VEHICLE COLLISION DETECTION IN IMAGE PLANE

Kunal A Khochare
B.E., Atharva C.O.E, India, 2007

PROJECT

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL AND ELECTRONIC ENGINEERING

at

CALIFORNIA STATE UNIVERSITY, SACRAMENTO

SPRING
2011
VEHICLE COLLISION DETECTION IN IMAGE PLANE

A Project

by

Kunal A Khochare

Approved by:

__________________________________________________, Committee Chair
Fethi Belkhouche, Ph.D.

__________________________________________________, Second Reader
Preetham Kumar, PhD.

Date
Student: Kunal A Khochare

I certify that this student has met the requirements for format contained in the university format manual, and that this project report is suitable for shelving in the Library and credit is to be rewarded for the Project.

_________________________ , Graduate Coordinator                  ________________
Preetham B. Kumar, Ph.D                  Date

Department of Electrical and Electronic Engineering
Abstract

of

VEHICLE COLLISION DETECTION IN IMAGE PLANE

by

Kunal A Khochare

In today’s world, many car accidents occur due to driver’s behavioral errors. Some of these accidents can be avoided by using intelligent driver assistance systems. Such systems must include a collision detection component. Different sensors can be used including vision sensors to detect whether there is a collision risk with other objects or vehicles. The main aim of this project is to design a system which uses a camera mounted on the test vehicle that will detect any preceding vehicle in the image frame. Depending upon the distance travelled by both vehicles in a particular time interval, this system can predict whether the vehicle in the frame will collide with the test vehicle. Ego motion and optical flow are some of the techniques used to detect vehicles. Ego motion is used to determine the 3D position of the camera mounted on an object relative to its surrounding whereas optical flow is used to detect moving vehicles by capturing image frames using optical flow vectors and comparing them with the model vectors depicting expected vehicle motion. In order to detect objects in the image frames, it is necessary to map the
image frame coordinates to the world coordinates. With the help of this mapping, the coordinates of the object in the real world can be found out. Also, as the image is made of pixels, it is necessary to find out the relation between the image coordinates and the pixel coordinates which is done by camera calibration. Finally, by converting the video captured by the camera into image frames, a collision detection method is used to predict if the cars are going to collide by calculating the distance travelled by the cars in the particular time intervals. In conclusion, this project aims to predict collision of vehicles in image plane by analyzing the images obtained from the video captured by the camera.

__________________________________________, Committee Chair
Fethi Belkhouche, Ph.D

__________________________
Date
ACKNOWLEDGEMENTS

I would like to extend my gratitude to my project advisor Dr. Fethi Belkhouche, assistant professor, Electrical and Electronic Engineering, for guiding me throughout this project and helping me in completing this project successfully. This project would not have been possible without the continuous support and encouragement from Dr. Fethi Belkhouche. I am also thankful to Dr. Preetham Kumar, Professor, Electrical and Electronics Engineering, for reviewing my report and providing valuable feedbacks. In addition, I would like to thank the Department of Electrical and Electronics Engineering at California State University for extending this opportunity for me to pursue this program and guiding me all the way to become a successful student.

Finally, I would like to thank my parents Arun Khochare and Anuradha Khochare and my friends for providing me all the support, strength and inspiration while working on this project and helped me to pass this critical phase of my graduation.
TABLE OF CONTENTS

Acknowledgements........................................................................................................vi

List of Tables................................................................................................................ix

List of Figures................................................................................................................x

Chapter

1. INTRODUCTION.................................................................1

2. RELATED WORK...............................................................3

3. EGO MOTION.................................................................21
   3.1 Introduction..........................................................21
   3.2 Vehicle ego motion estimation...............................23
   3.3 Moving object detection.......................................26

4. OPTICAL FLOW..........................................................27
   4.1 Introduction........................................................27
   4.2 Optical flow vector calculations..........................28

5. TRANSFORMATION OF COORDINATES......................29
   5.1 Relation between world coordinates and image coordinate.....29
   5.2 Perspective projection........................................31
   5.3 Relation between the image coordinates and the pixel coordinates.....32

6. CAMERA CALIBRATION...............................................33
6.1 Introduction........................................................................................................33
6.2 Extrinsic Camera Parameters..............................................................33
6.3 Intrinsic Camera Parameters..............................................................34
7. SIMULATION AND CALCULATIONS.......................................................36
  7.1 Introduction..................................................................................................36
  7.2 Case I........................................................................................................39
  7.3 Case II.........................................................................................................46
8. CONCLUSION AND FUTURE WORK....................................................52
Appendix.........................................................................................................54
References......................................................................................................56
LIST OF TABLES

1. Table 4.1: Distance travelled by both the cars and collision detection ratio in case I .................................................................40

2. Table 4.2: Different car positions at different time interval in case I ............45

3. Table 4.3: Distance travelled by both the cars and collision detection ratio in case II .........................................................................47

4. Table 4.4: Different car positions at different time interval in case II ............51
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Figure 2.1: Hardware structure</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Figure 2.2: Flowchart of the driver assistance system</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>Figure 2.3: Image plane and world coordinates definition</td>
<td>5</td>
</tr>
<tr>
<td>4.</td>
<td>Figure 2.4: Forward Collision Warning (FCW) Systems</td>
<td>8</td>
</tr>
<tr>
<td>5.</td>
<td>Figure 2.5: Real world coordinates system</td>
<td>10</td>
</tr>
<tr>
<td>6.</td>
<td>Figure 2.6: Feature extraction sub windows</td>
<td>15</td>
</tr>
<tr>
<td>7.</td>
<td>Figure 2.7: Real world coordinates and video camera setup</td>
<td>17</td>
</tr>
<tr>
<td>8.</td>
<td>Figure 2.8: Image domain coordinates</td>
<td>18</td>
</tr>
<tr>
<td>9.</td>
<td>Figure 2.9: The vanishing point and calibration</td>
<td>20</td>
</tr>
<tr>
<td>10.</td>
<td>Figure 3.1: Flow of proposed method</td>
<td>22</td>
</tr>
<tr>
<td>11.</td>
<td>Figure 3.2: Process flow for each frame</td>
<td>22</td>
</tr>
<tr>
<td>12.</td>
<td>Figure 3.3: Image divided into three regions</td>
<td>25</td>
</tr>
<tr>
<td>13.</td>
<td>Figure 3.4: Feature point selection</td>
<td>25</td>
</tr>
<tr>
<td>14.</td>
<td>Figure 5.1: Camera coordinate frame</td>
<td>30</td>
</tr>
<tr>
<td>15.</td>
<td>Figure 7.1: Scenario of two cars moving in perpendicular direction</td>
<td>36</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

In today’s world, intelligent transportation systems are used in order to improve safety, reduce vehicle wear, transportation times and fuel consumptions. Improvements in vehicle designs have steadily reduced injury and death rates in many countries as vehicle collision is one of the main reasons of injury – related deaths. There are two terms involved in vehicle safety namely active safety and passive safety. Active safety refers to an automatic technology such as driver assistance system which helps to prevent accidents whereas passive safety refers to the parts of the vehicle such as airbags, seatbelts etc which provide protection during accidents [1]. This project deals with one aspect of active safety which consists of collision detection.

One of the systems used today in vehicles is ADAS (Advanced Driver Assistance System). There are many features associated with the ADAS which provide safety and protection. Some of these features are lane-departure warning, night vision, adaptive cruise control and blind-spot monitoring. These features are used to avoid vehicle collision in advance unlike passive safety features which become active or are useful after the crash occurs. As passive safety features only decrease injury or improve the chances of survival in an accident, the ADAS features prevent the accident to occur in first place such as by controlling the car automatically. Features like collision detection
systems help to apply car’s brake automatically or lane departure prevention system help to steer the vehicle back on track [2].

Adaptive cruise control (ACC) is one of the most commonly known ADAS features used to detect vehicles in front and adjust the car’s speed accordingly. The system which consists of such feature generally employs a radar or laser sensors to monitor vehicles ahead and adjust the speed of the host vehicle. Such system can provide additional features, if it detects that the collision is imminent, such as warn driver by flashing lights and activating sounds, pre-charge the brakes, tighten seatbelts, close windows and sunroof etc. A good driver assistance system which has all or some of the above features can detect vehicles in the surrounding and predict whether collision can occur between two vehicles effectively [2]. However, since the road environment goes through a huge spectrum of vehicle distribution, vehicle shape, weather conditions, structures by the roadside, tunnel, light change in day and night times, it is vital to develop the image processing and pattern recognition algorithm that can precisely and reliably detect vehicles and extract information for vehicle detection and tracking.

Advantages of advanced driver assistance system can be summarized as follows:

• Maintain and keep a safe speed.

• Helps detect vehicle lane departure and prevent overtaking in critical situations.

• Provides safely passing at intersections.

• Avoid collisions with vulnerable road users [3].
Chapter 2

RELATED WORK

In this chapter, we discuss some of the techniques used in vision-based driver assistance systems. Development of a vision-based driver assistance system has been discussed by many researchers. The method used in [4] utilizes driver assistance system which can detect lane markings and also vehicles in front of the car by mounting a CMOS camera on it and also with the help of image processing unit. The lane boundary deviation and the distance between the two vehicles can be calculated by road geometry model and distance estimation model. The image detection method mentioned in this article uses the dynamic region of interest (ROI) to increase the real time processing and also helps to reduce the noise interference from the image. If the vehicle tries to make undesirable lane change or if it gets near to the vehicle in front then this driver assistance system will detect the changes and alert the driver by sound and light, and helps the to make necessary changes like correct driving directions or slow down the speed of the vehicle [4]. The block diagram and the flowchart of the driver assistance system are shown in the following figure 2.1 and figure 2.2, respectively.
Figure 2.1: Hardware structure [4].

Figure 2.2: Flowchart of the driver assistance system [4].
The relationship between the camera coordinates and the world coordinates is shown in the Figure 2.3:

![Figure 2.3: Image plane and world coordinates definition](image)

As the lane markings and the other vehicles are mapped in the image plane obtained from the video frames, the obtained 2D image should be transformed into 3D space through inverse mapping in order to obtain the real space positions. The 2D image coordinates can be converted into the world coordinates by following equations:

\[
x = \left( \frac{uH}{e.m\theta - v} \right) \left( \frac{e_v}{e_u} \right) \quad \text{...(2.1)}
\]

\[
y = \left( \frac{e.H}{e.m\theta - v} \right) \quad \text{...(2.2)}
\]

\[
z = \left( \frac{e.m\theta H}{e.m\theta - v} \right) \quad \text{...(2.3)}
\]
The parameters given in the above equations are as follows:

\( u \): x axis in image plane
\( v \): y axis in image plane
\( H \): Height of camera from ground
\( k, m, b \): Coefficients of road geometry
\( \theta \): Tilt angle of camera
\( W \): Actual lane width
\( X_w, Y_w, Z_w \): World Coordinates
\( e_u \): Pixel width of CMOS Sensor
\( e_v \): Pixel height of CMOS Sensor

According to the \( x, y, z \) position data from the above equations and by using least squares processing, a road geometry model can be obtained which can be used to detect the positions of the lane markings with respect to the vehicle. Further, the lane departure detection is done with the help of high gray value, range of lane marking width, etc. In order to enhance the search speed for lane marking, we used the road geometry model to limit the basic detecting zone within detectable range. Also, the forward collision detection is predicted. In order to obtain the distance between the host vehicle and preceding vehicle, the process of headway estimation is composed of three stages. The first stage is preceding vehicles detection, the second stage is vehicle tracking and the final stage is distance estimation [4].
Advantages:

(1) This system can be used under all the conditions and any time of the day and night.

(2) This system can be used for any lane boundaries detection and any vehicle detection.

(3) By using this system, the lane deviation and vehicle detection can be done accurately in different environmental conditions.

(4) This system provides image processing rate of 20 fps which helps in real time computing in an embedded system [4].

Vehicle localization is another important aspect of vehicle collision detection. In [5], a forward collision warning (FCW) system is used to avoid collision between vehicles using a single camera and also a new method is described for vehicle detection namely appearance-based hypothesis generation, template tracking-based hypothesis verification which can be used to reduce the noise interference and automatic image matting for fine image detection. The main aim of the Forward Collision Warning (FCW) system is to provide vehicle detection. The estimated time to collision with the preceding vehicle is called as Time to Collision (TTC). The single camera based Forward Collision Warning (FCW) system has following two parts: vehicle detection and detection refinement [5]. The block diagram of FCW is shown in the figure 2.4:
Firstly, Haar and Adboost algorithm [5] appearance-based as hypothesis generation is carried out, which is used for vehicle detection. Secondly, simplified Lucas-Kanade Algorithm and Virtual edge [5] is used to remove error detection and use of image matting to do detection refinement. Thirdly, hierarchical tracking system is introduced for vehicle tracking; camera calibration is utilized to get the headway distance and TTC.

Hypothesis generation and hypothesis verification are the two steps used to detect vehicles in the video frames based vehicle detection method. In hypothesis generation, the positions of the objects in the image are hypothesized and in hypothesis verification

Figure 2.4: Forward Collision Warning (FCW) Systems [5].
the objects present in the hypothesized regions are checked. As regards the first step, hypothesis generation, HG approaches can be classified into one of the two ways: (1) knowledge-based, (2) motion-based. Knowledge-based methods give information about the shape and color of the vehicle as well as general information about streets, roads, and freeways [5]. A motion-based approach is the one that uses projective invariant and vanishing lines to derive the motion constraint of the ground plane and the surface plane of the vehicles. As for hypothesis verification step, two categories are used: template-based and appearance-based methods. The template-based uses a predefined pattern of the vehicle class and performs a correlation between an input image and the template. Appearance-based methods acquire the properties of the vehicle class from various image frames which captures the variability in vehicle appearance. To improve performance, the variability of non-vehicle class is also modeled. There is a drawback of hypothesis verification methods i.e. if there is a complex environment then there will be false detections by hypothesis generation such as person, tree, shadow and so on. But the hypothesis verification only can remove some kinds of the false detections, because both template-based and appearance-based methods can only model one or some of the false detections [5]. In [5], appearance-based hypothesis generation is used for vehicle detection and template tracking-based hypothesis verification is used to remove the false positive detections. Then, image matting for detection refinement is discussed in this article. Accurately separating a foreground object from the background is known as image matting. In vehicle detection step, only the approximate location of the vehicle is
known. But, it is necessary to localize the vehicle accurately. So image matting is used for detection refinement instead of image segmentation. And image matting is also an initialization step for template tracking. Image matting is done only once. After image matting, the vehicle can be accurately located. Time to Collision (TTC) can be computed using camera calibration which is based on vanishing point. Usually, the vanishing point locates outside images, may even be infinitely far away. All the vanishing points of a spatial reference plane are on the vanishing line [5]. The real world coordinate system is shown in Figure 2.5.

![Figure 2.5: Real world coordinates system [5].](image)

The locations of host vehicle and the target vehicle are point H and D. Let the width be W and its length be L of the rectangle OABC. Quadrangle oabc in Figure 2.5 is the result of projective transform rectangle OABC from real world coordinate system into image.
coordinate system. The coordinates of D in real world coordinate can be found by using
the coordinates of a, b, c, o, d in the image coordinate system. Since line AB and line OC
are parallel, the cross point v1 of line oc and line ab is the vanishing point of lines in
image. And all lines parallel with line AB in real world vanish at this vanishing point in
image. Similarly, the cross point v2 of line oa and line bc is the vanishing point of image
lines corresponding to real world lines which are parallel with line OA. The cross point m
of line v2d and line ab can be calculated with the vanishing points, and point m is the
corresponding point of M, the projective point of D on line AB. The cross point n of line
v1d and line oa can be calculated, and point n is the corresponding
point of N, the
projective point of D on line OA. According to crossing ratio invariability,

\[(oa;nv2)=(OA;NV2)\] .......................... (2.4)

\[(bm;av1)=(BM;AV1)\] .......................... (2.5)

For v1, v2 are infinitive far away, we get formulas as following:

\[\frac{(on\times av2)}{(an\times ov2)}=t\] .......................... (2.6)

where, t is a constant value.

\[ON = AN \times \frac{(on\times av2)}{(an\times ov2)} = (L-ON)\times t = \frac{(L\times t)}{(1+t)}\] .......................... (2.7)

\[H = HD = HE + DE = HE + ON\] .......................... (2.8)

where, the distance between host vehicle and the target vehicle is called headway
distance H. The length of blind area that the camera cannot cover is called HE.
The FCW system uses time to collision (TTC) to trigger the warning and is issued when the TTC is lower than a certain threshold (2 seconds in this experiment). The momentary Time To Contact (Tm) is defined as

\[ Tm = \frac{H}{V} \]  \hspace{2cm} (2.9)

Where \( V \) is the relative velocity, \( H1 \) and \( H2 \) is the distance between two vehicles in two consecutive frames. When the time interval \( \Delta t \) between \( H2 \) and \( H1 \) is small we can write (\( \Delta t = 1/29s \) in our experiment):

\[ H2 = H1 - V\Delta t \]  \hspace{2cm} (2.10)

\[ V = \frac{(H1 - H2)}{\Delta t} \]  \hspace{2cm} (2.11)

The use of a single low cost camera results in an affordable system which is simple to install. The FCW system has been tested in outdoor environment, showing robust and accurate performance [5].

Feature extraction is done by Gabor filters and Support Vector Machines which are used for vehicle detection in [6]. Gabor filters provide a mechanism for obtaining selectivity in scale, some degree of invariance to intensity due to global illumination, as well as selectivity in orientation. Gabor filters and SVMs are basically edge and line detectors. Feature extraction using edges and lines of the vehicle can be used for vehicle detection as vehicles have accurate edges and lines. In order to obtain the edge and line information
of the vehicle, the original image is sub divided into sub windows from which several sub images are obtained. These features are then used to train two class classifiers known as SVM classifiers. The information of the vehicle obtained in the image frame can be generalized by using these classifiers which are used for structural risk minimization. These Gabor filters and SVM classifiers provides greater achievement in vehicle detection [6].

GABOR FILTERS AND SVMs REVIEW:

Gabor Filters:
The Gabor filter \( g (x, y) \) can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal as follows:

\[
G(x, y) = \left[ \frac{1}{(2\pi\sigma_x\sigma_y)} \right] . \exp \left[ -0.5 \left( \frac{\bar{x}^2}{\sigma_x^2} + \frac{\bar{y}^2}{\sigma_y^2} \right) \right] . \exp \left[ 2\pi jW\bar{x} \right] \quad \ldots \quad (2.12)
\]

where,

\[
\bar{x} = x \cos \theta + y \sin \theta \quad \ldots \quad \ldots \quad (2.13)
\]

and

\[
\bar{y} = -x \sin \theta = y \cos \theta \quad \ldots \quad \ldots \quad (2.14)
\]

Here,

\( \sigma_x \) and \( \sigma_y \) : the scaling parameters of the filter

\( W \) : center frequency
$\theta$: the orientation of the filter

Gabor filters act as local band pass filters. If $I(x, y)$ is an input image then feature extraction is done by convolving $I(x, y)$ with the Gabor filters. These filters are used to extract features based on moments from several sub windows of the input image.

SVMs:

In order to extract features using edge and line detectors, SVMs which are two class classifiers are used to learn about linear and non-linear decision boundaries. The two class classifiers have set of points which are divided by the hyper plane, leaving the class having largest fraction of points on the same side while increasing the distance of both the class from the hyper plane. This is similar to minimize the structural risk in order to achieve good generalization. Consider $l$ examples from two classes

$$(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l), x_i \in \{-l, +l\} \cdots \cdots \cdots \cdots (2.15)$$

to find an optimal hyper plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The width of the margin between classes is the optimization criterion. The discriminate hyper plane is defined as:

$$f(x) = \sum_{i=1}^{l} y_ia_k(x, x_i) + b \cdots \cdots \cdots \cdots (2.16)$$

where, $k(x, x_i)$ is a kernel function and sign of $f(x)$ indicates the membership of $x$ [6].
GABOR FEATURE EXTRACTION:

The Gabor feature extraction is done by using the Gabor filters. The sub images of the hypothesized vehicle obtained from the entire image are given as inputs to the feature extraction subsystem. Here, each sub image used for feature detection is scaled to 64 x 64 fixed size. Then, it is subdivided into 9 overlapping 32 x 32 sub windows. If the sub image is divided into 16 x 16 patches as shown in figure 2.6, patches 1,2,5 and 6 comprise the first 32 x 32 sub window, 2,3,6 and 7 the second, 5,6,9, and 10 the fourth, and so forth [6].

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

Figure 2.6: Feature Extraction Sub windows [6].

Then on each sub window, the Gabor filters are applied separately. The Gabor filters are extracted from several overlapping sub windows in order to compensate for the errors in the hypothesis generation step and hence the feature extraction is more reliable.

From each window, the magnitudes of the Gabor filter responses are extracted and are represented by three moments: the mean, the standard deviation and the skewness. As
only the moments are used, the position information is discarded and only the statistical properties of the group pixels are considered. To evaluate the Gabor filter’s performance, three-fold cross validation method is used to record the average accuracy ($AR$), false positives ($FPs$), and false negatives ($FNs$). Specifically, the training dataset is split randomly three times (Set 1, Set 2 and Set 3) by keeping 80% of the vehicle sub images and 80% of the non-vehicle sub images for training.

In [7], the speed of the vehicle is detected accurately and automatically by using digital image processing and camera optics. The speed of the vehicle in multiple lanes can be detected with the help of a single camera and an onboard processing computer. In the first part of this article, the coordinates in image domain are mapped to the world domain. And in the second part of [7], the procedure of detecting the vehicle in the image frames obtained from the video stream is mentioned.

The mapping of coordinates from image domain to real-world domain:
In order to detect the vehicle in the image frame, it is necessary to mapped the coordinates from image domain to the real world domain i.e. 2D to 3D mapping by using some additional constraints. Since, the vehicle travels on the road surface which is planar and does not leave the road surface, the motion of vehicle on the road becomes a 2D motion and mapping of coordinates from image domain to real-world domain becomes a
2D-2D mapping which can be determined accurately [7]. The transformation of coordinates from image domain to real-world domain can be done as follows:

As shown in figure 2.7, the distance between the camera and the road surface is $H$ and the camera’s optical axis is tilted at an angle $\theta$, from the forward direction of the road. The transformation of coordinates from image domain to real-world domain can be done with the help of geometrical optics. As shown in figure 2.8, the camera can be simplified as a convex lens with focal length $f$. 

Figure 2.7: Real world coordinates and video camera setup [7].
Consider:
the distance between an object and lens plane is $z$,
the distance between an object and the optical axis of the lens is $y$,
object’s image has a height $v$,
the distance to the lens plane is $w$. 

\[
\frac{1}{f} = \frac{1}{z} + \frac{1}{w} \quad \ldots \quad (2.17)
\]

\[
\frac{v}{y} = \frac{w}{z} \quad \ldots \quad (2.18)
\]

\[
\frac{v}{w} = \frac{y}{z} \quad \ldots \quad (2.19)
\]
As, the distance between vehicle and lens is greater than 5 meters and generally the camera’s focal length is very small (5 centimeters or less) therefore \( z \gg f \), so

\[
w = \frac{zf}{(z-f)} = f \quad \text{…………………………………… (2.20)}
\]

Here, \( w \) is constant after the camera setup.

This proves that the imaging plane is very close to the focal point. As shown in figure 2.7, there are 3 coordinate systems. The first is the real-world coordinate system \((x, y, z)\), where \( x \), \( y \), and \( z \) represent transverse direction in the road surface plane, the direction perpendicular to the road surface, and longitudinal or forward direction in the road surface, respectively. The second is the camera coordinate system \((x', y', z')\), where \( x' \) represents the horizontal direction in the lens plane and \( y' \) represents the direction perpendicular to \( x' \) direction in the lens plane, \( z' \) represents the optical axis direction. The third coordinate system is the image domain system \((u, v)\), with \( u \) for horizontal direction and \( v \) for vertical direction (unit is pixel). There is a direction mapping relationship between camera coordinate system and image coordinate system as shown in equations.

But, it is necessary to find out the relationship between real-world coordinate system and image coordinate system [7].

The Calibration of the Coordinates Mapping:

The calibration is carried out as follows. As shown in Figure 2.9, the position of the vanishing point \((U_v, V_v)\) of the road surface from the intersection of two edge mark lines
is obtained. Then, a car of known length $L$ and $W$ is used in the image to do the calibration.

![Figure 2.9: The vanishing point and calibration [7].](image)

Vehicle detection in video images:

The method used for vehicle detection in [7] is used to get the adaptive background, which is necessary for accurate segmentation for the vehicle. Then, the video images are used to compare with background image to extract the foreground.
Chapter 3
EGO MOTION

3.1 Introduction:

Ego Motion is used to determine the 3D position of the camera mounted on an object relative to its surrounding. For example, by using an onboard camera, the location of the car can be determined by finding its position with respect to the objects in the surrounding such as trees, sign boards etc. The main objective of determining the ego motion of the camera is to determine the 3D position of the camera with respect to the objects in the surrounding by obtaining different image frames from the video taken by the camera [8]. Feature detection techniques are used to determine the optical flow from different image frames. There are various ways for estimating the vehicle ego motion and detection of the moving vehicles in the camera frames. One such method is discussed here.

In this method, the ego motion of the vehicle is estimated by feature detection of objects and comparing the positions of moving vehicles in different frames. The process flow used in this method is shown in figure 3.1 [9].
Figure 3.1: Flow of proposed method [9].

The internal parameters of the camera are already determined i.e. the camera is calibrated. The main focus of this method is that the moving vehicles and the objects are detected in the current image frame. The process flow for each frame is shown in the figure 3.2 below.

Figure 3.2: Process flow for each frame [9].
According to the above figure, the moving vehicles and other objects are detected from two consecutive images so that the difference in the two images can be detected and measured. For this process, two image frames, one at time $t - 1$ and one at time $t$ are used for ego motion estimation and object detection at time $t$. It is assumed that in the initial frame there are no moving vehicles prior to this time and the road area is calculated depending upon the camera’s height on the stationary vehicle. Here the Harris corner detector is used to extract the feature points and Lucas-Kanade method [9] is used to detect the difference in the feature points on the two image frames which is then used for ego motion estimation [9]. In order that the ego motion estimation results to be accurate, the feature points are detected from the previous frames over a widely distributed area which does not contain moving vehicles. Then, the road and the moving vehicles are detected after the reconstruction of the surrounding scene from previous images. This process of moving vehicle and road detection and ego motion estimation continues for the next frames [9].

### 3.2 Vehicle ego motion estimation:

Here, some of the feature points are differentiated from the set of feature points and the associates of these selected points from various frames are used for ego motion estimation. Generally, the ego motion estimation has two problems while using the associates of the selected feature points. The first problem is that the features points selected for the moving vehicles can predict incorrect ego motion estimation. Also, there is difference in the number of feature points for objects in the image frames. So, there is a
problem in the associated feature points detection method from a single image frame due to the difference in the number of feature points in single image frame [9].

So, in order to eliminate the above mentioned problems, two consecutive image frames are used for object detection and ego motion estimation. First, the moving vehicles are detected in previous frame and the results are used to get the feature points of the previously detected moving vehicles. Then, the image frames obtained from the camera are divided into three regions namely

1. Region with the low height objects.
2. Region with the high height objects.
3. Region which contains road [9].

The image frame is divided such that the region that contains the road is at the bottom of the image according to the road region detection results obtained from the previous frame and the low height and high height objects regions are divided equally into the remaining region of the image frame. The image frame divided into the above three regions is shown in the figure 3.3.
As the image is divided into three regions, each region has different feature points and the number of feature points to be selected is pre-determined. The feature point selection is shown in the figure 3.4.

According to figure 3.4 (a), background objects has more feature points as compared to the road feature points if the whole image is used for feature point’s extraction. If
selected feature points are used as in figure (b), then the feature points are uniformly distributed throughout the image. Sometimes, there are many features points present on the moving vehicle which cannot be removed due to improper detection of moving vehicles in the previous frames but such feature points can be suppressed by selecting feature points from three different regions. Thus this method can be used accurately for vehicle detection and ego motion estimation.

The correspondence matrix of the selected feature points can be used to estimate the essential matrix using the 8 point algorithm [10] and RANSAC [11]. The essential matrix is used to estimate the motion parameters which consist of three rotational and three translational parameters.

3.3 Moving object detection:

The positions of the feature points in the image frame can be calculated using triangulation process. The points that have negative distance between them or that are away from their epipolar lines called as outlier points are detected in each frame. Feature points on moving objects along with false correspondence points together form a set of outlier points. In order to detect feature points present only on the moving vehicles, these points are detected and matched over consecutive image frames. Such points which are detected and tracked over consecutive frames called as outliers form a group of candidate points for moving vehicles. These candidate points for moving vehicle form a group depending upon their position in the image frame which then is defined as a rectangle which has all points of same group [9].
Chapter 4

OPTICAL FLOW

4.1 Introduction:

Optical flow evaluation is the technique that can be used to detect preceding moving vehicles and stationary objects in the image frame obtained from the camera mounted on the vehicle. In this technique, the optical flow vectors are used to determine the relative motion of the camera mounted on the vehicle and scene in the image obtained from the camera. Also, these optical vectors consist of the 3D information of the captured image from the camera [12].

The steps involved in using the optical flow technique for detection of moving vehicles and stationary objects are as follows:

1. Using the captured image frame sequences, calculate the optical flow vectors $u$.
2. Calculate the model vectors $u_M$ which depicts the expected image motion.
3. Calculate and evaluate the differences $u_D$ between the above two vectors called as $u$ and $u_M$.

The comparison between the optical flow vectors and model vectors which gives the expected image shift can be used to detect objects in the image frame. The optical flow vectors can be calculated by using different algorithms. To calculate the model vectors, the ego – motion of the camera can be obtained from the sensors on the experimental vehicles [12].
4.2 Optical flow vector calculations:

The optical flow vectors can be calculated by using a method which uses the following equation [13]:

\[
A \cdot u = \begin{pmatrix} g_x & g_y \\ g_{xx} & g_{yy} \\ g_{xy} & g_{yx} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -g_x \\ -g_{xx} \\ -g_{xy} \end{pmatrix} = B \quad \text{.......................................... (4.1)}
\]

The hardware module of the image sequence analysis system MiniVISTA is used to determine the first and second order spatio – temporal derivatives of the gray value function in video real-time. As each pixel has underlying gray value distribution, the spatio temporal derivatives at each pixel should pass the structure so that the gray value distribution has enough structure to locally estimate the optical flow vectors. The derivatives which pass this test can be passed to subsequent processes which are implemented on the transputer network. These processes are used to determine the expected model vectors, compute the optical flow vectors and perform the obstacle test [12].
Chapter 5

TRANSFORMATION OF COORDINATES

5.1 Relation between world coordinates and image coordinate:

In order to map the location of the object in the image plane with that of the object’s location in the real world, it is useful to map the objects coordinates to the camera centered coordinate frame. So, to simplify the camera coordinate frame, the image plane is considered to be containing the sensing array. The axes parallel to the vertical and horizontal axes of the image are called as $x_c$ and $y_c$ which forms the basis for the image place. The optical axis of the lens which passes through the focal center of the lens is aligned to axis $z_c$ which is perpendicular to the image plane [14].

The distance between the origin of the camera frame and the image place is $\lambda$. This point of origin located behind the image plane is called center of projection. The principal point is the intersection point of the optical axis and image plane. The coordinates of the any point present in the image plane will be $(u, v, \lambda)$ after assignment of the camera frame. So, the coordinates $(u, v)$ can be used to parameterize the image plane and are called as image plane coordinates [14]. The camera coordinate frame is shown in the figure below.
Figure 5.1 Camera coordinate frame [14].
5.2 Perspective projection:

Pinhole lens model is used to model the image formation process. Assuming this, the lens is said to be ideal pinhole that is present at the focal center of the lens. The image plane is intersected by the light rays passing through this pinhole. Consider point $P$ coordinates in the world be $(x, y, z)$ relative to the camera frame. Consider point $p$ coordinates which is projection of $P$ onto the image plane be $(u, v, \lambda)$. As per the pinhole model, points $P$, $p$, and the origin of the camera frame will be collinear. Thus, the relation between the image plane and the world coordinates for an unknown positive constant $k$ is given by:

$$
k \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} \quad \text{........................................ (5.1)}$$

The above equation can be solved as

$$kx = u \quad \text{........................................ (5.2)}$$

$$ky = v \quad \text{........................................ (5.3)}$$

$$kz = \lambda \quad \text{........................................ (5.4)}$$

From these equations, we can substitute $k = \lambda/z$ to get images coordinates $(u, v)$:

$$u = \lambda \left( \frac{x}{z} \right) \quad \text{........................................ (5.5)}$$

$$v = \lambda \left( \frac{y}{z} \right) \quad \text{........................................ (5.6)}$$

These equations of coordinates $(u, v)$ are known as perspective projection [14].
5.3 Relation between the image coordinates and the pixel coordinates:

Let the pixel coordinates for the row and column indices for the pixel be \((r, c)\). The digital images can be related to the 3D world by finding the relationship between the image plane coordinates \((u, v)\) and the pixel coordinates \((r, c)\). The pixel array origin is located at the corner of the image. Consider \((o_r, o_c)\) be the pixel array coordinates of the principal point. Let the horizontal and vertical dimensions of the pixel be \(s_x\) and \(s_y\) respectively. Also, consider that the horizontal and vertical axes of the camera coordinate frame points in the opposite direction to the horizontal and vertical axes of the coordinate system of the pixel array. Considering the above assumptions, the relationship between the image plane coordinates and pixel array coordinates obtained is as follows:

\[
-u = (r - o_r) \quad \text{..................................................} (5.7)
\]

\[
-v = (c - o_c) \quad \text{..................................................} (5.8)
\]

The coordinates \((r, c)\) are the discrete indices in the array which will be stored in the computer memory so these coordinates will be integers. The value of \((r, c)\) can be found by rounding off the LHS of the above equations [14].
Chapter 6

CAMERA CALIBRATION

6.1. Introduction:

The main aim of camera calibration is to find a relation between the pixel coordinates \((r, c)\) and the point coordinates in the camera’s field of view \((x, y, z)\). Here, after the camera is calibrated, the point coordinates in the world frame are converted into the image pixel coordinates for the projection of that point in the image frame. The camera calibration is used to find the internal parameters of the camera that affect the imaging process. Various parameters which affect the imaging process are the position of the image center, focal length, different scaling factors for row and column pixels, skew factor, lens distortions.

There are two types of camera parameters namely extrinsic camera parameters and intrinsic camera parameters [14].

6.2 Extrinsic Camera Parameters:

The coordinates used in previous equations are expressed relative to the camera frame. Usually, these equations are expressed relative to the world coordinate frame. If the location and orientation of the camera frame relative to the world coordinate frame is known then

\[
x^w = R_c^e x^e + O_c^e
\]

(6.1)
Here,

\( x^w \): position of object in world coordinate frame

\( x^c \): position of object in camera coordinate frame

If \( x^w \) is known then \( x^c \) can be solved as follows:

\[
x^c = R_c^x (x^w - O_c^x)
\]

(6.2)

The above equation can be simplify by substituting,

\[
R = R_c^x
\]

(6.3)

\[
T = -R_c^x O_c^x
\]

(6.4)

After substituting in the above equation,

\[
x^c = Rx^w + T
\]

(6.5)

Here, the parameters \( R \) and \( T \) are called as the extrinsic camera parameters.

### 6.3 Intrinsic Camera Parameters:

The equations obtained from perspective projection and relationship between world and camera frame coordinates are given as:

\[
u = \lambda \left( \frac{x}{z} \right)
\]

(6.6)

\[
v = \lambda \left( \frac{y}{z} \right)
\]

(6.7)
and

\[- \left( \frac{u}{s_x} \right) = (r - o_r) \] \hspace{1cm} \text{(6.8)}

\[- \left( \frac{v}{s_y} \right) = (c - o_c) \] \hspace{1cm} \text{(6.9)}

The relationship between the world coordinate frame and pixel coordinate frame can be obtained by combining the above equations:

\[ r = - \left( \frac{\lambda}{s_x} \right) \left( \frac{x}{z} \right) + o_r \] \hspace{1cm} \text{(6.10)}

\[ c = - \left( \frac{\lambda}{s_y} \right) \left( \frac{y}{z} \right) + o_c \] \hspace{1cm} \text{(6.11)}

If the values of the parameters \( s_x, s_y, o_r, o_c, \lambda \) are found out then \((r, c)\) can be determined from camera frame coordinates \((x, y, z)\). It is not necessary to know all the parameters such as \( \lambda, s_x, s_y \) as only the following ratios can be useful.

\[ f_x = \frac{\lambda}{s_x} \] \hspace{1cm} \text{(6.12)}

\[ f_y = \frac{\lambda}{s_y} \] \hspace{1cm} \text{(6.13)}

the parameters \( f_x, f_y, o_r, o_c \) are called as the intrinsic parameters of the camera and are constant for any camera even when the camera is moving [14].
Chapter 7
SIMULATION AND CALCULATIONS

7.1 Introduction:

Now, in order to test the system in vehicles for collision detection, a particular scenario is considered in which a camera is mounted on the test vehicle (car 1) and another vehicle (car 2) is coming towards the test vehicle in the direction perpendicular to the direction of the test car as shown in the figure 7.1.

![Figure 7.1: Scenario of two cars moving in perpendicular direction](image-url)
The equations of coordinates \((u, v)\) known as perspective projection are given by,

\[ u = \lambda \left( \frac{x}{z} \right) \]  \hspace{1cm} (7.1)

\[ v = \lambda \left( \frac{y}{z} \right) \]  \hspace{1cm} (7.2)

The equation for vehicle collision detection can be obtained by taking the ratio of \(u\) and \(v\) as follows,

\[ \frac{u}{v} = \lambda \left( \frac{\frac{x}{z}}{\frac{y}{z}} \right) = \left( \frac{x}{y} \right) \]  \hspace{1cm} (7.3)

\[ \frac{u}{v} = \frac{x}{y} \]  \hspace{1cm} (7.4)

So, by taking distance formula, if two cars namely car 1 and car 2 are considered, then the above equation will become,

\[ \frac{u}{v} = \frac{x_1}{y_1} \]  \hspace{1cm} (7.5)

So, if the ratio of \(u\) and \(v\) remains constant along the period of time then the vehicles car 1 and car 2 will collide after a particular time. This can be shown in the following example.

Consider two cars car 1 and car 2 have coordinates \((x_1, y_1)\) and \((x_2, y_2)\) respectively as shown in figure 7.1. Car 1 is the test car on which camera is mounted and car 2 is the car...
coming towards car 1. It is assumed that both cars are moving with constant speed towards each other.

So, considering above equation, collision detection can be done by,

\[
\text{Collision detection} = \frac{u}{v} = \frac{x_{2} - x_{1}}{y_{2} - y_{1}} = \frac{x_{obj} - x_{cam}}{y_{obj} - y_{cam}} \quad \text{.................................. (7.6)}
\]

As car 1 moves vertically, at time interval 0 of car 1,

\[
\frac{u}{v} = \frac{x_{2} - x_{1}}{y_{2} - y_{1}} = \frac{6 - 0}{6 - 0} = 1 \quad \text{.................................. (7.7)}
\]

At time interval 2,

\[
\frac{u}{v} = \frac{x_{2} - x_{1}}{y_{2} - y_{1}} = \frac{4 - 0}{6 - 2} = \frac{4}{4} = 1 \quad \text{.................................. (7.8)}
\]

And at time interval 4,

\[
\frac{u}{v} = \frac{x_{2} - x_{1}}{y_{2} - y_{1}} = \frac{2 - 0}{6 - 4} = \frac{2}{2} = 1 \quad \text{.................................. (7.9)}
\]

This shows that the ratio of the world coordinates $u$ and $v$ are constant over a period of time for the particular distance travelled by both the cars. As the cars are moving at constant speed, there will be no change in the ratio of world coordinates for any time $t$ in future. Hence, it can be predicted that at time interval 6, both the cars car 1 and car 2 will collide with each other. In this way, the above method can be used to detect the collision between the two vehicles moving in perpendicular direction.
7.2 Case I:

In this case, car 1 and car 2 will collide after a particular time. The code written in Matlab is used to calculate the distance of both cars car 1 and car 2 from the point of origin to the point of collision at regular intervals of time.

The following table shows the distance travelled by both cars. Also, by using the collision detection equation, ratio of the distance travelled by both cars are taken. If the ratio is constant for all time intervals then both the cars will collide.

<table>
<thead>
<tr>
<th>Time</th>
<th>Distance</th>
<th>Time</th>
<th>Distance</th>
<th>( \frac{u}{v} = \frac{x}{y} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150</td>
<td>0</td>
<td>300</td>
<td>( \frac{u}{v} = \frac{300}{150} = 2 )</td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>10</td>
<td>300</td>
<td>( \frac{u}{v} = \frac{300}{150} = 2 )</td>
</tr>
<tr>
<td>20</td>
<td>140</td>
<td>20</td>
<td>280</td>
<td>( \frac{u}{v} = \frac{280}{140} = 2 )</td>
</tr>
<tr>
<td>30</td>
<td>130</td>
<td>20</td>
<td>265</td>
<td>( \frac{u}{v} = \frac{265}{130} = 2.03 )</td>
</tr>
</tbody>
</table>
Table 1: Distance travelled by both the cars and collision detection ratio in case I

As shown in the table, the ratio of the distance travelled by both the cars is almost constant i.e. 2 throughout the time interval. The figures obtained by using Matlab code at different time intervals are shown below.
<table>
<thead>
<tr>
<th>At interval 0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>At interval 10</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>At interval 20</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
</tbody>
</table>
At interval 30

At interval 40
At interval 45

At interval 46
At interval 47

At interval 48
Hence, it can be predicted that the cars car1 and car2 will collide with each other sometime later after time 48. The collision between two cars is shown in the last row of the table 4.2.
7.3 Case II:

In this case, the distance travelled by both the cars and the corresponding ratio is calculated. If the cars are moving at constant speed and if the ratio does not remain constant then the cars car 1 and car 2 will not collide at any time as shown below:

<table>
<thead>
<tr>
<th>Time</th>
<th>Distance</th>
<th>Car 1 ((x_1, y_1)) (Bottom to top)</th>
<th>Car 2 ((x_2, y_2)) (Left to Right)</th>
<th>Collision Detection Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150</td>
<td>0</td>
<td>0</td>
<td>(u = \frac{x}{y} = 2)</td>
</tr>
<tr>
<td>10</td>
<td>135</td>
<td>10</td>
<td>250</td>
<td>(u = \frac{250}{135} = 1.85)</td>
</tr>
<tr>
<td>20</td>
<td>120</td>
<td>20</td>
<td>200</td>
<td>(u = \frac{200}{120} = 1.67)</td>
</tr>
<tr>
<td>30</td>
<td>110</td>
<td>30</td>
<td>140</td>
<td>(u = \frac{140}{110} = 1.27)</td>
</tr>
<tr>
<td>40</td>
<td>95</td>
<td>40</td>
<td>60</td>
<td>(u = \frac{60}{95} = 0.63)</td>
</tr>
<tr>
<td>43</td>
<td>70</td>
<td>43</td>
<td>10</td>
<td>(u = \frac{10}{70} = 0.14)</td>
</tr>
<tr>
<td>44</td>
<td>60</td>
<td>44</td>
<td>0</td>
<td>(u = \frac{0}{60} = 0)</td>
</tr>
</tbody>
</table>
As shown in the table, the ratio of distance travelled by both cars is not constant and goes on decreasing. At some point, the ratio becomes zero indicating that the car 2 has reached the expected point of collision and car 1 is still at some distance away from it. The ratio again goes on increasing indicating that the car 2 moves further away from the point of collision and thus it shows that car 1 will not collide with car 2. The figures obtained by using Matlab code at different time intervals are shown below.

<table>
<thead>
<tr>
<th>45</th>
<th>50</th>
<th>70</th>
<th>10</th>
<th>( \frac{u}{v} = \frac{10}{50} = 0.2 )</th>
</tr>
</thead>
</table>

Table 4.3: Distance travelled by both the cars and collision detection ratio in case II

At interval 0

At interval 10
At interval 20

At interval 30
At interval 40

At interval 43
At interval 44

At interval 45
Table 4.4: Different car positions at different time interval in case II

As shown in the table above, after time 47, car 2 moves away from the point of collision while car 1 still hasn’t reached the point of collision. Thus, the two cars do not collide with each other. This proves that by checking the collision detection ratios obtained by calculating the distance travelled by cars, it can be determined whether or not the two cars will collide with each other.
In this project, the vehicle detection and tracking algorithms were studied in detail and vehicle collision detection algorithm was developed successfully. Also, the transformation of coordinates were studied and implemented in the collision detection algorithm which helped to detect the vehicles accurately. Each step involved in this algorithm was implemented in MATLAB and the corresponding results obtained were used for predicting whether collision will occur between the two vehicles.

Future Work:

There are many new features in ADAS (Advanced Driver Assistance System) that help to avoid collision between vehicles. One of such feature is the navigation system connected to the Adaptive Cruise Control (ACC) and other systems in the vehicle. If the vehicle in front slows down and signals that it is exiting the freeway, then the ACC generally will slow down the host car. But the new GPS assisted ACC system checks that there is an exit in its navigation system and detects the turn signal by using camera and accordingly thinks that the vehicle ahead is exiting and maintain the current speed if its safe to do so.

ITS (Intelligent Transportation System), developed in many countries, is the latest technology in which vehicles and roadways are linked with each other using
telecommunication and information processing technology. In this, vehicles will be equipped with wireless networking along with onboard sensors and control systems, while the sensors and wireless networks will be installed along the roadways. Vehicles will communicate with other vehicles and also with the infrastructure network on road. Information about road accidents, speed and locations of other vehicles would be send to the on board systems of the host vehicle due to which the host vehicle will have accurate scenario of other vehicles around and also of the things happening on the road.
APPENDIX

The following code can be used for collision detection in vehicles. The procedure for converting the captured video into image and finding out the distance travelled by the cars is as follows:

1: Use mmread to view video:

```matlab
video = mmread('movienew.mpg', [1:100], [], false, true);

movie(video.frames)
```

2: Convert frames into images:

```matlab
[x, map] = frame2im(video.frames(1));

a = [x, map];

imshow(a);
```

3: Write the image into the Jpeg File:

```matlab
imwrite(a, '1.jpeg');
```

4. Function to find centroid of the image:

```matlab
function [meanx, meany] = ait_centroid(pic)
[x, y, z] = size(pic);  
% Checking whether the picture is colored or monochromatic, if colored then converting to gray.
```
if(z==1);

else    pic = rgb2gray(pic);

end

im = pic;

[rows,cols] = size(im);

x = ones(rows,1)*[1:cols];    % Matrix with each pixel set to its x coordinate

y = [1:rows]'*ones(1,cols);   % "     "     "    "    "  "   "  y    "

area = sum(sum(im));

meanx = sum(sum(double(im).*x))/area;

meany = sum(sum(double(im).*y))/area;

5. Centroid of the image:

pic = imread('1.JPEG');

[x,y] = ait_centroid(pic);

x

y

imshow(pic); pixval on

In order to prove that the above code can be used for collision detection in vehicles, the following two cases are considered.
REFERENCES

[1] American Honda Safety [online]

[2] Advanced Driver Assistance Systems Take Control in the Car [online]

[3] Intelligent Car: Driver Assistance System [online]


