have assumed the road to be planar. However the algorithm can also be used for a non-planar road. For such a case, $I_1$ and $I_2$ do represent certain points on the roads but may not be at the same depth. But the midpoint m will be bounded by the road boundaries and the same algorithm will work.

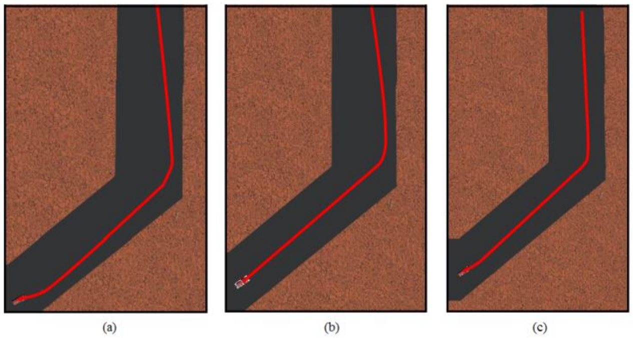

Fig. 4. These images are depictions of the travelled path of the robot for a fixed road withvarying parameter $V_0$. The red curve indicates the path followed by the robot when $V_0$ is at (a)90% of the height. (b) 85% of the height. (c) 80% of the height. The curve shifts more toward the road boundary at the turn as $V_0$ increases.

## 3. Results

As mentioned above, we simulated our navigation algorithm on Webots (Web-1) on multiple roads of varying shapes. All roads were wide enough to implement the road segmentation algorithm described above and were at a uniform level. Thus, the bottom part of the image captured by the camera, centrally mounted on the robot, consisted only of road pixels, in accordance with the standard assumption of the road segmentation algorithm. We used the standard PID algorithm to drive the robot, where the error $e$ was defined by the difference in U-coordinates of the reference point and the image centre (Equation 8).

$$e = \mathrm{m}_u - \mathrm{u}_0 \tag{8}$$

Figure 5 represents the working of the algorithm in discrete steps. For every point $M_i$, the robot approaches $M_{i+1}$ after running the algorithm at $M_i$. $M_i$ is not the reference point, but the point at which the algorithm is applied in discrete steps. However, directed line segment $(M_i, M_{i+1})$, has the same direction as $(M_{i,} R_i)$, where $R_i$is the reference point computed at $M_i$. Now $M_0$ = O, the initial position of the robot.

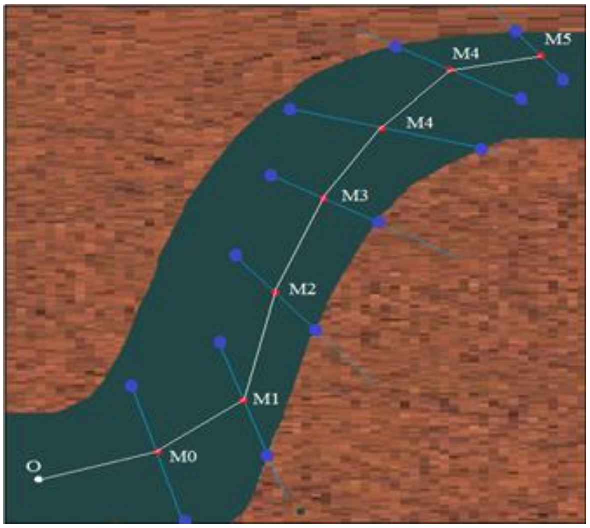

Fig. 5. O is the initial position of the robot. It heads towards $M_0$, by using the algorithm for case (3). $R_0$, the reference point computed at O, lies on the extended line segment $(O, M_0)$. The robot then heads towards $M_{0.}$ Similarly it computes $R_i$ and heads towards $M_i$. The blue points indicate $I_1$ and $I_2$.

The results for a minimal time step between successive implementation of the algorithm are shown in Figure 6.

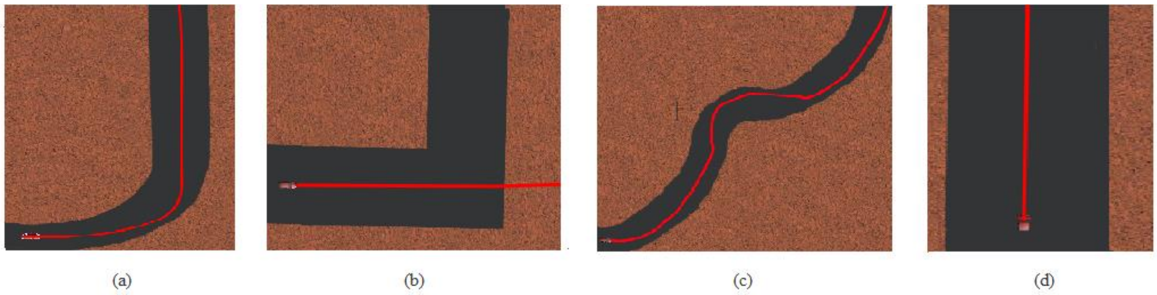

Fig. 6. (a) Indicates a hard left turn. The robot almost moves centrally with respect to the road. (b) Failure of the algorithm for $90^0$ turn. (c) Navigation for a spline road. (d) For straight road, the robot moves centrally.

For right angle turns, the robot shifts its position from case (3) to case (4) suddenly and therefore continues in the previous direction of navigation. We tested on various other shapes as well. For all roads except the ones with right angled turns or turns harder than a right angle, we achieved 100 percent accuracy, where the accuracy is defined to be the fraction of the distance, covered by the robot, on the road. For straight roads, when the robot is placed centrally, facing in direction of the central axis, our algorithm will naturally give the true road centre at every point.

## 4. Conclusion

We conclude that under the assumptions described, the algorithm in section (2) can be used for autonomous navigation. We intend to implement the same algorithm and test it on a physical robot in our future works. We tested our navigation algorithm for different planar roads using value of $V_0$ as 80-90 percent of the height of the image. We consider this value as safe for navigating in most of the scenarios. This value is dependent on the geometry of roads, and hence impossible to determine exactly, to guarantee autonomous navigation. We assume that our value of $V_0$ is such that the depth $Z_0$ is suitable to avoid cases (4) and (5).Due to limited span, the true road center cannot be calculated hence accuracy is measured in terms of the fraction of the distance travelled by the robot, on the ground.

## 5. References

Alvarez, J.M, Lopez, A., Baldrich, R.(2008) Illuminant-Invariant Model-Based Road Segmentation, Intelligent Vehicles Symposium, 2008 IEEE, On page(s) 1175 - 1180 .

Chin-Kai Chang, Siagian, C., Itti, L.(2012) Mobile robot monocular vision navigation based on road region and boundary estimation, Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pp.1043 - 1050 .

Frew, E. ,McGee, T. , ZuWhan Kim , Xiao Xiao , Jackson, S. , Morimoto, M. , Rathinam, S. , Padial, J. , Sengupta, Raja.(2004) Vision-based road-following using a small autonomous aircraft, Aerospace Conference, 2004. Proceedings 2004 IEEE (Volume 5) pp. 3006-3015.

H. Kong, J.Y. Audibert, J. Ponce.(2010) General road detection from a single image, IEEE Transactions on Image Processing, vol. 19, no. 8, pp. 2211 - 2220.

Siagian, C., Chin-Kai Chang, Itti, L.(2013) Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition, 2013 IEEE International Conference on Robotics and Automation(ICRA), pp. 564–571.

Tan, C.,Tsai Hong ,Chang, T., Shneier, M. (2006) Color Model-Based Real Time Learning for Road Following, Intelligent Transportation Systems Conference, 2006. IEEE, pp. 939 – 944.

Web sites:

Web-1: Webots. http://www.cyberbotics.comCommercial Mobile Robot Simulation Software.